FEDERAL PUBLIC KEY INFRASTRUCTURE (PKI) TECHNICAL SPECIFICATION: PART D - INTEROPERABILITY PROFILES

DRAFT

Prepared By: Federal PKI Technical Working Group
Edited By: CygnaCom Solutions, Inc.

09/27/95

CygnaCom Solutions, Inc. ♦ #201, 1477 Chain Bridge Rd ♦ McLean, Virginia 22101
Telephone (703) 848-0883 ♦ Fax (703) 848-0960

FEDERAL PUBLIC KEY INFRASTRUCTURE (PKI)
TECHNICAL SPECIFICATION: PART D - INTEROPERABILITY PROFILES

Table Of Contents

SECTION 1 (INTRODUCTION) **6**

1.1 BACKGROUND 6
1.2 PURPOSE AND SCOPE 8
1.3 APPROACH 9
1.4 AUDIENCE 11
1.5 DOCUMENT ORGANIZATION 11

SECTION 2 (FEDERAL PKI) **12**

2.1 PKI STRUCTURE 13
2.2 SIGNATURE ALGORITHMS 14
2.2.1 End Systems Verification 15
2.2.2 Multiple Certification Paths 18
2.2.3 Trusted Gateways 19
2.2.4 Universal DSS Adoption 21
2.2.5 RSA Adoption 21
2.2.6 Alternatives Summary 22
2.3 KEY DISTRIBUTION ALGORITHMS 24
2.3.1 One Algorithm Certificate 26
2.3.2 Multiple Algorithm Certificates 27
2.3.3 Trusted Gateways 28
2.3.4 Universal Diffie Hellman Adoption 30
2.3.5 Universal RSA Adoption 30
2.3.6 Alternatives Summary 31
2.4 DATA ENCRYPTION ALGORITHMS 32
2.5 DATA FORMATS 33
2.5.1 Certificate Format 33
2.5.2 Certificate Revocation List Format 36
2.6 DATA DISSEMINATION 37
2.6.1 Certificate Dissemination 37
2.6.2 Revocation Notification 38
2.6.3 Directory 39
2.7 POLICY 40
2.8 NAMING 41
2.9 MISSI ENHANCEMENTS 42

SECTION 3 (INTERNET PKI) **44**

3.1 INTEROPERABILITY CONSTRAINTS 44
3.1.1 PKI Structure 44

3.1.2 Signature Algorithms *45*
3.1.3 Key Distribution Algorithms *45*
3.1.4 Data Encryption Algorithms *45*
3.1.5 Data Formats *45*
3.1.6 Data Dissemination *46*
3.1.7 Policy *46*
3.1.8 Naming *47*
3.2 INTEROPERATION WITH FEDERAL PKI 47
3.3 FEDERAL PKI ENHANCEMENTS 48

SECTION 4 (CANADIAN FEDERAL PKI) **50**

4.1 INTEROPERABILITY CONSTRAINTS 50
4.1.1 PKI Structure *50*
4.1.2 Signature Algorithms *51*
4.1.3 Key Distribution Algorithms *51*
4.1.4 Data Encryption Algorithms *51*
4.1.5 Data Formats *51*
4.1.6 Data Dissemination *53*
4.1.7 Policy *53*
4.1.8 Naming *53*
4.2 INTEROPERATION WITH THE FEDERAL PKI 53
4.3 FEDERAL PKI ENHANCEMENTS 54

SECTION 5 (USPS PKI) **56**

5.1 INTEROPERABILITY CONSTRAINTS 56
5.1.1 PKI Structure *56*
5.1.2 Signature Algorithms *56*
5.1.3 Key Distribution Algorithms *56*
5.1.4 Data Encryption Algorithms *57*
5.1.5 Data Formats *57*
5.1.6 Data Dissemination *59*
5.1.7 Policy *59*
5.1.8 Naming *61*
5.2 INTEROPERATION WITH THE FEDERAL PKI 61
5.3 FEDERAL PKI ENHANCEMENTS 64

REFERENCES **66**

LIST OF ACRONYMS **68**

APPENDIX A (TRUSTED DIRECTORY) **71**

APPENDIX B (DIFFIE HELLMAN KEY DISTRIBUTION USING DSS PARAMETERS) **73**

APPENDIX C (TIS PCA POLICY STATEMENT) **76**

APPENDIX D (CROSS CERTIFICATION PROCESS) **82**

FEDERAL PUBLIC KEY INFRASTRUCTURE (PKI)
TECHNICAL SPECIFICATION: PART D - INTEROPERABILITY PROFILES

List of Figures

FIGURE 1-1 -- FEDERAL PKI ARCHITECTURE 7
FIGURE 1-2 -- ELEMENTS OF PKI INTEROPERATION 8
FIGURE 2-1 -- PROPOSED FEDERAL PKI 14
FIGURE 2-2 -- END SYSTEMS APPROACH FOR SIGNATURE VERIFICATION 16
FIGURE 2-3 -- MULTIPLE CERTIFICATION PATHS FOR SIGNATURE VERIFICATION 18
FIGURE 2-4 -- TRUSTED GATEWAY APPROACH FOR SIGNATURE VERIFICATION 20
FIGURE 2-5 -- KEY DISTRIBUTION USING SINGLE ALGORITHM CERTIFICATE 26
FIGURE 2-6 -- KEY DISTRIBUTION USING MULTIPLE ALGORITHM CERTIFICATES 28
FIGURE 2-7 -- TRUSTED GATEWAY KEY DISTRIBUTION 29
FIGURE 3-1 -- INTERNET PKI STRUCTURE 44
FIGURE 3-2 -- 1988 CCITT X.509 AND PEM CERTIFICATE FORMAT 45
FIGURE 3-3 -- PEM CRL FORMAT 45
FIGURE 3-4 -- FEDERAL PKI TO INTERNET CROSS-CERTIFICATION OPTIONS 48
FIGURE 4-1 -- DCEKMS STRUCTURE 50
FIGURE 5-1 -- USPS PKI STRUCTURE 56
FIGURE 5-2 -- USPS PKI - FEDERAL PKI CROSS CERTIFICATION 62
FIGURE A-1 -- USER - TRUSTED DIRECTORY TRANSACTION 71
FIGURE A-2 -- CA - TRUSTED DIRECTORY TRANSACTION 72
FIGURE B-1-- KEY DISTRIBUTION PROTOCOL 74

FEDERAL PUBLIC KEY INFRASTRUCTURE (PKI)
TECHNICAL SPECIFICATION: PART D - INTEROPERABILITY PROFILES

List of Tables

TABLE 2-1: SIGNATURE ALGORITHMS INTEROPERABILITY ALTERNATIVES ASSESSMENT 22
TABLE 2-2: FIPS KEY DISTRIBUTION ALGORITHM OPTIONS 24
TABLE 2-3: KEY DISTRIBUTION ALGORITHMS BASED ON SINGLE CERTIFICATE 26
TABLE 2-4: KEY DISTRIBUTION ALGORITHMS INTEROPERABILITY ALTERNATIVES ASSESSMENT 31
TABLE 2-5: SUMMARY OF VERSION 3 CERTIFICATE EXTENSIONS 35
TABLE 2-6: VERSION 2 CRL BASIC EXTENSIONS 37
TABLE 3-1: FEDERAL PKI ENHANCEMENTS FOR INTEROPERATION WITH THE INTERNET PKI 48
TABLE 3-2: INTERNET PKI ENHANCEMENTS FOR INTEROPERATION WITH THE FEDERAL PKI 49
TABLE 4-1: SUMMARY OF DCEKMS VERSION 3 CERTIFICATE EXTENSIONS 51
TABLE 4-2: SUMMARY OF DCEKMS VERSION 2 CRL EXTENSIONS 52
TABLE 4-3: FEDERAL PKI ENHANCEMENTS FOR INTEROPERATION WITH THE DCEKMS 54
TABLE 4-4: DCEKMS ENHANCEMENTS FOR INTEROPERATION WITH THE FEDERAL PKI 54
TABLE 5-1: SUMMARY OF POSTAL VERSION 3 CERTIFICATE EXTENSIONS 58
TABLE 5-2: FEDERAL PKI ENHANCEMENTS FOR INTEROPERATION WITH THE POSTAL PKI 64
TABLE 5-3: POSTAL PKI ENHANCEMENTS FOR INTEROPERATION WITH THE FEDERAL PKI 65

SECTION 1 (INTRODUCTION)
INTRODUCTION

1.1 Background

The purpose of a Public Key Infrastructure (PKI) is to facilitate generation and dissemination of public key certificates using automated means. Based on a study conducted by the National Institute of Standards and Technology (NIST), in conjunction with seven other Federal agencies [1], the United States Federal PKI will consist of the following elements of a trust tree hierarchy[1]:

- **Policy Approving Authority (PAA):** This is the root of the Federal PKI tree structure. The PAA approves the security policy of each second level node in the tree.
- **Policy Creation Authority (PCA):** This entity (also known as a policy certification authority in some literature) is a second level node of the Federal PKI. A PCA formulates a security policy under which it and its subordinates will issue public key certificates.
- **Certification Authority (CA):** This entity is a third level node in the Federal PKI. A CA issues certificates to users under the parent PCA defined security policy.
- **Organizational Registration Authority (ORA):** This entity authenticates the identity and the organizational affiliation of the users.
- **User:** This entity is a leaf of the Federal PKI tree. A user is generally an individual. The scope of users who can register with a CA is a local matter for the CA. It may be further constrained by the PCA policy. For example, it may include only the Federal employees, on-site contractors, all contractors, or citizens an agency is serving. Defining the user population is beyond the scope of this paper.

The term Certificate Management Authority (CMA) is used to designate any or all of the first three layers of the certification hierarchy: PAA, PCA, CA. In addition to the above hierarchy, the Federal PKI also contains the directory system[2] which can be used to store and retrieve certificates and Certificate Revocation Lists (CRLs). Figure 1-1 illustrates the Federal PKI. Of the proposed Federal PKI components, the Department of Defense

[1] The PKI structures presented in this paper are logical and conceptual entities. Issues such as whether each CMA is a separate physical entity, government owned and operated, government owned contractor operated, or just a service, is an acquisition matter that is beyond the scope of this paper.
[2] In the rest of the paper, the term directory is used to mean the directory system.

(DoD) Multilevel Information Systems Security Initiative (MISSI) type 2[3] has already been implemented. The current implementation includes an NSA managed PAA, a DoD PCA, an NSA CA, and a Directory System Agent (DSA). All of these components are located at Finksburg, Maryland. The current MISSI implementation also includes a GSA PCA, a GSA CA, and a DSA, all located at Berryville, Virginia. MISSI is undergoing enhancements to support mission needs and to support the evolving version 3 certificate and version 2 CRL standards.

The National Information Infrastructure (NII) Federal Information Security Infrastructure Program Management Office (SI-PMO) has been established in accordance with the direction of the Government Information Services Working Group and the NII Security Issues Forum. The SI-PMO is administered by the General Services Administration (GSA) and is staffed by several Federal (DoD and non-DoD) agencies. The SI-PMO is responsible for making the Federal PKI a reality. The SI-PMO will manage the PAA and ensure interoperability of the Federal PKI with other PKIs. This document forms one of the key inputs to the SI-PMO. For further details on SI-PMO plans to help implement the Federal PKI, see [16]. The MISSI PAA is located at Finksburg, MD, and managed by the National Security Agency (NSA). The Federal PAA will replace the MISSI PAA. The Federal PAA will be managed by the SI-PMO. The SI-PMO will determine the location of the PAA.

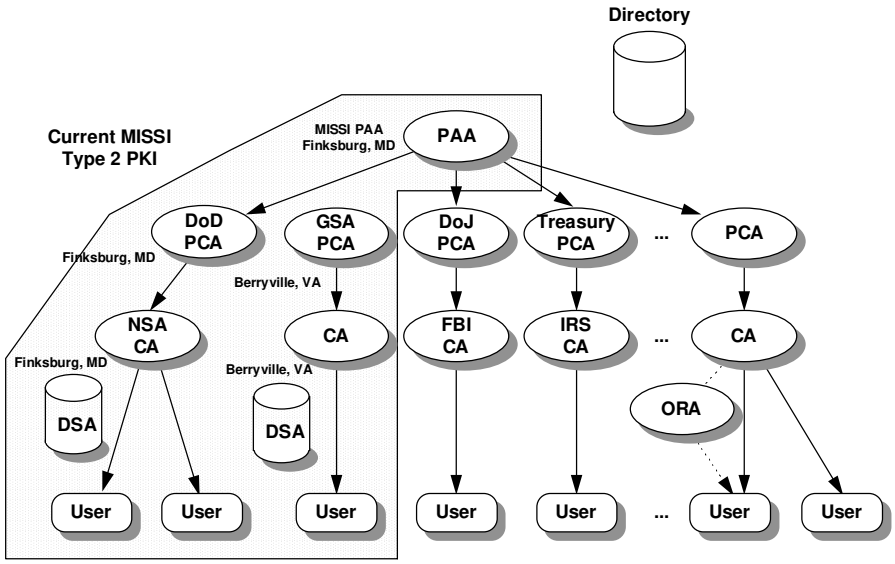

Figure 1-1 -- Federal PKI Architecture

[3] The term MISSI type 2 refers to domains dealing with the protection of sensitive-but-unclassified information. In the rest of this paper, whenever the term MISSI appears, it implies MISSI type 2, unless explicitly stated otherwise.

While the Federal PKI satisfies the need for Federal PKI users to interoperate among themselves, the Government recognizes the need for Federal PKI users to securely communicate with users who are not registered with the Federal PKI. These users could represent private citizens, corporations, state and local governments, and other governments. It is important that the Federal PKI users be able to securely communicate with these entities in order to make electronic commerce a reality on a national and international scale.

1.2 Purpose and Scope

The objective of this document is to define the Federal PKI so that it can interoperate with other PKIs. Thus, this paper first defines a Federal PKI that is likely to interoperate with other PKIs. Once this Federal PKI has been defined, interoperability with specific PKIs is analyzed to determine required enhancements to the Federal PKI and to the other PKIs. In summary, the purpose of this document is to:

- Define the Federal PKI,
- Define interoperability between the Federal PKI and each of the following PKIs: Internet, Canadian Government, and Commercial or Postal CA (assuming the Postal CA is not in the Federal PKI),
- Describe the relationship between the Postal Service CA and the Federal PKI, and
- Allocate additional functionality required for interoperation of the Federal PKI components (e.g., CMA, Directory, user agent, etc.), and similar components of other PKIs.

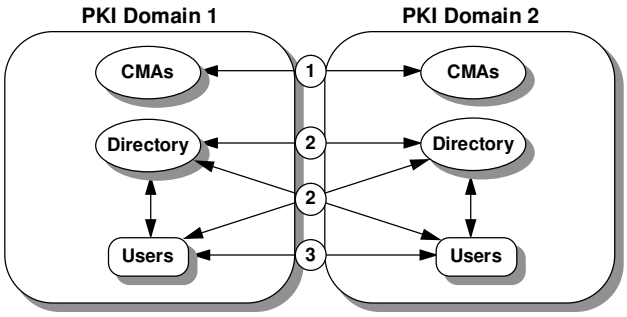

Figure 1-2 -- Elements of PKI interoperation

As illustrated in Figure 1-2, in order for a user from one PKI domain to interact with another user in another PKI domain, the following must hold true:

1. There must be a trust chain among the CMAs,

2. The users must be able to obtain certificate chains and CRLs, typically by accessing the directories in each others' domain (directly, through referrals, or through directory to directory chaining), and
3. The users must be able to interact with each other and be able to process each other's application protocol.

The first issue (CMA - CMA trust chain) is considered within the scope of this paper. Only trust chain processing related aspects of the second issue are considered within the scope of this paper. Thus, the directory interoperation as it relates to the ability to retrieve certificates and CRLs issued by another domain are considered within the scope of this study.[4] The third issue, user - user interaction, is considered beyond the scope of this study. However, it is recognized that user - user interaction is just as important as other interoperability issues. Without user - user interoperability, including at the application layer, true interoperability can not be achieved. For example, it is recognized that in order for a user to communicate with a MISSI Defense Messaging System (DMS) user, the user must be able to process the Message Security Protocol (MSP).

1.3 Approach

The Federal PKI users and the other PKI users must be able to communicate securely. Secure communication between two parties means the ability to invoke security service such as:

- Sender Authentication
- Message Integrity
- Confidentiality (driven by privacy, need to know, or sensitivity)
- Sender Non-repudiation
- Proof-of-Receipt

The two parties should be able to communicate securely in spite of the following differences:

- **Signature Algorithms:** The two users could use different public key cryptography algorithms for hashing (e.g., Secure Hash Standard or (SHS), Message Digest 5 (MD-5), etc.) or for digital signatures (e.g., Digital Signature Algorithm, Rivest, Shamir, and Adelman (RSA), etc.)

[4] It is assumed that each domain can populate its own directory. Thus, only the ability to retrieve directory information from another domain is required.

- **Key Distribution Algorithms:** The two users could use different public key cryptography based key distribution algorithms (e.g., Diffie-Hellman, RSA, Key Exchange Algorithm (KEA), etc.)
- **Data Encryption Algorithms:** The two users could use different data encryption algorithms (e.g., Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), etc.)
- **Data Formats:** The two users could have different certificate or CRL formats, due to various standards, options in standards, or versions of standards
- **PKI Structure:** The two users could be registered in PKIs with different architectures (e.g., strict tree hierarchy, tree hierarchy with multiple parents, graph with peer certification, etc.)
- **Data Dissemination:** The PKIs could have different data (certificate and CRL) dissemination techniques (e.g., X.500 based, trusted directory[5], proprietary, etc.)
- **Policy:** The two users could be certified under two different certificate policies.
- **Naming:** The two PKIs could use naming conventions that are different. The difference could cause incompatibility or name collision.

In the rest of this paper, these differences are referred to as interoperability constraints.

The interoperability analysis presented in this document was conducted using the following chronological steps:

- **Define the Federal PKI:** The Federal PKI is defined in terms of the interoperability constraints using the following inputs: NIST Public Key Infrastructure Study [1], MISSI type 2 Security Management Infrastructure (since MISSI is the only operational component of the Federal PKI) [2], Federal PKI Concept of Operations [3], and Federal PKI Technical Security Policy [4]. The definition of the Federal PKI was also based on the need to interoperate with other existing and planned infrastructures. Various options for interoperation were analyzed prior to making recommendations for the Federal PKI. Based on the recommendations, enhancements to MISSI components were identified.
- **Describe and Analyze other PKIs:** In this step, each of the following three PKIs (Internet, Canada Federal, and United States (US) Postal Service) are described and analyzed independently. Each PKI is defined in terms of the interoperability constraints. Approaches for interoperation with the Federal PKI are identified and recommendations are made. Based on the recommendations, enhancements to the Federal PKI and the other PKIs (Internet, Canada Federal, or United States (US) Postal Service) are identified.

[5] For a description of trusted directory concept, see Appendix B.

1.4 Audience

The anticipated audience for this document is the Federal PKI - Technical Working Group (TWG). This paper was written for the Federal PKI - TWG to assist the group in recommending approaches for PKI interoperation. A reader of this report is expected to have an in-depth understanding of the following concepts and technologies:

- Information security
- Public key cryptography
- Public key infrastructures
- Cryptography and PKI related national and international standards

Another audience for this document is the SI-PMO. This report will form one of the key inputs for the PKI interoperation and for the Federal PAA requirements.

1.5 Document Organization

This section has provided document introduction. The Federal PKI, Internet PKI, Canadian Federal PKI, and USPS PKI are described in Sections 2 through 5. Sections 3 through 5 contain sub-sections analyzing interoperability constraints of each PKI with the Federal PKI.

Appendices A and B provide technical details on trusted directories and key distribution alternatives. Appendix C contains a statement of the security policy of the Trusted Information Systems (TIS), Inc. commercial assurance PCA registered within the Internet PKI.

SECTION 2 (FEDERAL PKI)
FEDERAL PKI

The Federal PKI is a strict hierarchy as illustrated in Figure 2-1.[6] It is used for sensitive-but-unclassified communications inside the Federal Government and with outside organizations and citizens. The structure is based on the NIST PKI Study [1]. The root of the hierarchy (PAA) is managed by the SI-PMO housed at the General Services Administration (GSA). The PCAs are managed by cabinet level departments and cabinet level agencies, while CAs are maintained by agencies.

The MISSI hierarchy provides a much greater degree of detail to the definition of the Federal PKI. The MISSI type 2 hierarchy is consistent with the NIST PKI Study findings. Because the MISSI hierarchy is well detailed and currently implemented, the Federal PKI is based on the MISSI type 2 hierarchy and the enhancements required for the Federal PKI to interoperate with other PKIs.

The Federal PKI will include all Federal agencies, however, there are other quasi-governmental agencies that must be specifically included or excluded from the Federal PKI. The most notable of these is the U.S. Postal Service, which is developing an Electronic Commerce Services (ECS) CA [5]. The Postal Service plans to offer certificate registration, retrieval, and revocation services to customers on a fee-for-service basis. The Postal ECS CA is excluded from the Federal PKI for two reasons:

1. The Postal Service is not a Federal Government organization.
2. The ECS CA provides fee-for-service to government, citizens and private organizations.

Since the private sector is offering these same services to citizens and industry, it may be inappropriate for the Federal Government to offer such services on a fee basis. Including the Postal Service ECS CA in the Federal PKI could be viewed as allowing the Federal PKI to offer competitive commercial services. Thus, it is best to exclude the Postal CA from the Federal PKI and maintain a clear distinction. However, this does not prevent Federal agencies from procuring Postal Service's system or services to meet their PKI requirements.

[6] The view presented in this paper deals largely with the executive branch of the Federal Government. The Legislative and Judiciary branches could also come under the same structure. These two branches could have their PCAs under the Federal PAA. The SI-PMO (an executive branch entity) managing a PAA for the other two branches may be acceptable, or the PAA may require co-administration from the three branches of the Federal Government.

Despite the distinction between the Federal PKI and the USPS ECS CA, the two organizations need to closely interoperate. The USPS's unique qualities of nation-wide presence, longevity, stability, and public trust positions the Postal Service to facilitate secure electronic commerce between the Federal Government and citizens and industry. The Postal Service ECS CA can issue certificates to citizens and organizations who want to conduct electronic commerce, electronic data interchange (EDI), or electronic filing with the agencies in the Federal Government. Cross-certification between the Federal PKI and the Postal ECS CA would facilitate secure communication between the Federal Government and Postal Service customers. Details of how the Federal PKI can interoperate with the Postal ECS CA are explored in Section Five (5).

2.1 PKI Structure

In order to provide agencies flexibility, the Federal PKI will be largely hierarchical as opposed to the strict hierarchy depicted in Figure 1-1. The structure will have the following salient features:

- CMAs will issue CA certificates to their subordinates. In addition, a CMA and a subordinate will issue each other certificates for a cross-certificate pair.
- CMAs may cross certify each other. They will be encouraged to cross certify only for the appropriate name spaces and policies.
- The hierarchy will not be necessarily limited to three levels.
- A node at any level in the hierarchy may certify users, including the PAA.
- PAA will not only perform CA functions, but will also be responsible for the following: policy development and coordination, policy object identifiers administration, policy enforcement with PCAs, cross certification with other domains, and cross certification agreement enforcement with other domains.
- The term PCA will stand for Principal Certification Authority. The policy creation role will not necessarily be limited to the PCAs, since policy may be created by a CA at any level using version 3 certificates. Typically PCA will be cabinet level departments and large independent agencies.
- A PCA will perform certification function. In addition, a PCA may perform policy development, coordination and enforcement function on behalf of itself and its subordinates. Alternatively, a separate organization may perform these policy management function. The decision will be left up to each department and agency and may vary from organization to organization.
- Typically, a CA will be an agency. A CA will perform certification function. In addition, a CA may perform policy development, coordination and enforcement function on behalf of itself and its subordinates. Alternatively, a separate organization may perform these policy management function. The decision will be left up to each agency and may vary from agency to agency.

As illustrated in Figure 1-1, current Federal PKI consists of the PAA located at Finksburg, Maryland; the DoD PCA located in Finksburg, Maryland; the GSA PCA located in Berryville, Virginia; an NSA CA located at Finksburg, Maryland; and a GSA CA located at Berryville, Virginia. Two DSA are also located at these two sites -- Berryville, and Finksburg. In the future, the Federal PKI will be augmented by several PCAs and CAs located at the cabinet and agency levels respectively. Currently, the MISSI PAA is located at Finksburg, MD, and managed by the National Security Agency (NSA). The Federal PAA will replace the MISSI PAA. The Federal PAA will be managed by the Security Infrastructure - Program Management Office (SI-PMO) administratively supported by the General Services Administration (GSA). The SI-PMO will determine the location of the PAA. The Proposed Federal PKI will be a hierarchy as described in Figure 2-1.

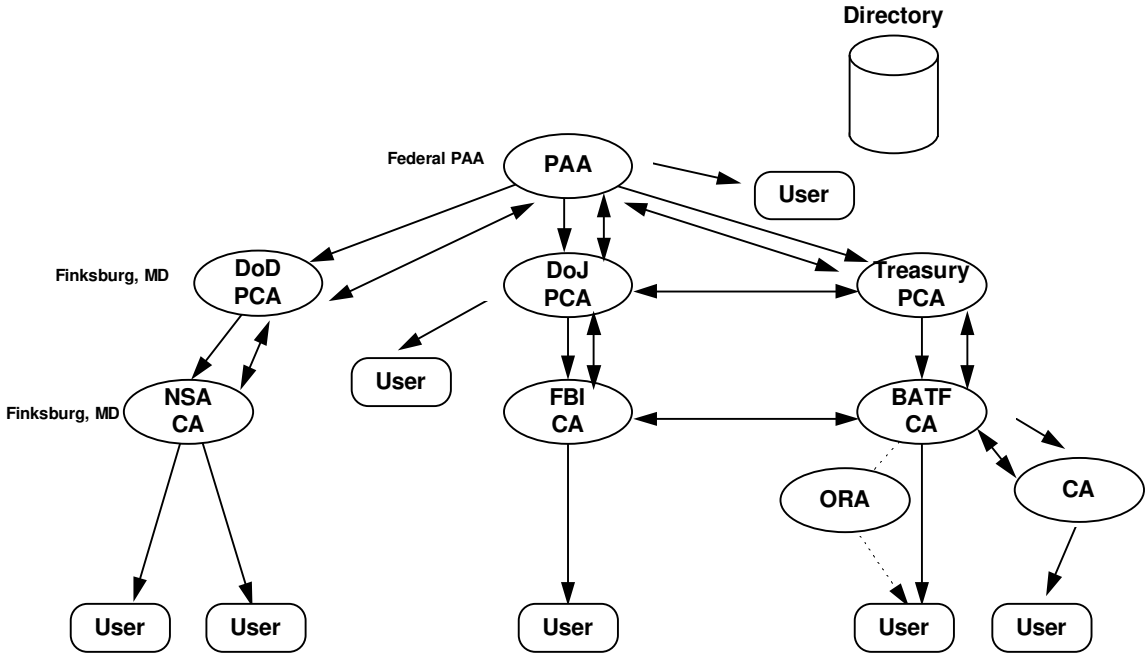

Figure 2-1 -- Proposed Federal PKI

2.2 Signature Algorithms

There are several public key cryptography algorithm suites. Examples include:

- RSA with MD-5 message digest
- DSS with Secure Hash Standard (SHS) message digest
- El Gamal cryptosystem

Of these, RSA and DSS are the most widely used. Federal PKI users must use the Digital Signature Standard (DSS) as specified in Federal Information Processing Standards (FIPS) 186. DSS includes both the Digital Signature Algorithm and Secure Hash Algorithm (SHA) as defined in FIPS 186 and FIPS 180-1 respectively. The above suite is a mandatory Federal standard and is currently used by the only operational component of the Federal PKI, the MISSI. Any exception to FIPS mandates requires a waiver issued by an agency head which requires extensive justification, persuasion, notification of Congress, and publication in the Federal Register.

Unfortunately, commercial digital signature implementations make wide use of the RSA signature algorithm and rarely use the DSS. (See the later sections that introduce other PKIs for more information.) Thus, Federal users, who are mandated to use DSS face interoperation issues when interacting with non-Federal users employing RSA.

In addition, Federal users may, due to cost and availability reasons, buy commercial applications (e.g. Secure Hyper Text Transport Protocol or S-HTTP) that employ the RSA cryptosystem. If such applications require installation of an infrastructure to facilitate key management, it would be most economical for the Federal PKI to provide that key management support. Therefore, the Federal PKI may have to interoperate with non-Federal users employing RSA in addition to any Federal applications which by necessity employ RSA.

The following five options were considered for interoperation between Federal users using DSS and non-Federal users using the RSA cryptosystem:

- Allow end systems to verify signatures from both cryptosystems (Federal users would add RSA verification, and RSA users would add DSS verification capabilities to their workstations)
- Issue both RSA and DSS certificates to Federal users who need to communicate with users registered in domains using RSA
- Use trusted gateways
- Convince the rest of the world to use DSS
- Abandon DSS FIPS for RSA

Each of these options is described and then analyzed below. The analysis is summarized in Table 2-1, Signature Alternatives Assessment.

2.2.1 End Systems Verification

One way to provide interoperability between various public key cryptosystems is to enable end users to verify signatures from multiple types of cryptosystems. Under this approach (see Figure 2-2), the Federal PKI user workstations would have the RSA verification suite

(among others), allowing them to verify certificate chains, CRLs and messages from RSA users. Similarly, others would have the DSS verification suite to verify signatures from the Federal PKI users.

For example, if Alice of the Department of Defense (DoD) is communicating with Bob in the commercial sector, Alice would sign her transactions to Bob using DSS. Bob would verify Alice's signatures using DSS, but would sign any return messages using the RSA algorithm, confident that Alice can verify RSA signatures. Alice will not sign transactions using the RSA cryptosystem (due to a FIPS mandate), and Bob probably would not sign messages using the DSS cryptosystem.

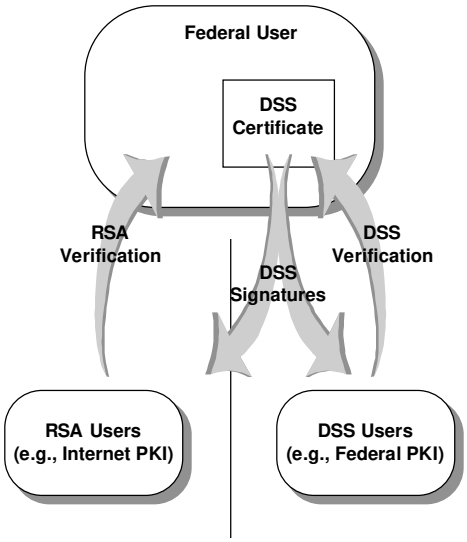

Figure 2-2 -- End Systems Approach for Signature Verification

In order to allow end users to verify signatures made with foreign cryptosystem algorithms, there must exist an appropriate chain of trust to the foreign certificates. To effect this, the appropriate CMA in the Federal PKI will issue a certificate to the appropriate RSA CMA. The Federal CMA will sign the RSA public key of the RSA CMA using the Federal CMA's DSS private key. Similarly, the RSA CMA will sign the Federal CMA's DSS public key using the RSA CMA's RSA private key.

The end system approach will require implementation of RSA verification software on Federal user workstations. It will also require implementation of DSS verification software on RSA user workstations. Informal discussions with product vendors have revealed that in spite of the DSS patent dispute, vendors will tend to provide DSS and RSA capability at little added cost. Discussions with five vendor experts revealed a common theme -- the need for both DSS and RSA to coexist in a technology-agile environment. In other words, vendors feel that users need to verify signatures from a variety of algorithms, including new algorithms and future technologies, as is evident from

their dual cryptosystem product offerings. Examples include, American Telephone and Telegraph's (AT&T) Secret Agent, Northern Telecom's Entrust, and Datakey's Signasure. The situation is akin to Microsoft Word having the capability to read and create documents in WordPerfect format. The scaleability of this approach is good because each user has a single certificate. It is not clear, however, whether this approach requires a FIPS waiver for verifying RSA signatures. A strict interpretation of the language in the FIPS 186 would suggest that the Federal users can only verify DSS signatures.

Further complicating the advantages and disadvantages of the end-system approach is the scenario where Federal agencies employ commercial RSA applications requiring key management infrastructure. Federal PKI should be able to interoperate with these applications. In this case, there are four possible scenarios for these RSA applications vis-à-vis PKI:

1. There is no public key management requirement. Users generate and maintain public keys locally and do not need to share the public keys with others.

2. The public key management is embedded with the application.

3. The application assumes existence of an infrastructure for key management to distribute the public keys. The application does not interact with the infrastructure. It assumes some other local software (e.g., user agent) will interact with the infrastructure.

4. The application assumes existence of an infrastructure for key management to distribute the public keys. The application interacts with the infrastructure using embedded capabilities. This means that the application parses certificates and CRLs, builds certification paths, and validates those paths. These functions become quite complex for X.509 v3 certificates once validity, revocation, name constraint, policy constraints, etc. are factored in.

Of the above four scenarios, numbers one and two do not affect PKI. Scenario three does affect PKI; however, an end system approach makes integration seamless because the user (entity) RSA key will be certified using the DSS key of the CMA. The end system (e.g. on the users' desktop) should be able to process this kind of certificate. The fourth scenario above is not likely to happen. Vendors are not likely to develop applications that also include the capability to interact with an external PKI. The first three are much more likely. Standards are being developed in the area of Cryptographic Applications Programming Interface (CAPI). Applications that integrate cryptography using these standards should be able to easily replace or augment RSA based cryptosystem with a DSS based one. Thus, properly built software may help mitigate the fourth situation also.

Hence, COTS RSA applications will be able to interoperate with the PKI under the end systems approach.

2.2.2 Multiple Certification Paths

A second solution to incompatibilities between the Federal Government's use of DSS and other public key cryptosystems' use of RSA is to provide RSA certificates to the Federal users who need to communicate with RSA users. Both RSA and DSS certificates would have to be issued to all the CMAs in the Federal PKI. Each Federal user would receive a DSS certificate. In addition, a Federal user would receive an RSA certificate if he needs to communicate with an RSA user. The RSA certificate will be signed using the RSA key of the appropriate CMA.

An appropriate CMA in the Federal PKI will issue a certificate to the appropriate external CMA that uses RSA. The Federal CMA will sign the RSA public key of the RSA CMA using the Federal CMA's RSA private key. Similarly, the RSA CMA will sign the Federal CMA's RSA public key using the RSA CMA's RSA private key.

Under this alternative, when a Federal PKI user (e.g. Alice) wishes to communicate with a non-Federal RSA user (e.g. Bob), she would sign her transactions with her RSA private key (see Figure 2-3). Bob would obtain Alice's RSA certificate chain to verify her transactions.

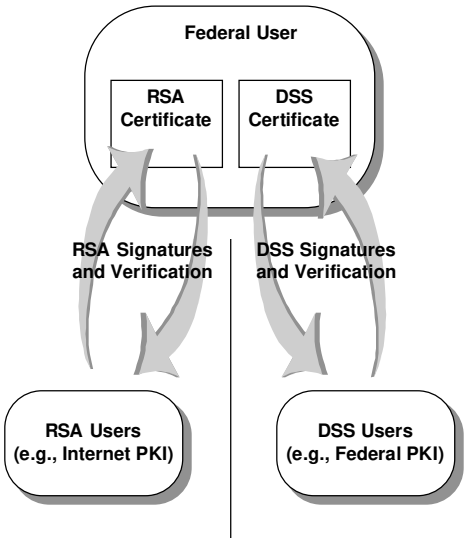

Figure 2-3 -- Multiple Certification Paths for Signature Verification

The multiple certification paths approach is likely to add to infrastructure costs in terms of key generation and certificate storage for multiple algorithms. In essence, it is creating a completely parallel logical PKI hierarchy; although, personnel and computing resources may be shared by the DSS and RSA based Federal PKI.

This alternative is likely to complicate end-user or user-software decisions, specifically certificate selection for signature and/or encryption. To further illustrate this point, let us say that Alice in the DoD wants to send a message to Bob. Let us say that, Alice has both DSS and RSA certificates. Alice's end user software must retrieve Bob's certificate(s) (or employ some other means) in order to determine whether to use the DSS or the RSA certificate to sign a message. The most obvious and practical approach for determining Bob's verification algorithm is to assume that it is the same as his signing algorithm. Assuming Bob has all his signature certificates (we use the term signature here since under X.509 version 3 a public key could be registered for purposes other than digital signatures, e.g. key distribution) issued under one algorithm, the choice is simple. But, what if Bob has multiple signature certificates (which may be the case if Bob is registered under the Federal PKI)? Alice can use DSS in that case. The decision making is not problematic. But, it requires Alice to have a priori knowledge about Bob's preferred verification algorithm or to infer something from his certificates. Thus, Alice must retrieve Bob's certificates even for signing. Unfortunately, this decision-making logic does add to the complexity of end-user software.

Scaleability is not a major issue with this approach, since it only doubles the number of certificates. However, using this approach with the Federal PKI will violate the mandatory use of FIPS standards for digital signatures (i.e., FIPS 186; DSS). Additionally, the approach does not fully exploit the standards -- the intent of the standards is to allow multiple algorithms to interoperate. Under this alternative, the end systems in the Federal PKI domain will require both DSS and RSA signature and verification software.

In addition to the above considerations for the multiple certification paths approach, it is worth noting that interoperation with applications is facilitated -- i.e. if federal agencies employ commercial RSA applications requiring key management infrastructure, users can employ the Federal RSA hierarchy for RSA keys and certificates.

2.2.3 Trusted Gateways

Under the trusted gateway alternative, the Federal PKI will deploy a gateway trusted by the entire Federal user community. There are several ways for design and operation of trusted gateways. Figure 2-4 illustrates one of them. When Federal users receive messages from the RSA users, they will send these messages to the gateway for verification (see Figure 2-4). The gateway will have the RSA and DSS signature and verification capability. The gateway will send a response to the user with the gateway's DSS signature. Similarly, a Federal user will send the message to RSA users through the gateway. The gateway will verify the Federal user's signature, sign it using RSA, and forward the message to the RSA user.

The PAA in the Federal PKI will issue a DSS certificate to the trusted gateway. An appropriate RSA CMA will issue an RSA certificate to the trusted gateway.

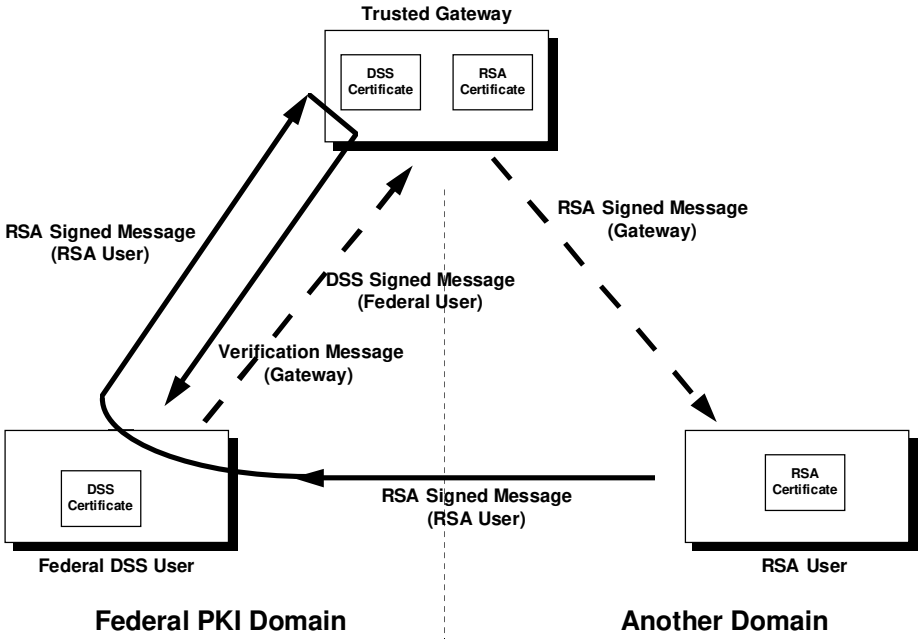

Figure 2-4 -- Trusted Gateway Approach for Signature Verification

Design and operation of trusted gateways to handle message traffic is likely to be costly and may not be scaleable. Security will be a minus for the trusted gateway approach due to the need to build another trusted system. End user trust and non-repudiation may also be impacted due to the introduction of an intermediary in every transaction. There are no standards for trusted gateways.

Additionally, since the gateway must be trusted by both Federal and non-Federal users who will see only the gateway's signature, the Federal Government must vouch for transactions signed by the gateway in order for the Federal users and non-Federal users to trust the gateway's signature.

Let us now consider the scenario of Federal use of commercial RSA applications requiring key management infrastructure. There are four possible scenarios vis-à-vis PKI:

1. There is no public key management requirement. Users generate and maintain public keys locally and do not need to share the public keys with others.

2. The public key management is embedded with the application.

3. The application assumes existence of an infrastructure for key management to distribute the public keys. The application does not interact with the infrastructure. It assumes some other local software (e.g., user agent) will interact with the infrastructure.

4. The application assumes existence of an infrastructure for key management to distribute the public keys. The application interacts with the infrastructure using embedded capabilities. This means that the application parses certificates and CRLs, builds certification paths, and validates those paths. These functions become quite complex for X.509 v3 certificates once validity, revocation, name constraint, policy constraints, etc. are factored in.

Of the above four scenarios, 1 and 2 do not affect PKI. Scenario 3 affects PKI. The trusted gateway can sign the RSA public key of the user (entity). The fourth approach is not likely to happen. Vendors are not likely to develop applications that also include interaction capability with an external PKI whereas the first three scenarios are more likely. Standards are being developed in the area of Cryptographic Applications Programming Interface (CAPI). Applications that integrate cryptography using these standards should be able to easily replace or augment an RSA-based cryptosystem with a DSS-based one. Thus, properly built software may also help mitigate the fourth situation. The fourth scenario can also be accommodated by the gateway issuing the RSA certificates Hence, COTS RSA applications will be able to interoperate with the PKI under the trusted gateway approach. But, in two of the scenarios, the gateway will be required to issue RSA certificates.

2.2.4 Universal DSS Adoption

Under this approach, the commercial sector will be convinced to support the use of DSS with all electronic signature applications. The installed base of the RSA cryptosystem within the commercial sector is an obstacle for this approach. If this scenario came to pass, the Federal PKI will not have to worry about interoperability in terms of signature algorithm. Thus, this alternative is not analyzed further in this paper.

2.2.5 RSA Adoption

Under this approach, the Federal Government will abandon DSS as a FIPS and make RSA an exclusive new FIPS.

The adoption of RSA as a FIPS will have significant cost impact on the existing Federal PKI component, namely MISSI. In addition, Federal users will not be able to communicate with DSS users. For example, Shell Oil is planning to use DSS as their signature algorithm, and several foreign governments are planning to use both DSS and RSA.

This approach will facilitate use of commercial RSA based applications with no interoperability concerns.

2.2.6 Alternatives Summary

Table 2-1 provides a summary assessment of the alternatives. Universal DSS adoption is omitted since it seems unlikely that DSS will become a universal standard to the exclusion of RSA.

Table 2-1: Signature Algorithms Interoperability Alternatives Assessment

Alternative	Pros	Cons
end systems verification	• scaleable • industry bundling both RSA and DSS with little added cost • technology agility • can work with RSA commercial applications	• Federal users require RSA verification software • RSA users require DSS verification software
multiple certification paths	• scaleable • no impact on RSA users • no interoperability concerns for using RSA based commercial applications	• some added infrastructure cost • users and CMAs in the Federal PKI need multiple certificates • will require FIPS modification or FIPS waivers • Federal users require RSA signature and verification software • Federal user and software must determine which certificate to sign with
trusted gateway	• gateway can be used to facilitate interoperation in other areas	• requires building an additional secure system (gateway) • may not be scaleable • no standards • no COTS • may be very costly • must be trusted by the Federal PKI domain and other domains • may dilute non-repudiation

		• may require mandatory FIPS waiver for the gateway • may have to issue RSA certificates for commercial application users
RSA	• interoperable with commercial sector • single certificate • single cryptographic algorithm on end systems • no interoperability concerns for commercial applications based on RSA • Favorable patent climate (no patent dispute, RSA patent expires in 2000 and is only limited to US. Schnorr patent expires in 2008 and covers North America and Europe)	• will require a change to FIPS • will not interoperate with DSS or other technology based hierarchies • will have significant cost and schedule impact on MISSI • will lock Federal PKI into a single technology

The most appropriate solution for the Federal Government seems to be allowing the end systems to verify multiple algorithms (i.e. DSS and RSA). This approach requires no significant changes to current standards, while allowing a high degree of interoperation. Under this recommended approach, the end systems in the Federal domain and RSA domains will require both verification suites (i.e., RSA and DSS.), but the incremental cost per user workstation may be small or insignificant.

There is some risk that industry will not embrace DSS in terms of usage and cryptographic products. In order reduce the risk and enhance the viability of the end systems approach, the Government should take the following steps:

- **Work towards resolving the patent dispute:** Resolution of the dispute may make licensing the DSS technology attractive from a cost viewpoint.
- **Provide free implementation of DSS**
- **Implement DSS based applications:** The Federal Government should begin using the digital signature technology. The demand will help build the supply of DSS based COTS applications
- **Support CAPI standards:** The Federal Government should participate in developing CAPI standards. The Federal Government should encourage integration of cryptography in applications using CAPI, facilitating easy integration

and replacement of cryptosystems (e.g., the ability to substitute RSA cryptosystem with DSS and vice versa).

If an agreement on end systems verification is not forthcoming, the next most attractive option is issuing multiple certification paths, and should be explored as a fallback option.

2.3 Key Distribution Algorithms

Before we can determine the nature of interoperability issues, we need to know what the FIPS will be in this area. The following alternatives are examined for FIPS standards in this area:

- **Diffie Hellman:** Key distribution using Diffie Hellman. There are several Diffie Hellman key distribution alternatives. Appendix B describes one such alternative using DSS public key parameters.[7]
- **KEA:** This algorithm is used by MISSI and therefore was considered.
- **RSA:** This algorithm was considered because of its widespread use in the commercial sector.

KEA would be a good choice for key distribution as the FIPS since MISSI uses KEA. However, the classification of KEA rules out software implementations. Informal discussions with the Federal users have revealed that not all users are likely to invest in hardware based cryptography. Thus, KEA may not be a viable FIPS alternative. While recommendation for FIPS in this area is beyond the scope of this paper, the following analysis is offered between Diffie Hellman and RSA.

Table 2-2: FIPS Key Distribution Algorithm Options

Algorithm	Advantages	Disadvantages
Diffie Hellman	• license expires in 1997 • perfect forward secrecy efficient[8] • both parties can contribute to the secret easily	• no one standard • few products
RSA	• PKCS Standard • wide-spread commercial products	• license expires in 2000 • perfect forward secrecy highly inefficient[9]

[7] The issue of whether DSS parameters are the same as ones used in a signature certificate is a matter of policy and user decision. The point of this paper is to illustrate the use of DSS parameters for carrying out key exchange.

[8] Requires computing exponent of a random number.

		• no standard on how both parties contribute to a secret

If the commercial installed base, product availability, and product popularity is emphasized, RSA may be the key distribution FIPS algorithm. If there is a preference for discrete logarithms based systems (as evidenced by DSS), Diffie Hellman may be the key distribution FIPS algorithm. Regardless of the FIPS algorithm, interoperability must be further studied. Even if RSA were to be become FIPS for key distribution, Federal users may have to conduct key exchange with enclaves or users using Diffie Hellman. The following approaches were considered for key distribution interoperability:

- One algorithm certificate-- Under this approach a Federal user will have key distribution certificate(s) for one algorithm, be it RSA or Diffie Hellman. Since KEA is not likely to be a FIPS algorithm (see discussion above), a MISSI user will require both the KEA and another algorithm certificate.
- Multiple algorithm certificates -- Under this approach a Federal user will have certificates for algorithms of their choice. Normally, the algorithms will be determined on the basis of the domains the Federal user wishes to communicate with.
- Use a trusted gateway
- Assume universal Diffie Hellman
- Assume universal RSA

Each of these options is described and then analyzed below.

The key pair for signature and encryption must be different, to allow for escrow of the private key used for decryption or for key distribution.[10] The escrow requirements for encryption keys will depend on the law-enforcement requirements, organizational needs and individual needs. The issue of key escrow system interoperability is considered beyond the scope of this paper.

Due to cost and availability reasons, Federal users may use commercial applications (e.g. S-HTTP) that employ the RSA cryptosystem. If these applications require the installation of an infrastructure to facilitate key management, the Federal PKI should be designed to support such key management requirements.

---

[9] Requires generation of entire public, private key pair.

[10] There is no current or envisioned law-enforcement, organizational or personal requirements to escrow signature keys of users. CMAs may wish to backup their signature keys to facilitate easy recovery when keys are destroyed. Backup capability will eliminate the need to promulgate a new key due to accidental destruction of an old key. This is especially helpful for the PAA.

2.3.1 One Algorithm Certificate

Under this approach, a Federal user will have a key distribution certificate for one algorithm, be it RSA, or Diffie Hellman (DH). One way to provide key distribution between a Diffie Hellman user and an RSA user is to allow the RSA user to carry out Diffie Hellman key distribution (Appendix B illustrates one such approach). The RSA user need not have a Diffie Hellman public key or associated parameters. He just needs the requisite Diffie Hellman key distribution software. The Diffie Hellman user can use the public key of the RSA user to encrypt the session key.

The appropriate CMA in the Federal PKI will issue a certificate to the appropriate RSA CMA. The Federal CMA will sign the RSA public key of the RSA CMA using the Federal CMA's DSS private key. Similarly, the RSA CMA will sign the Federal CMA's DSS public key using the RSA CMA's RSA private key. This will help provide a chain of trust to verify the public keys of the end entities across the two domains.

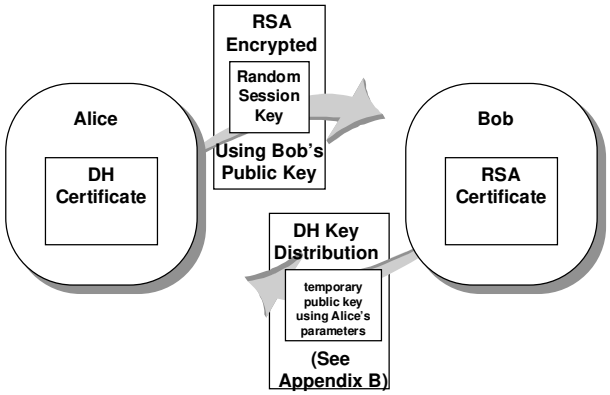

Figure 2-5 -- Key Distribution using Single Algorithm Certificate

For example, if Alice, a Diffie Hellman user, is communicating with Bob, an RSA user, Alice would create a random session key and encrypt it using Bob's RSA public key (see Figure 2-5). If Bob wants to communicate with Alice, he will use Alice's Diffie Hellman public key parameters to create a shared secret as illustrated in Appendix B. Since there is no FIPS in the area of key distribution, either Alice or Bob could be the Federal user in this scenario. Under this approach, the message sender must choose the key distribution algorithm based on the recipient key distribution public key certificate. Furthermore, in order for the KEA to be used, both the sender and the receiver must have KEA certificates.

Table 2-3: Key Distribution Algorithms Based on Single Certificate

	Message Recipient		
Message Initiator	**KEA**	**DH**	**RSA**

KEA	KEA	DH	RSA
DH	DH	DH	RSA
RSA	DH	DH	RSA

Let us now consider the scenario of Federal use of commercial RSA applications requiring key management infrastructure. There are four possible scenarios vis-à-vis PKI:

1. There is no public key management requirement. Users generate and maintain public keys locally and do not need to share the public keys with others.

2. The public key management is embedded with the application.

3. The application assumes existence of an infrastructure for key management to distribute the public keys. The application does not interact with the infrastructure. It assumes some other local software (e.g., user agent) will interact with the infrastructure.

4. The application assumes existence of an infrastructure for key management to distribute the public keys. The application interacts with the infrastructure using embedded capabilities. This means that the application parses certificates and CRLs, builds certification paths, and validates those paths. These functions become quite complex for X.509 v3 certificates once validity, revocation, name constraint, policy constraints, etc. are factored in.

Of the above four scenarios, 1 and 2 do not affect PKI. Scenario 3 affects PKI, but the end system approach makes integration seamless -- the user (entity) RSA key is certified using the DSS key of the CMA. An end system should be able to process such a certificate. The fourth approach is not likely to happen. Vendors are not likely to develop applications that include the capability to interact with an external PKI, whereas the first three scenarios are more likely. Standards are being developed in the area of Cryptographic Applications Programming Interface (CAPI). Applications that integrate cryptography using these standards should be able to easily replace or augment an RSA-based cryptosystem with a Diffie Hellman based system. Thus, properly built software may help mitigate the fourth situation. Hence, COTS RSA applications will be able to interoperate with the PKI under the end systems approach.

2.3.2 Multiple Algorithm Certificates

Under this alternative, a Federal user will have choice of certificates for multiple algorithms. He can choose these algorithms based on the preferences of the communities (domains) he interacts with. For example, a Federal user can have both an RSA and a Diffie Hellman pubic key certificate. The situation is different from the one for signature

algorithm in one subtle, but important manner. In the signature algorithm case, multiple certificate were issued not just to the users, but to the entire hierarchy. Thus, in was really multiple certification paths. Here only the end entities will be given multiple key distribution certificates. The certificates will be signed using the parent CMA's signature algorithm, if the end system approach is chosen for signature algorithm interoperability. The certificates will be signed by the CMA using the complementary signature algorithm if the multiple certificate approach is chosen for signature algorithm interoperability. In other words, RSA key exchange certificate will be signed using the RSA algorithm and Diffie Hellman key exchange certificate will be signed using the DSS algorithm.

The appropriate CMA in the Federal PKI will issue a certificate to the appropriate CMA in other domain and vice versa. This will help provide a chain of trust to verify the public keys of the end entities across the two domains.

Under this alternative, when Alice, a Federal user, wishes to communicate with Bob, an RSA user, Alice would create a random session key and encrypt it using Bob's RSA public key (see Figure 2-6). If Bob wants to communicate with Alice, he would create a random session key and encrypt it using Alice's RSA public key. When Alice and Eve wish to communicate with each other, Eve being a Diffie Hellman user, they would carry out Diffie Hellman key exchange.

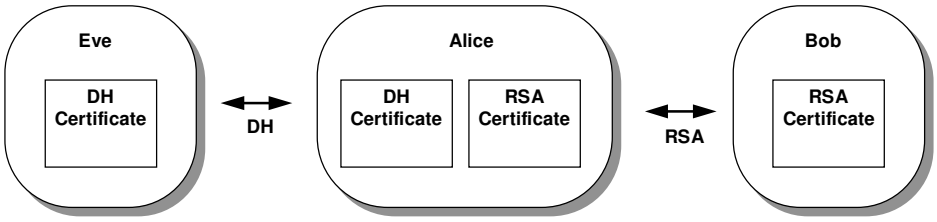

Figure 2-6 -- Key Distribution using Multiple Algorithm Certificates

If the Federal government employs commercial RSA applications requiring key management infrastructure, under this approach, Federal users can register their RSA key distribution keys. The analysis is similar to one presented in Section 2.3.1 if the signature algorithm interoperability is achieved using end systems approach. If the signature algorithm interoperability is achieved using multiple certificates approach, RSA key distribution keys can be certified using the RSA hierarchy.

2.3.3 Trusted Gateways

Under the trusted gateway alternative, the Federal PKI will deploy a gateway trusted by the entire Federal user community. The gateway will maintain RSA, Diffie Hellman, and KEA public key certificates to carry out key exchange with the users using these key exchange protocols. The gateway will have appropriate signature certificates from the Federal PKI and other PKIs. The number and type of signature certificates will depend on

the choice made for signature algorithm interoperability. If the end systems approach is chosen, the Federal PKI will issue a DSS certificate to the gateway. If the multiple certification path approach is chosen, the Federal PKI will issue an RSA as well as a DSS certificate to the gateway.

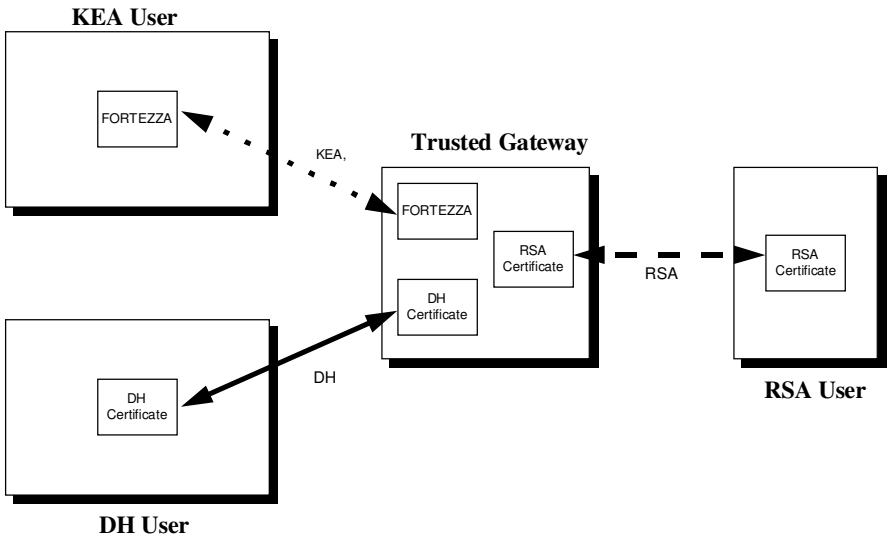

Figure 2-7 -- Trusted Gateway Key Distribution

There are several scenarios how a gateway may operate. Figure 2-7 above illustrates one such scenario. If Bob, an RSA user, wants to send encrypted messages to Alice, a Diffie Hellman user, he uses the RSA encryption feature to send the message encrypting key to the Federal PKI gateway. (see Figure 2-7) The gateway decrypts the key. The gateway carries out Diffie Hellman (or KEA if Alice is a FORTEZZA user) key distribution with Alice. The gateway also needs to provide a computed value to "exclusive or" with the computed secret to arrive at the original message encrypting key. The computed value can be sent in the clear.

If Alice wants to communicate with Bob, she carries out Diffie Hellman (or KEA if Alice is a FORTEZZA user) with the gateway. The gateway uses Bob's RSA public key to encrypt the Diffie Hellman shared secret.

In addition to the above considerations, key distribution should support and interoperate with any existing Federal applications. If there is Federal use of commercial RSA applications requiring key management infrastructure, there are four possible scenarios vis-à-vis PKI:

1. There is no public key management requirement. Users generate and maintain public keys locally and do not need to share the public keys with others.

2. The public key management is embedded with the application.

3. The application assumes existence of an infrastructure for key management to distribute the public keys. The application does not interact with the infrastructure. It assumes some other local software (e.g., user agent) will interact with the infrastructure.

4. The application assumes existence of an infrastructure for key management to distribute the public keys. The application interacts with the infrastructure using embedded capabilities. This means that the application parses certificates and CRLs, builds certification paths, and validates those paths. These functions become quite complex for X.509 v3 certificates once validity, revocation, name constraint, policy constraints, etc. are factored in.

Of the above four scenarios, 1 and 2 do not affect PKI. Scenario 3 affects PKI. The trusted gateway can sign the RSA public key of the user (entity). The fourth approach is not likely to happen. Vendors are not likely to develop applications that also include the capability to interact with an external PKI, whereas the first three scenarios are more likely. Standards are being developed in the area of Cryptographic Applications Programming Interface (CAPI). Applications that integrate cryptography using these standards should be able to easily replace or augment RSA based cryptosystem with a DSS based one. Thus, properly built software may help mitigate the fourth situation also. The fourth scenario can also be accommodated by the gateway issuing the RSA certificates Hence, COTS RSA applications will be able to interoperate with the PKI under the trusted gateway approach. But, in two of the scenarios, the gateway will be required to issue RSA certificates.

2.3.4 Universal Diffie Hellman Adoption

Under this approach, the commercial sector will be convinced to use Diffie Hellman for key distribution. This is highly unlikely given the installed base of RSA within the commercial sector. If this scenario came to pass, the Federal PKI will not have to worry about interoperability in terms of key distribution algorithm. Thus, this alternative is not analyzed any further.

2.3.5 Universal RSA Adoption

Under this approach, the Federal Government and rest of the world will make RSA the FIPS for key distribution.

This approach will facilitate use of commercial RSA based applications with no interoperability concerns.

2.3.6 Alternatives Summary

Table 2-4 provides a summary of the assessment of the alternatives. The universal Diffie Hellman adoption is not analyzed since it is unlikely that it will become a universal standard for key distribution.

Table 2-4: Key Distribution Algorithms Interoperability Alternatives Assessment

Alternative	Pros	Cons
one algorithm certificate	• scaleable • industry bundling both RSA and DH with little added cost • technology agility • can work with RSA commercial applications	• Users may require multiple algorithms on their workstations
multiple algorithm certificates	• scaleable • no interoperability concerns for using RSA based commercial applications	• Federal users may need multiple certificates • Federal users may require multiple algorithms on their workstations
trusted gateway	• gateway can be used to facilitate interoperation in other areas	• requires building an additional secure system (gateway) • may not be scaleable • no standards • no COTS • may be very costly • may dilute privacy • must be trusted by the Federal PKI domain and other domains
RSA	• interoperable with commercial sector • single certificate • single cryptographic algorithm on end systems • no interoperability concerns for commercial applications based on RSA	• will not interoperate with Diffie Hellman • will lock Federal PKI into a single technology

The one algorithm certificate approach may require implementation of multiple key distribution algorithm software on Federal user workstations and user workstations in

other domains, since the selected algorithm is determined by the recipient certificate (see Table 2-3 above).

The multiple algorithm certificates approach is likely to add little to the PKI costs in terms of key generation and certificate storage for multiple algorithms since the approach only requires issuing multiple certificates to the end entities (e.g., users).

Scaleability is not a major issue with the multiple algorithm certificates approach since it only increases the number of user certificates linearly based on number of algorithms. Under this alternative, only the end systems in the Federal PKI domain may require multiple key distribution algorithms. Thus, it imposes no additional requirements on the other domains.

Design and operation of trusted gateways to handle message traffic is likely to be costly and may not be scaleable. Security will be a minus for the trusted gateway approach due to the need to build another trusted system. The end user trust and privacy may be impacted due to the introduction of an intermediary in every transaction. There are no standards for trusted gateways.

The assumption of universal RSA adoption for key distribution may be unrealistic and may lock the Federal Government in one technology. It will also inhibit interoperability with DH users, however small.

Multiple algorithm certificates seems to be the most appropriate solution. Under the recommended approach, a Federal user may obtain both RSA and DH certificates based on his needs.

The Federal Government should also encourage the integration of cryptography in applications using CAPI, facilitating interoperation with applications using RSA.

2.4 Data Encryption Algorithms

The following options were considered for symmetric message (data) encryption:

- Date Encryption Standard (DES)
- SKIPJACK
- International Data Encryption Algorithm (IDEA)
- Triple DES
- Rivest's Code number 4 (RC4)
- Rivest's Code number 5 (RC5)

MISSI uses the SKIPJACK algorithm. Since SKIPJACK is a classified algorithm, it must be implemented in hardware. The risk tradeoffs (threat vs. cost of countermeasures) may not warrant a hardware solution for all Federal users. Thus, SKIPJACK is not appropriate as a Government-wide mandatory standard. The rest of the Federal community may use both software and hardware implementation of DES which is a mandatory FIPS for protection of sensitive-but-unclassified data. The commercial acceptance of DES and large number of products and installed base will also facilitate interoperation. Thus, the Federal PKI will use DES. FORTEZZA users may communicate using SKIPJACK, but they will also require DES to interoperate with rest of the Federal users and with non-Federal users.

For the sake of completeness, it is noted that IDEA is 128 bit key, block cipher used in Pretty Good Privacy (PGP). RC4 is an RSA proprietary cipher. RC5 is an RSA proprietary variable byte key (1-255), block cipher. Triple DES uses two 56 DES keys and carries out three rounds of the DES cryptographic function. To encipher a message, the sender encrypts using the first key; the result is decrypted using the second key, and that result is in turn encrypted using the first key. The outcome is sent to the recipient. The recipient performs the following three steps to recover the plaintext: decrypt using the first key, encrypt using the second key, and decrypt using the first key.

2.5 Data Formats

The Federal PKI will use the proposed version 3 certificate and version 2 CRL formats, as modified per U.S. recommendations, when it becomes a standard. The ongoing efforts to reconcile ISO, CCITT, and ANSI standards in this area are supported by the U.S. Government.

2.5.1 Certificate Format

Currently, MISSI uses version 1 certificates and makes non-standard use of the subject public key information field. When MISSI moves to version 3, it will make standard-based use of all fields, including the subject public key information. Subject public key information field may contain subject public key parameters. Inclusion of these parameters is optional. This could result in interoperability problem. If a certificate contains the parameters, the certificate processing software may use this field. But, if a certificate does not contain the parameters, the X.509 standard is silent on how these parameters are obtained. A natural choice would be to default to the last parameter set in the certificate chain processing.

The Federal PKI will use version 3 X.509 certificates. The following analysis represents the proposed use of version 3 extensions. Table 2-5 provides a summary of recommended version 3 X.509 certificate extensions for the Federal PKI.

The Authority Key Identifier field is useful to differentiate between multiple CMA keys during key rollover period. MISSI calls for this field to contain the Authority Key Management Identifier (KMID). The Federal PKI will use key hash, KMID, or least significant n bits of the public key as the key identifier. The final decision will be made shortly.

MISSI plans to use the KMID as the key identifier subfield within the Key Attribute field. The Federal PKI will use key hash, KMID, or least significant n bits of the public key as the key identifier. The final decision will be made shortly. MISSI does not use the key usage subfield within the Primary Key Attribute field. The reason for this is that MISSI certificate object identifiers are different for different types of public keys (e.g., signature, key exchange, etc.). This is a non-standard technique for defining a key usage field. It is recommended that the Federal PKI use the key usage field.

Use of the KMID as key identifier by MISSI will not inhibit interoperation with other PKIs with version 3 compliant certificate processing software, since the MISSI's use of the field is consistent with the version 3 standard.

MISSI plans to provide a policy identifier in the Certificate Policies and Key Usage Restriction fields. The Federal PKI should do the same.

MISSI plans to hold the Partitioned Rule Based Access Control (PRBAC) information in the Subject Directory Attribute field. MISSI plans to perform access control on transmission and receipt of messages based on the PRBAC information. In order for the PRBAC concept to be adopted Federal Government-wide, there needs to an agreement in classification and marking of Sensitive But Unclassified (SBU) information. Furthermore, there needs to be an agreement that CMAs will also have SBU classification authority Government-wide. That may be unlikely. Thus, the rest of the Federal Government may or may not put the access control information in the subject directory attribute field. This will not hinder interoperability since the PRBAC attribute is represented by a MISSI registered object identifier. MISSI end user software must be modified to allow transmission and receipt of messages to the users in the Federal PKI and outside the Federal PKI domain who do not possess any PRBAC. Such transmission and receipt could be limited to purely unclassified messages.

The Federal PKI should use the subject type subfield in the basic constraints field as recommended by the standard and as used by MISSI. The Federal PKI and MISSI should use the subtrees constraint subfield for name space constraint. This is further discussed in Section 2.8, Naming.

MISSI plans to use policy mapping for cross-certification at the PAA level only. The Federal PKI should do the same. Thus, the Federal PKI will certify CMAs from other domains at the PAA level only.

Table 2-5: Summary of version 3 Certificate Extensions

version 3 Extension	Critical?	MISSI	Recommended for Federal PKI
authority key id	No	KMID	key identifier
(key attributes) - key id - key usage - private key usage period	No	 KMID None Yes	 key identifier Yes Yes
(certificate policies) - policy ids - policy qualifiers	No	 Policy ids None	 Policy ids None
(key usage restriction) - policy ids - key usage	Yes	 Policy ids Yes	 Policy ids Yes
(policy mapping) - issuer policy sets - subject policy ids	No	 Policy set Policy id	 Policy set Policy id
(supported algorithms: directory attribute) - algorithm - usage - usage restrictions (policies and qualifiers)	No	None	None
subject alter. name	No	None	None
issuer alter. name	No	None	None
subject dir. attributes	No (make optional, US comment)	PRBAC, mandatory	None
(basic constraints) - subject type - path length constr. - subtrees constraint	Optional	Critical Yes Yes Yes	Critical Yes Yes Yes
(name constraints) - policy set - name space constr. - name subord. constr.	Critical	 No No No	 No No No

Policy Constraint - policy set - require exp. policy - inhibit policy mapping	Critical	 No No Yes, except PAA	 No No Yes, except PAA

2.5.2 Certificate Revocation List Format

This section contains analysis and recommendation for the version 2 CRL extensions, which are then summarized in Table 2-6.

Authority Key Identifier: The Authority Key Identifier field is useful to differentiate between multiple CMA keys during key rollover period. This field will have the same syntax as the one for the certificate.

CRL Sequence Number: The inclusion of this field is recommended because it helps detect suppression of emergency CRL.

Reason code for CRL entry: The inclusion of this field is recommended because it helps determine the nature and severity of compromise.

Invalidity date: The inclusion of this field is recommended because it will help determine the last time the private key was valid or the suspected date and time of its loss or compromise.

To illustrate the value of these fields, let us say Alice received a message from Bob on 1/1/95. Alice verifies the signature on the message on 2/1/95. Bob's public key certificate is on the revocation list on 2/1/95. The reason code is 3 (affiliation changed) and the invalidity date is 1/15/95. Alice now knows that she can use the certificate to verify the message because Bob's change in affiliation occurred after he signed the message to Alice. This reasoning can be carried out by Alice interactively, or by Alice's user agent software. However, it may be important for Alice's user agent to alert Alice that Bob's certificate was revoked subsequent to the signing of the message.

The version 2 delta CRL extensions can be used to identify distribution points for CRLs issued for a certificate. The extensions can be used to further refine the distribution points based on reason code. Independently, the extensions can be used to issue a delta CRL from a baseline CRL. In other words, if the changes indicated in the delta CRL are applied to the baseline, the result will be an absolute CRL defined by the CRL number in the delta CRL. The distribution point and delta CRLs are used to improve CRL processing performance by breaking a potentially large single CRL in to smaller multiple CRLs. Thus, when a user retrieves a relevant small CRL, the processing burden on the

end user is reduced. As described later, the use of distribution points is recommended for the Federal PKI for the key compromise notification.

The Federal PKI would use the reason code to issue the compromise notifications discussed in Section 2.6, Data Dissemination.

The following table summarizes the extensions that the Federal PKI will use from the proposed version 2 CRL:

Table 2-6: version 2 CRL Basic Extensions

Extension	Critical?	Federal PKI Recommendation
Authority Key ID	No	use it, helps when issuer has multiple keys
CRL Number	No	Use it, helps detect suppression
Reason Code	No	Use it, user is better informed
Hold Expiration Date	No	Do not use it, I am unsure of the value of an automated hold instruction, especially for applications. I can however see their value for an interactive user.
Hole instruction	No	Do not use it
Invalidity Date	No	Use it, keeps the user informed

2.6 Data Dissemination

In this section, the following issue are considered:

- certification dissemination
- revocation notification
- directory interoperation

2.6.1 Certificate Dissemination

MISSI offers two mechanisms for certificate dissemination:

1. certificate path in the message, and
2. X.500 directory.

The Federal PKI will use both of these approaches.

2.6.2 Revocation Notification

MISSI offers revocation notification through the CRL mechanism. The CRLs are provided through the X.500 directory. The Federal PKI will provide X.500 based CRLs.

2.6.2.1 MISSI CKL Design

MISSI also offers a Compromised Key List (CKL) service. Users notify the CA of compromises. The CAs notify the PCAs of their as well their user compromises. PCAs periodically (every two weeks) issue a consolidated CKL for their domain. A PCA pushes the list to the CAs in its domain. The CAs push the list to the users in their domains. Thus, the same CKL is shared within a PCA domain. Optionally, a PCA may share its CKL with other PCAs. In that case, the PCA receiving PCAs consolidate the received CKL with the one generated in their domain. MISSI also uses the CKL to prohibit communication with a PCA domain. The CKL always grows. Once a KMID is on a CKL, it never leaves the CKL.

2.6.2.2 Alternatives for Compromise Notification

The MISSI CKL was designed when version 2 CRL and associated reason code were not available. For the Federal PKI, the compromise notification will be issued using the CRL reason code = key compromise. Even with this scenario, there are several alternatives. They are derived from the following three independent attributes:

1. The first attribute is the issuer. It could be the certificate issuing CA or an ancestor of issuing CA (e.g., a PCA).

2. The second attribute relates to the dissemination technique. The key compromise CRL could be pushed to the end entities or it could be pulled by the end entities, just like certificates and other CRLs, using the X.500 directory model.

3. The third attribute relates to the distribution point for the CRL. It could be trusted or untrusted.

Using the above attributes, the MISSI CKL will be described as the following: issuer: issuing CA parent (PCA), dissemination: push, and distribution point is trusted.

2.6.2.3 Recommendation for the Federal PKI

Since the X.509 standard does not allow CRL issuance by entities other than the certificate issuer, and all the properties of the MISSI CKL can be achieved using the X.509 standard, it is recommended that the Federal PKI use the key compromise CRL issuer = certificate issuing CA. It is recommended that the dissemination technique be the X.500 pull approach. In terms of the distribution point, each domain could decide

whether it wants to offer a trusted distribution point[11] or an untrusted distribution point. The use of a trusted distribution point will allow for immediate notification, but it will require the end entities to request the CRL during each signature check. Similar effect can be achieved using a very short CRL issuance cycle for reason code = key compromise. MISSI CKL was designed to keep the user agents simple and to reduce the directory access. Similar affect can be achieved if all the CAs in a PCA domain use the same distribution point for their key compromise CRLs and the end entities pull all these CRLs based on an event (such as system boot) or periodically.

In summary, the Federal PKI will use the version 2 CRL extensions to achieve timely notification of key compromises.

2.6.3 Directory

MISSI uses an X.500 directory with Directory Access Protocol (DAP) for Directory User Agent (DUA) - Directory Service Agent (DSA) communication and Directory System Protocol (DSP) for DSA - DSA communication. DUA - DSA communication requires strong authentication (digital signatures) during binding. The DSAs support chaining, but the DUA must be able to process referrals. DSA - DSA communication requires strong authentication during binding. All information request transactions are unauthenticated. All update transactions require strong authentication. The directory implementation is consistent with current X.500 standard. The standard allows for strong authentication, but does not provide a full specification.[12] This problem is supposed to be rectified in the next draft standard.

The MISSI Directory System (DS) also calls for access by non-authenticated entities for retrieval only in order to provide interoperability [14]. Suggested options to implement this capability are:

- **Partitioned Directory:** This option entails creating special DSAs. The DSAs will contain information whose disclosure has no risk. These DSAs will not be able to access the MISSI DMS directory. The users would be required to pass simple (password) authentication.

- **Anonymous Access:** Under this option, a portion of the Directory Information Tree (DIT) would be accessible by anyone without any authentication.

[11] The X.509 and X.500 standards currently do not have provisions for the concept of trusted distribution point. A trusted distribution point will require some features to associate and secure the response to particular request as explained in Appendix A of this report.

[12] The standard specifies ASN.1 notation for the signature field, but not for the algorithm identifier.

- **Firewall DUAs:** Under this concept, several DUAs will be able to access a partitioned or main DMS directory. They would use simple authentication for their users and represent them as a single subject class. The DUAs are called firewall DUAs and can be disabled in case of real or suspected security compromise. Main DMS directory would use access control lists to limit retrieval of information by the firewall DUAs.

MISSI type 2 certificates and CRLs have unique object identifiers. Unfortunately, these object identifiers are different from certificate and CRL object identifiers specified by X.500. There are three options for fixing this problem:

- have MISSI move to standard object identifiers
- in addition to MISSI unique object identifiers, have MISSI issue its certificates and CRLs under standard object identifiers
- make other users query directories using knowledge of MISSI object identifiers

In order to promote interoperability, prevent proliferation of object identifiers, and prevent complicated DUAs, it is recommended that MISSI use one of the first two options. Since both options will provide interoperability, it is left up to MISSI to decide which one to choose.

It is assumed that the Federal PKI will use the above constraints on its directory. In summary, the Federal PKI directory will observe the following constraints:

- register certificates and CRLs under standard X.500 attributes
  - a superior CMA will store a subordinate CMA certificate under "CA certificate" attribute
  - superior and subordinate will store certificates issued to each other under "cross certificate pair" attribute

In terms of authentication, the Federal PKI requirements are that the entities by strongly authenticated when updating the directory entries and entry attributes such as the X.509 certificate and the X.509 CRL. The Federal PKI requirements in terms of retrieval for these two attributes is no authentication. These finding are being coordinated with the GSA personnel responsible for Federal X.500 directory and with MISSI. This will help develop the solutions for MISSI and Federal PKI directory interoperation with taking into account the MISSI need for strong authentication.

2.7 Policy

The CMAs certificate policy helps users of the certificate decide on the degree of confidence to place in a certificate. A lot of factors go into making that decision, including the CA policy and procedures for authentication of end entities, CA operating policy, procedures and security controls, end entity policy and procedures for handling private keys, etc. Even the liability assumed by the certificate issuers and end entities play a role in the degree of trust. If a CA or end entity is willing to assume liability for certain situations, a user of the certificate need not be worried about consequences of those situations. Appendix E describes a hierarchical taxonomy to be used by the Federal PKI in formulating the certificate policies.

The Federal PKI will support several policies, based on the needs of the agencies. The version 3 X.509 certificate specification calls for the policy identifiers to be registered based on the International Organization for Standardization/Internet Electrotechnical Committee (ISO/IEC Joint Technical Committee 1) 9834-1 object hierarchy. The Federal PKI will observe these constraints and will include the appropriate policy identifiers in the certificates. These policies will form the basis for cross-certification and policy mapping (based on equivalencies in the cross-certificates) between the Federal PKI domain and other PKI domains.

For a list of initial Federal policies, see [4].

2.8 Naming

The Federal PKI will use X.500 distinguished names for all certificates and CRLs. The Federal PKI will generally observe name subordination. Typically, the CMAs will be named as subordinates to the organizations they issue certificates for. Thus, the CMAs will normally use the subordinate-to-CA's-superior name constraint in issuing certificates. In some cases, the PCAs may allocate non-overlapping name spaces to subordinate CAs. For example, the DoD PCA may allow NSA to issue certificates for the entire DoD. In addition, the PAA may provide a set of name spaces for a PCA to administer. This will be implemented using the basic constraints field in the version 3 certificates. Under this feature a set of subtrees can be allocated to a CMA, for example the set of subtrees @c=US@u=U.S.Federal Govt.@ou=Justice and @c=US@o=U.S. Federal Govt.@ou=Treasury would allow a single PCA to issue certificates to Treasury and Justice employees. Thus, this feature can be used to permit a single PCA to issue certificates to users in multiple cabinet level departments and agencies.

The names of the current Federal PKI elements are:

PAA: @C = US@O = U.S. Federal Govt.@CN = MISSI Type 2 PAA
DoD PCA: @C = US@O = U.S. Federal Govt.@OU = DoD@CN = DoD PCA
GSA PCA: @C = US@O = U.S. Federal Govt.@OU = GSA@CN = GSA PCA

NSA CA: @C = US@O = U.S. Federal Govt.@OU = DoD@OU = NSA@CN = MISSI Type 2 NSA CA
GSA CA: @C = US@O = U.S. Federal Govt.@OU = GSA@OU = GSA@CN = MISSI Type 2 GSA CA

Under the proposed Federal PKI, the names will change to:

PAA: @C = US@O = U.S. Federal Govt.@OU = US Federal PAA
DoD PCA: @C = US@O = U.S. Federal Govt.@OU = DoD@OU = DoD PCA
GSA PCA: @C = US@O = U.S. Federal Govt.@OU = GSA@OU = GSA PCA
NSA CA: @C = US@O = U.S. Federal Govt.@OU = DoD@OU = NSA@OU = MISSI Type 2 NSA CA
GSA CA: @C = US@O = U.S. Federal Govt.@OU = GSA@OU = GSA@OU = GSA CA

2.9 MISSI Enhancements

Based on the above analysis, Table 2-7 contains the summary of the enhancements required for the MISSI type 2 to be compliant with the Federal PKI defined above. The table allocates the enhancements to the appropriate MISSI components and interoperability constraint. It is assumed that MISSI has the ability to process version 3 certificate extensions and version 2 CRL extensions.

Table 2-7: MISSI Enhancements for the Federal PKI

	Signature Algorithm	**Key Distribution**	**Encryption**	**Structure**
CMA	PAA needs ability to receive & sign certificates from other domains	Ability to issue DH and RSA key distribution certificates	None	Issue cross certificate to superior
Directory	RSA verification to support binding to RSA based Directory Service Agent	None	None	Populate "CA certificate" attribute and "cross certificate pair" attribute
User Agent	RSA verification	DH, RSA	DES	None

FORTEZZA	None	None	None	None

	Data Format	**Data Dissemination**	**Policy**	**Naming**
CMA	Implement Key usage field. Issue certificate and CRLs under standard object identifiers	None	None	None
DS	None	None	None	None
User Agent	Need to use key usage field is place of certificate object identifiers. Accept standard certificate and CRL object identifiers. Needs to implement a policy for Federal or other PKI certificates with no PRBAC	None	None	None
FORTEZZA	None	None	None	None

SECTION 3 (INTERNET PKI)
INTERNET PKI

The Internet PKI interoperability constraints are described in this section. These constraints are analyzed for interoperation with the Federal PKI. This section will be reanalyzed after the Internet PKI based on version 3 certificate is developed.

3.1 Interoperability Constraints

3.1.1 PKI Structure

The current Internet PKI consists of an Internet Policy Registration Authority (IPRA) at the Massachusetts Institute of Technology (MIT) and three PCAs as illustrated in Figure 3-1. One PCA is located at Trusted Information Systems (TIS) in Glenwood, Maryland, and offers commercial assurance policy. TIS also operates a CA under this PCA, and that CA is also located at TIS. Two additional PCAs are operated by RSA Data Security in Redwood Shores, CA. Both of these PCAs are located in Redwood Shores, CA. One PCA offers persona security policy, i.e., anyone can register under any name (We need to find out if name subordination is required -- Request For Comment (RFC) 1422 [7] notes that "In general, CAs are expected to sign certificates only if the subject DN in the certificate is subordinate to the issuer (CA) DN"). Another one operates as a high assurance PCA. The PAA private key resides in hardware (Bolt, Beranek, and Newman's (BBN's) Safekeyper), under multiperson control.

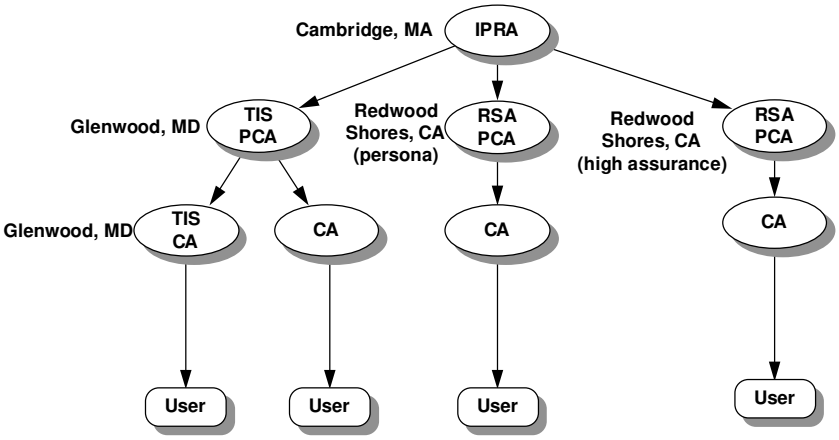

Figure 3-1 -- Internet PKI Structure

3.1.2 Signature Algorithms

The Internet users will employ the MD-2 or MD-5 message digest and RSA signature algorithm.

3.1.3 Key Distribution Algorithms

The Internet users will use RSA encryption for key distribution.

3.1.4 Data Encryption Algorithms

The Internet users will employ DES in Cipher Block Chaining (CBC) mode.

3.1.5 Data Formats

The Internet standard is based on the X.509 version 1 certificates. The certificate includes the following elements shown in figure 3-2:

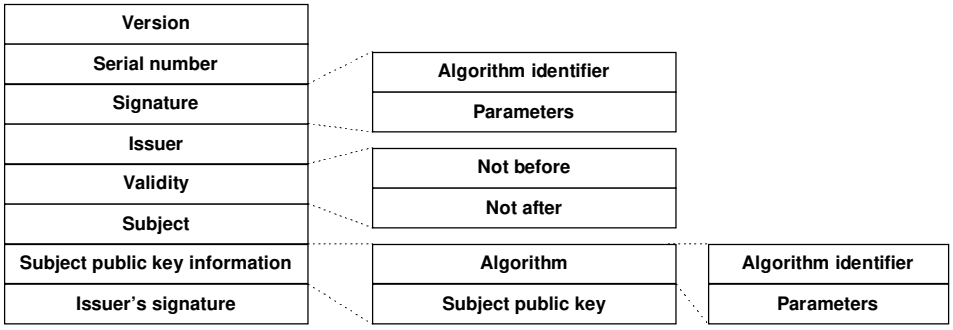

Figure 3-2 -- 1988 CCITT X.509 and PEM Certificate Format

The above represents the standard X.509 1988 (version 1 certificate). Anyone using version 3 certificates should be able to process the Internet certificates.

The Internet CRL contains the elements shown in figure 3-3:

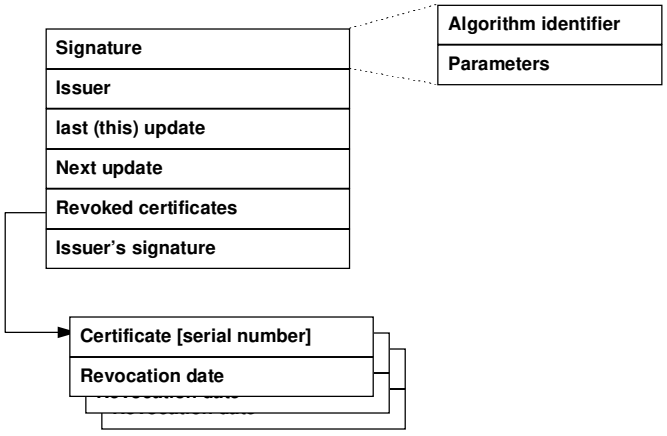

Figure 3-3 -- PEM CRL Format

Anyone using version 2 CRLs should be able to process the Internet CRLs.

3.1.6 Data Dissemination

Initially it was intended that the Internet PKI would offer X.500 directory for certificates and CRLs; however, that hierarchy is not being used. Users are expected to send certificate chains with their messages. The originator certificate field in the PEM specification is used to send certificates in the absence of a directory. Additionally, the issuer certificate field in the PEM allows the signing entity's (CA's) certificate to be included. Multiple instances of this field can specify a whole chain of trust for verification. There are no CRLs.

Privacy Enhance Mail (PEM) specifies a CRL format that can be sent as a special type of PEM message. RFC 1422 [7] states that:

> In the interim [before X.500], the IPRA will coordinate with PCAs to provide a robust database facility which will contain CRLs issued by the IPRA, by PCAs, and by all CAs. Access to this database will be provided through mailboxes maintained by each PCA. Every PEM UA must provide a facility for requesting CRLs from this database using mechanisms defined in RFC 1424 [9]. This IPRA will update monthly a CRL for all the PCAs it certifies, and each PCA (subject to constraints of its individual policy) will maintain a CRL for all the CAs which it certifies.)

However, there appear to be no current implementations of this structure. The TIS PCA and CA plan to provide CRLs using e-mail responders.

3.1.7 Policy

As stated earlier, the Internet supports the following levels of authentication and certificate types:

- persona certificates -- certificates issued with no identity authentication check, allowing anonymous or aliased certificates
- commercial assurance -- to be used with organizational (private and public) and residential (or private individual) certificates
- high assurance

We are in the process of obtaining the details of these policies. We will also attempt to find out whether the Internet plans to register these policies and use the policy identifiers field of version 3 certificates. A document that contains details of the policy of the TIS PCA (the only operational policy we currently have a copy of) is contained in Appendix C.

In the Internet (PEM RFC 1422) each PCA is responsible for defining and publishing an informational RFC describing its certificate issuance policy. This policy must be filed with the IPRA before a certificate is issued to the PCA, and the document will be signed by the

IPRA and made available to users through e-mail. The IPRA and PCAs do not issue user certificates. Only CAs may issue and sign user certificates.

PEM currently specifies RSA parameters and key sizes as follows
exponent e: either 65537 or 3
modulus n: 508 to 1024 bits

3.1.8 Naming

The Internet PKI will use X.500 distinguished names for all certificates and CRLs. As currently drafted, Internet naming policy calls for strict name subordination. We are in the process of finding out the naming policies of the operational PCAs.

The PEM specifies the use of "user@domain-qualified-host" as the distinguished originator and recipient user name. However, these names must be linked to the X.500 DNs in the X.509 certificates.

The names of the current Internet PKI components include:

IPRA: @O = Internet Society@OU = Internet PCA Registration Authority
TIS PCA: @C = US@ST= MD@O=Trusted Information Systems PCA

The TIS PCA does not require PCA-CA name subordination. TIS does not wish to provide CA names, nor does the Internet wish to provide registered PCA names. We have not yet received information from RSA on their PCA and CA.

3.2 Interoperation with Federal PKI

Ideally the Federal PKI would cross certify with the Internet at the PAA to IPRA level. A PAA to IPRA cross certificate would include policy equivalencies between the Federal and Internet PKIs, assuming that the Internet has adopted the new version 3 certificates.

Unfortunately, since so many varied international organizations will potentially be registering with the Internet, this level of certification may prove undesirable or politically infeasible. Figure 3-4 shows a PAA to IPRA cross certification (see Appendix D for the cross-certification process) and its alternative which employs individual cross certifications at the PAA to PCA level. This would be necessary if the Federal PKI wishes to establish policy equivalencies with one PCA (e.g. a British PCA) but does not wish to allow the same policy equivalence with another PCA (e.g. a Cuban PCA).

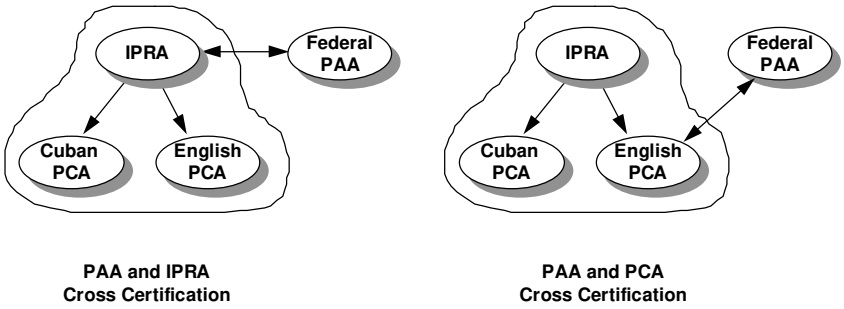

Figure 3-4 -- Federal PKI to Internet Cross-Certification Options

The drawback with PAA to PCA cross certification is an undesirable level of management -- all Internet PCAs would negotiate individually with the Federal PKI to perform cross certification and policy mapping. It would also greatly increase the number of cross certificates.

Assuming that the Federal PKI is capable of authenticating RSA signatures, there aren't any major impediments for Federal and Internet interoperation. Federal users will have the capability to verify RSA signatures, and Internet users will have the capability to verify DSS signatures (see section 2).

Since the Federal PKI uses version 3 certificates and version 2 CRLs, it will be able to process previous versions that the Internet is currently using. For the Internet, however, PEM software will have to be updated to process the version 3 certificates and the version 2 CRLs.

3.3 Federal PKI Enhancements

Many of the enhancements necessary for Federal and Internet PKI interoperation identified above require modifications only to the Internet PKI. The required enhancements that affect the Federal PKI are summarized below:

Table 3-1: Federal PKI Enhancements for Interoperation with the Internet PKI

	Signature Algorithm	**Key Distribution**	**Encryption**	**Structure**
CMA	None	None	None	PAA-IPRA or PAA-PCA cross certify
Directory	None	None	None	None

User Agent	None	None	None	None
FORTEZZA	None	None	None	None

	Data Format	**Data Dissemination**	**Policy**	**Naming**
CMA	None	None	None	None
Directory	None	Ability to authenticate RSA based DUA	None	None
User Agent	None	Ability to obtain CRLs through e-mail	None	None
FORTEZZA	None	None	None	None

The following table summarizes the enhancements required to the Internet PKI for interoperation with the Federal PKI.

Table 3-2: Internet PKI Enhancements for Interoperation with the Federal PKI

	Signature Algorithm	**Key Distribution**	**Encryption**	**Structure**
CMA	None	None	None	PAA-IPRA or PAA-PCA cross certify
Directory	None	None	None	None
User Agent	DSS verification	Diffie Hellman	None	None

	Data Format	**Data Dissemination**	**Policy**	**Naming**
CMA	None	None	None	None
Directory	None	None	None	None
User Agent	Ability to process version 3 certificates	DUA to obtain Federal PKI certificates and CRLs.	None	None

SECTION 4 (CANADIAN FEDERAL PKI)
CANADIAN FEDERAL PKI

The Canadian Government PKI is called Designated/Electronic Commerce Canadian Electronic Key Management System (DCEKMS). It is hierarchical in structure and assumes a Canadian government-wide X.500 directory. The DCEKMS interoperability constraints described in the following section are derived from [10].

4.1 Interoperability Constraints

4.1.1 PKI Structure

The DCEKMS will consist of a hierarchy as illustrated in Figure 4-1 below.

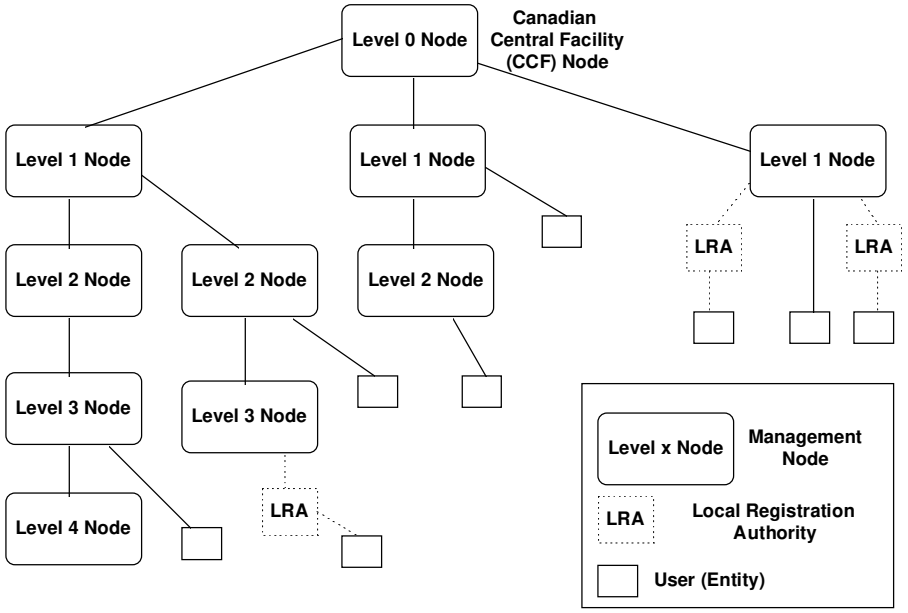

Figure 4-1 -- DCEKMS Structure

The DCEKMS management nodes correspond identically to CMAs in the Federal PKI nomenclature. The Level 0 node, Canadian Central Facility (CCF) is akin to the PAA, level 1 to PCA , and level 2 to CA, Local Registration Authority (LRA) to ORA. In the DCEKMS, the policy constraints of PAA, PCA, and CA hierarchy do not hold. The DCEKMS is being designed based on the version 3 certificates.

The CMAs (management nodes) will be able to specify the policies in the certificates and will not require any PAA, PCA, CA hierarchy from the viewpoint of policy constraints. The CMAs will issue certificates to their superiors. Thus, users will be able to develop a

chain of trust starting at their parent CMA as opposed to the common ancestor. This will increase the certificate path length.

In order to reduce path length and provide each CMA domain the flexibility to choose the name space and policy it wants to honor from another CMA domain, the DCEKMS will allow CMAs at various levels to cross certify each other. The DCEKMS will not limit the lowest level CMA to certifying users -- any level CMA may issue both user and subordinate CMA certificates.

4.1.2 Signature Algorithms

The DCEKMS will support both DSS and RSA. The management nodes will use the multiple certificate approach, i.e., a management node will certify an RSA key using its RSA key and a DSS key using its DSS key.

4.1.3 Key Distribution Algorithms

The DCEKMS plans to support RSA and Diffie Hellman for key distribution.

4.1.4 Data Encryption Algorithms

The DCEKMS plans to support DES. We need to confirm this from CSE.

4.1.5 Data Formats

The DCEKMS will use the proposed version 3 certificate and version 2 CRL formats. The following table summarizes the version 3 X.509 certificate extensions to be used by DCEKMS. Recommended Federal PKI extensions are included for the sake of comparison.

Table 4-1: Summary of DCEKMS version 3 Certificate Extensions

v3 Extension	Critical?	DCEKMS	Recommended for Federal PKI
authority key id	No	Explicit key id or (issuer DN, certificate serial number)	KMID
(key attributes) - key id - key usage - private key usage period	No	 Yes Yes Yes	 KMID Yes None
(certificate policies) - policy ids - policy qualifiers	No	 Policy ids Qualifiers	 Policy ids None

(key usage restriction) - policy ids	Yes	None	Policy ids
(policy mapping) - issuer policy sets - subject policy ids	No	Policy set[13] Policy id	Policy set[14] Policy id
(supported algorithms: directory attribute) - algorithm - usage - usage restrictions (policies and qualifiers)	No	None	None
subject alter. name	No	Yes	None
issuer alter. name	No	Yes	None
subject dir. attributes	No (make optional, US comment)	Yes (Clearance)	None (PRBAC for MISSI)
(basic constraints) - subject type - path length constr. - subtrees constraint	Optional	TBD Yes Yes Yes	Critical Yes Yes Yes
(name constraint) - policy set - name space constr. - name subord. constr.	Critical	Yes Yes Yes	No No Yes
(policy constraint) - policy set - require exp. policy - inhibit policy mapping	TBD (make critical, US comment)	Yes Yes Yes	No No Yes, except PAA

The following table summarizes the version 2 CRL extensions to be used by DCEKMS. Recommended Federal PKI extensions are included for the sake of comparison.

Table 4-2: Summary of DCEKMS version 2 CRL Extensions

v2 Extension	Critical?	DCEKMS	Recommended for Federal PKI
Authority Key ID	No	TBD[15]	Yes
CRL Number	No	Yes	Yes

[13] Limited to the CCF for policy mapping with domains outside DCEKMS.
[14] Limited to the PAA
[15] My draft of DCEKMS does not address this field at all.

Reason Code	No	Yes	Yes
Hold Expiration Date	No	No	No
Instruction Code	No	No	No
Invalidity Date	No	No	Yes
CRL Distribution Point[16]	No	Yes	No
Issuing Distribution Point	Yes	Yes	No
Delta CRL Indicator	Yes	No	No

4.1.6 Data Dissemination

The DCEKMS will offer an X.500 directory for certificates and CRLs. Certificates may also be communicated using the messaging application protocol. In addition, some CMAs may offer trusted directory services. Appendix A further describes the concept of trusted directories.

The Canadian X.500 directory will use simple authentication for certificates and CRLs. The directory will verify signatures on the certificates and CRLs before storing them.

The Canadian Government plans to use the MSP for user - user communication.

4.1.7 Policy

The DCEKMS will support several policies, based on the needs of the agencies. The version 3 X.509 certificate specification calls for the policy identifiers to be registered based on ISO/IEC 9834-1 object hierarchy. The DCEKMS will observe these constraints and will include the appropriate policy identifiers in the certificates. The policy mapping will be performed during cross-certification. Certification paths to external domains should be only considered valid when they pass through the CCF.

4.1.8 Naming

The DCEKMS will use X.500 distinguished names for all certificates and CRLs. It will use subtrees constraints subfield, "subordinate-to-CAs-superior", and "subordinate-to-CA" (also called strict name subordination) for naming.

4.2 Interoperation with the Federal PKI

Interoperation between the U.S. Federal PKI and DCEKMS should be achieved by cross certification between the U.S. Federal PAA and the DCEKMS CCF. The cross certificates should identify the name spaces and policy mappings. No major impediments are seen to the interoperation of the two infrastructures. The next section summarizes the required enhancements.

[16] This field actually appears in a certificate and not in a CRL.

The U.S. DSAs and Canadian DSAs need to be properly chained. The U.S. DSAs should be able to accept simple authentication from Canadian DSAs for information retrieval. In addition, the U.S. DSAs should provide simple authentication (password) when requesting information from Canadian directories. In order to minimize impact on the DUAs in both domains due to differences in authentication mechanisms, the DSAs should not use referrals; the DSAs should strictly chain the request, requiring only DSA interoperation.

All DCEKMS domain users wishing communicate with the U.S. Federal domain users will require DSS verification (for U.S. domain signature verification).

4.3 Federal PKI Enhancements

The required enhancements that affect the Federal PKI are summarized below:

Table 4-3: Federal PKI Enhancements for Interoperation with the DCEKMS

	Signature Algorithm	**Key Distribution**	**Encryption**	**Structure**
CMA	None	None	None	PAA-CCF cross certification
Directory	None	None	None	None
User Agent	None	None	None	None
FORTEZZA	None	None	None	None

	Data Format	**Data Dissemination**	**Policy**	**Naming**
CMA	None	None	None	None
Directory	None	Ability to invoke and accept simple authentication for retrieval	None	None
User Agent	None	None	None	None
FORTEZZA	None	None	None	None

The following table summarizes the enhancements required to the DCEKMS for interoperation with the Federal PKI.

Table 4-4: DCEKMS Enhancements for Interoperation with the Federal PKI

	Signature Algorithm	Key Distribution	Encryption	Structure
CMA	None	None	None	CCF-PAA cross certification
Directory	None	None	None	None
User Agent	DSS verification	None	None	None

	Data Format	Data Dissemination	Policy	Naming
CMA	None	None	None	None
Directory	None	None	None	None
User Agent	None	None	None	None

SECTION 5 (USPS PKI)
UNITED STATES POSTAL SERVICE PKI

The United States Postal Service (USPS) PKI interoperability constraints are described in this section and are then followed by analysis of interoperation with the Federal PKI. The Postal Service PKI consists of a single CA located in San Mateo, CA.

5.1 Interoperability Constraints

The following sections detail the interoperability constraints of the USPS PKI.

5.1.1 PKI Structure

The Postal Service will operate a single CA located in San Mateo, CA. The CA will be contacted directly by users and will issue certificates for organizations, organizational units, and users. All transactions with the USPS CA will be on a fee-for-services basis, including certificate issuance, time stamp service, and trusted directory service. The USPS CA may cross-certify other CAs at a future date.

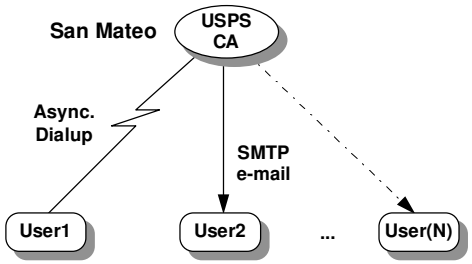

Figure 5-1 -- USPS PKI Structure

5.1.2 Signature Algorithms

The USPS Certificate Authority currently supports only the DSS; however, the Postal Service is currently adding support for RSA signatures using the MD-5 hash algorithm. The Postal Service will maintain two independent hierarchies for DSS and RSA users. The DSS users' public keys will be signed using the Postal CA's DSS private key. The RSA users' public keys will be signed using the Postal CA's RSA private key.

5.1.3 Key Distribution Algorithms

The RSA users will use the RSA encryption feature for key distribution with the Postal CA. The DSS users will use Diffie Hellman as described in Appendix B for key distribution with the Postal CA. The Postal CA, however, does not currently address communication, key distribution, and encryption issues between users. Nor does the

USPS CA address signature and verification issues between users. The Postal Service focus is on users accessing the Postal CA, and the services it provides.

Since the Postal CA will not mandate or support any user-to-user key distribution, it is assumed that the user-to-user key distribution would be as described in Section 2.

5.1.4 Data Encryption Algorithms

The encryption algorithm for communication between the Postal CA and users will be DES. As stated earlier, the Postal CA does not address any user to user communication issues, including data encryption.

5.1.5 Data Formats

The Postal CA will use the proposed version 3 certificate format with the following extensions:

1. Authority Key Identifier: This field will include the serial number of the certificate that the Postal CA issues itself. To make it standard compliant, the Postal Service should (but does not currently) include the Postal CA DN with the serial number.

2. Primary Key Attributes: The Postal Service does not plan to implement the Key Identifier field. The Key Usage field will be implemented. The Postal Service will also implement the private key usage period field.

3. Certificate Policies Field: The Postal Service is planning to use this field to define certificate class (organization authority, organization affiliate, residential, Postal CA, other CA, etc.) and certificate proof level (generic, signed, proxy, certified, biometrics, etc.).

4. The Postal Service would also include Authorizer DN in the certificate. This field is applicable to the organization affiliate and organization authority certificates. It identifies the organization authority that requested than an affiliate or authority be granted a certificate. It is not clear where this DN will go. This is a Postal unique extension for this field.

5. The Postal Service would put some free text in a Postal unique extension.

6. The Postal Service would put a Postal Security field, yet to be determined (e.g., an alternate signature or cryptographic checksum), in a Postal unique extension.

7. The Postal Service is in the process of making decision whether it should have CRLs.

The following table summarizes the version 3 X.509 certificate extensions used by the Postal Service. Recommended Federal PKI extensions are included for the sake of comparison.

Table 5-1: Summary of Postal version 3 Certificate Extensions

v3 Extension	Critical?	Postal	Recommended Federal PKI
authority key id	No	(issuer DN), certificate serial number	KMID
(key attributes) - key id - key usage - private key usage period	No	 No Yes Yes	 KMID Yes None
(certificate policies) - policy ids - policy qualifiers	No	 Cert. type and proofing level	 Policy ids None
(key usage restriction) - policy ids	Yes	 None	 Policy ids
(policy mapping) - issuer policy sets - subject policy ids	No	None	 Policy set Policy id
(supported algorithms: directory attribute) - algorithm - usage - usage restrictions (policies and qualifiers)	No	None	None
subject alter. name	No	None	None
issuer alter. name	No	None	None
subject dir. attributes	No (make optional, US comment)	None	None (PRBAC for MISSI)
(basic constraints) - subject type - path length constr. - subtrees constraint	Optional	None	Critical Yes Yes Yes
(name constraint) - policy set - name space constr.	Critical	None	 No Yes

- name subord. constr.			Yes
(policy constraint) - policy set - require exp. policy - inhibit policy mapping	Critical	None	 No No Yes, except PAA
authorizer DN	-	Postal unique	-
free text	-	Postal unique	-
postal security field	-	Postal unique	-

5.1.6 Data Dissemination

The USPS CA communicates directly with users, and provides a trusted directory service. (See Appendix A for an explanation of trusted directories.) All communications between the users and the CA are via e-mail transactions. Users must digitally sign all transactions they send to the ECS CA, including certificate issuance requests, requests for copies of certificates in the trusted directory, and revocation requests. Similarly, the CA signs and dates all responses returned to the users and links them to the users' original transactions. This ensures that all responses from the CA can be authenticated, are current (not replayed), and are in response to the request originally sent.

The USPS CA does not maintain a user-accessible CRL; however, requests for certificates can ascertain the status of particular certificates. Thus, revoked certificates must be discovered by retrieving a certificate from the CA and noting its current status. Currently, there are no plans to make USPS ECS CA-issued certificates available through an X.500 directory; and there is no USPS CKL. The CA to user communication satisfies needs for directory to user communication, since the Postal CA in essence performs the directory functions.

The Postal CA does not currently address communication, key distribution, and encryption issues between users. The Postal Service focus is on users accessing the Postal CA, and the services it provides.

5.1.7 Policy

The USPS has implemented the following policies for DSS keys.

- common parameters (p, q, and g) for user DSS keys
- p, and hence the public keys, are 1024 bits long
- a private key validity period of 1-2 years (user defined)
- a public key (certificate) validity period of 2-4 years (user defined)

The RSA modulus size is expected to be 512 or 1024 bits, based on the user's choice. Further details on the RSA keys will be added as they become available.

In addition, the USPS has defined several classes of certificates and different levels of identification and authentication requirements for each class. The classes of certificates are:

- Postal ECS -- USPS ECS infrastructure component certificates. e.g. the CA's certificate issuance public key
- Personal -- private citizen or residential certificates
- Organization -- commercial entities that register their organization and affiliates (e.g. employees)
- Cross Certification -- certificates for interoperation with other PKIs

Each issued certificate requires authentication of identity that can be performed at the following levels:

- Biometrics -- Either for use with a device that requires a biometrics reading to employ the private key, or for use in identification of individuals by including the biometrics data in the certificate. {This level can be renewed electronically}
- Certified -- By personally presenting documentation such as a driver's license or passport. {This level can be renewed electronically}
- Proxy Certified -- Where a proxy proves his identity, and provides identity documentation of the certificate requester and an authorization letter from the certificate requester. {This level cannot be renewed}
- Organization -- Where the organization is responsible for the identities of certificate owners within their organization. {This level can be renewed electronically}
- Basic -- Which requires a signed application to be submitted. {This level cannot be renewed}
- Generic -- Generic certificates would not be issued to an identity, rather, they would have an impersonal ID or number, and could be acquired without identification. {This level cannot be renewed}

USPS certificates can be revoked in three manners: a certificate revocation request signed by the corresponding private key will be accepted; organizations may revoke certificates issued to their organization affiliates (e.g. employees); and the USPS CA reserves the right to revoke any certificate they suspect is compromised.

In addition, if a Postal Service key is compromised, USPS policy specifies steps for notification, revocation of compromised certificates, and reissue of user certificates.

The USPS currently specifies a five year policy for retaining audit data and a forty year policy for retaining archival data.

5.1.8 Naming

The Postal CA's distinguished name is: @c=US@o=United States Postal Service@cn=CA01. User certificates need not be subordinate to the Postal CA. Users can register unique organizations and the CA will issue them certificates. The Postal Service will ensure that the organization affiliate certificates are subordinate to the organization authority.

5.2 Interoperation with the Federal PKI

Since the structure of the USPS PKI is flat with the CA communicating directly with all users, (shown in Figure 5-1) cross certification with the Federal PKI is simple. The Federal PAA should treat the USPS CA as a PAA, and establish a normal PAA level cross-certification with the USPS (shown in Figure 5-2). This will allow a certificate chain between private citizens (USPS PKI clients) and Federal users, minimize cross certifications, and maintain the independence of the USPS PKI from the Federal PKI.

The Postal Service CA does not observe any naming constraints. The cross certificate issued to the Postal CA by the Federal PAA will include the following:

- no name constraints
- basic constraints field indicating CA as subject type and path length of zero (0)[17]

The Postal Service CA will issue a cross certificate to the Federal PAA that includes the following:

- subtrees constraint subfield defining the Federal PAA name space
- basic constraints field indicating CA as subject type and path length of 2

Further work is required in defining policy equivalencies for cross certification.

[17] The path length of "0" will imply that the Postal CA can issue only certificates to end entities. This feature will help maintain security in spite of no naming constraints. It will allow the Federal users to ignore certifcates issued by a masquarding user.

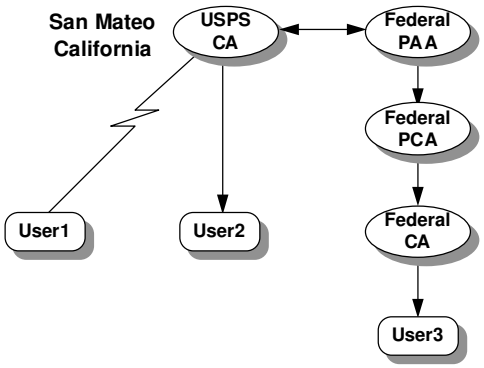

Figure 5-2 -- USPS PKI - Federal PKI Cross Certification

The Federal PKI and the USPS PKI have no compatibility problems with signature algorithms or data encryption algorithms since the USPS will support DSS, RSA, and DES. The Federal PKI's PAA will certify both the DSS and RSA public keys of the Postal CA using the PAA's DSS private key. This will allow the Federal users to verify the certificate chains ending with Postal RSA as well as DSS users. The Postal CA may issue one or two certificates to the Federal PAA. All Postal users will require the DSS verification suite on their workstations. In addition, the Postal users will require the ability to obtain and verify certificate chains. Currently, the Postal user agent only supports verification of transactions from the Postal CA and certificates issued by the Postal CA.

Key distribution will not be an interoperability concern since Federal users will have ability to obtain certificates for multiple key distribution algorithms. It should be noted that the Postal Service plans to offer both RSA and DH anyway.

DES should be used for data encryption among the users. Postal users should be able to accommodate this, since they have DES for encrypted communication with the Postal CA.[18]

The Postal CA will issue version 3 certificates. This will minimize interoperability concerns in the areas of certificate formats. The Postal CA should be encouraged to include the issuer distinguished name in the authority key identifier. This will allow for selection of the appropriate issuer certificate.

Data dissemination is problematic since the USPS does not issue a CRL, and only users with Postal certificates can create signed transactions to query the Postal CA database.

[18] The Postal Service may not support encryption for the time being

There is a need for the Federal users to obtain Postal issued certificates and for the Postal users to obtain and verify the certificate chain from the Federal domain.

The following options were considered for the Federal users to obtain the Postal user certificates and revocation lists:

Postal PKI employs an X.500 directory
Under this approach, the Postal CA or another system will provide an X.500 compliant directory.

Postal PKI provides trusted directory services
Under this approach, the Postal PKI will be able to authenticate the Federal users since the two domains (the Postal PKI and the Federal PKI) are cross-certified. This will require enhancements to the Postal CA software's ability to obtain and verify certificates and CRLs from the Federal X.500 directory. The Postal PKI will provide certificates and certificate status in a manner akin to the services it provides for Postal users. This approach will require entities in the Federal domain to directly query the Postal CA for certificates and their status.

The following suboptions within this alternative are considered:

- User agent querying Postal CA
- Trusted gateway querying Postal CA
- DSA querying Postal CA

If a trusted gateway or DSA queries the Postal CA, the transaction between the user agent and the gateway or the DSA will have to be secured, since the Postal Service offers an on-line secured response. Thus, the easiest solution may be for the user agent to have the ability to query the Postal CA.

Of the above options for Federal users to obtain Postal certificates and revocations, the most appropriate choice is the option that requires a Postal Service X.500 directory.

The following options are analyzed for the Postal users to obtain the Federal user certificates and revocation lists:

Enhance Postal User Agent
The users should be able to query the Federal X.500 directory for certificates and CRLs. If the Postal Service offers an X.500 directory and CRL, this alternative is ideal for Postal users to obtain Postal PKI as well as Federal PKI certificates.

Postal CA Retrieval

The Postal CA can implement a DUA to retrieve certificates and CRLs from the Federal PKI. Since the Postal CA is certified by the Federal PKI, this DUA can be strongly authenticated per the Federal PKI policy. Thus, directory authentication alternatives will not pose an interoperability problem. Within this alternative the following suboptions are possible:

- the Postal CA processes the certificate chain and the CRLs and returns the result to the user
- the Postal user agent processes certificate chain and CRLs

All of these approach are equally feasible from the point of view of the Federal PKI. None of these puts additional demands on the Federal PKI. Thus, the final decision must be made by the Postal PKI. But, from a global interoperability perspective, enhancing Postal user agents seems to be the most appropriate alternative. It will allow Postal users to communicate with other standards compliant infrastructures. Section 5.3 is based on this recommendation.

Future analysis will identify specific policy equivalencies. The Postal PKI should be encouraged to use the policy equivalency field in version 3 certificates.

5.3 Federal PKI Enhancements

The required enhancements that affect the Federal PKI are summarized below:

Table 5-2: Federal PKI Enhancements for Interoperation with the Postal PKI

	Signature Algorithm	**Key Distribution**	**Encryption**	**Structure**
CMA	None	None	None	PAA-Postal CA cross certify
Directory	None	None	None	None
User Agent	None	None	None	None
FORTEZZA	None	None	None	None

	Data Format	**Data Dissemination**	**Policy**	**Naming**
CMA	None	None	None	None
Directory	None	Ability to authenticate RSA DUAs and DSAs	None	None

User Agent	None	None	None	None
FORTEZZA	None	None	None	None

The following table summarizes the enhancements required to the Postal PKI for interoperation with the Federal PKI.

Table 5-3: Postal PKI Enhancements for Interoperation with the Federal PKI

	Signature Algorithm	**Key Distribution**	**Encryption**	**Structure**
CMA	None	None	None	PAA-Postal CA cross certify
Directory	None	None	None	None
User Agent	DSS verification on all user workstations	None	None	Ability to process certificate chain and CRLs

	Data Format	**Data Dissemination**	**Policy**	**Naming**
CMA	policy equivalencies on Federal PAA cross certificate, issuer DN in the issuer key ID field	DUA to populate the directory	None	None
Directory	None	Implement X.500 directory	None	None
User Agent	None	DUA to retrieve certificates and CRLs	None	None

REFERENCES

1. Public Key Infrastructure Study, National Institute of Standards and Technology, April 1994.

2. MISSI Key, Privilege, and Certificate Management Plan, Version 3.0, 6 June, 1995

3. Proposed Federal PKI Concept of Operations, Federal PKI Technical Working Group, 25 April, 1995.

4. Technical Security Policy for the Federal PKI, Federal PKI Technical Working Group, 17 May, 1995.

5. United States Postal Service Support for Electronic Commerce: Draft Security Policy, 30 November, 1994.

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

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

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

9. Kaliski, B., "Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services", RFC 1424, RSA Laboratories, February 1993.

10. Designated/Electronic Commerce Canadian Electronic Key Management System Project Definition, BNR Final Report to Communications Security Establishment, 28 February, 1995.

11. Proposed X.500 Standard, ISO SC21 N9215, July 1995.

12. Ford, W., Computer Communications Security: Principles, Standard Protocols and Techniques, Prentice Hall, New Jersey, 1994.

13. Stewart, W., Proposed X.509 Certificate Extensions for MISSI: Appendix B, 7 June 1995.

14. X.500 Directory Access/Utilization: Non-Messaging/Non-DoD Requirements, Naval Computer and Telecommunications Station, October 1994.

15. Digital Signature Standard, FIPS 186, National Institute of Standards and Technology, December 1993.

16. Action Plan of the NII Federal Information Security Infrastructure, SI-PMO, general Services Administration, 5 June 1995.

LIST OF ACRONYMS

ANSI	American National Standards Institute
API	Application Programming Interface
AT&T	American Telephone and Telegraph
BBN	Bolt, Beranek, and Newman
CA	Certification Authority
CA	California
CAPI	Cryptographic Applications Programming Interface
CBC	Cipher Block Chaining
CCF	Canadian Central Facility
CCITT	Consultative Committee for International Telegraph & Telephone
CKL	Compromised Key List
CMA	Certificate Management Authority
CRL	Certificate Revocation List
COTS	Commercial Off The Shelf
DAP	Directory Access Protocol
DES	Data Encryption Standard
DCEKMS	Designated/Electronic Commerce Canadian Electronic Key Management System
DH	Diffie Hellman
DMS	Defense Messaging System
DN	Distinguished Name
DoD	Department of Defense
DoJ	Department of Justice
DS	Directory System
DSA	Directory Service Agent
DSP	Directory System Protocol
DSS	Digital Signature Standard
DUA	Directory User Agent
ECS	Electronic Commerce Services
EDI	Electronic Data Interchange
FIPS	Federal Information Processing Standard
GSA	General Services Administration

I&A	Identification and Authentication
IEC	Internet Electrotechnical Committee
ID	Identifier
IDEA	International Data Encryption Algorithm
IPRA	Internet Policy Registration Authority
ISO	International Organization for Standardization
KEA	Key Exchange Algorithm
KMID	Key Management Identifier
LRA	Local Registration Authority
MD	Maryland
MD-2	Message Digest 2
MD-5	Message Digest 5
MIC	Message Integrity Code
MISSI	Multilevel Information Systems Security Initiative
MIT	Massachusetts Institute of Technology
MSP	Message Security Protocol
NIST	National Institute of Standards and Technology
NSA	National Security Agency
ORA	Organizational Registration Authority
PAA	Policy Approving Authority
PCA	Policy Creation Authority or Policy Certification Authority or Principal Certification Authority
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PRBAC	Partitioned Rule Based Access Control
RC4	Rivest's Code number 4
RC5	Rivest's Code number 5
RFC	Request For Comment
RSA	Rivest, Shamir, and Adelman
SBU	Sensitive But Unclassified
SHA	Secure Hash Algorithm

SHS	Secure Hash Standard
SI-PMO	Security Infrastructure - Program Management Office
S-HTTP	Secure Hyper Text Transport Protocol
TIS	Trusted Information Systems
TWG	Technical Working Group
US	United States
USPS	United States Postal Service
VA	Virginia

APPENDIX A (Trusted Directory)
TRUSTED DIRECTORY

The trusted directory can be used to obtain the latest status of a certificate. The directory is trusted to verify the certificate against the CRL and real-time compromise notifications. In order to rely on the response of a trusted directory, the directory needs to secure the response to the end user. The end user must be able to rely on the communication security mechanisms provided by the directory to ensure that the response is not modified or is not a replay of a previous response. The user must be able to unambiguously associate the response with a pending request. One way to secure the trusted directory to user communications, even against a replay threat, is to do the following:

- user attaches a unique identifier (ID) to a transaction request to the trusted directory (e.g. requester DN, Date, Time, Sequence number.)
- directory response includes the unique ID. The response is signed by the directory
- user can verify directory signature on the response and reconcile the response with an outstanding request, based on the unique IDs associated with the response and the outstanding requests

Figure A-1 below illustrates the concept.

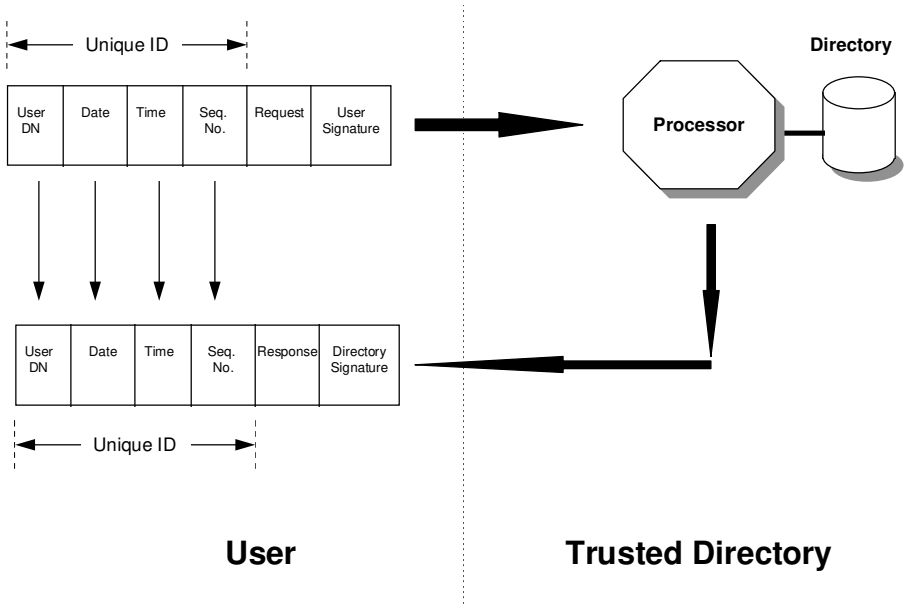

Figure A-1 -- User - Trusted Directory Transaction

In order to ensure the trustworthiness and currency of the directory database, CA to trusted directory communication should be immune to modification, deletion, and insertions. Replay is not a major concern. One way to secure the CA to trusted directory communication against these threats is to do the following:

- CA attaches a unique ID to an update to a trusted directory
- The update package, including the ID, is signed by the CA
- CA sends the package to the directory and puts the unique ID in a local list of pending acknowledgments
- The directory sends a signed acknowledgment that includes the unique ID
- CA reconciles the acknowledgment with the pending acknowledgments list, using the unique ID

Figure A-2 below illustrates the concept

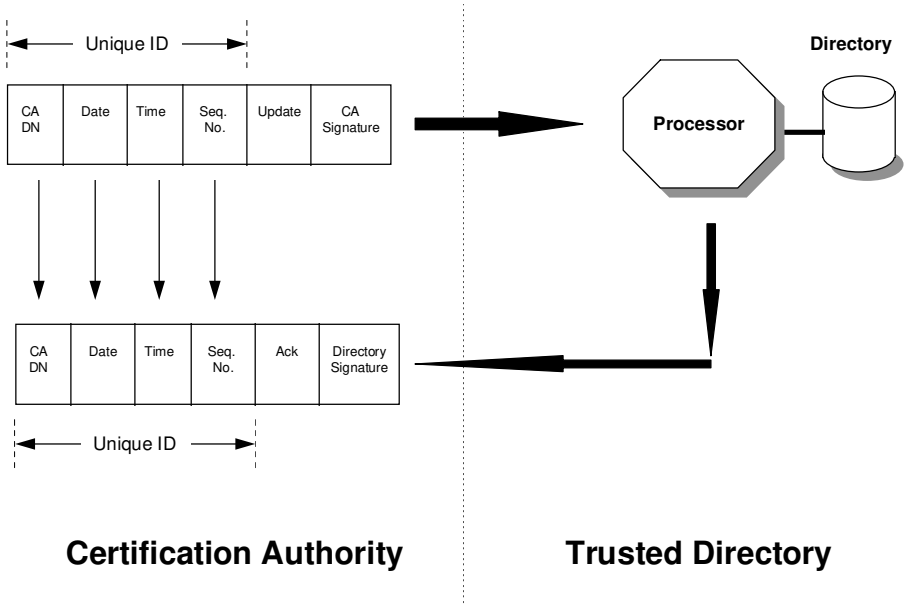

Figure A-2 -- CA - Trusted Directory Transaction

APPENDIX B (Diffie Hellman Key Distribution Using DSS Parameters)
DIFFIE HELLMAN KEY DISTRIBUTION USING DSS PARAMETERS

In this Appendix, we define a key distribution scheme that can be carried out with a DSS certificate owner. Let us say that Alice is a DSS certificate holder, and let her public key parameters be p, q, and g. Furthermore, let us suppose her public key is y and private key is x. Then, we know that

$$y = g^x \bmod p \qquad \{1\}$$

Now consider Bob, a user who wants to communicate with Alice. Bob can obtain Alice's public key information from her public key certificate. Bob then calculates a random number k, such that $k < q$, where q is the small prime for Alice. This becomes Bob's temporary DSS private key. Bob calculates his temporary public key using Alice's parameters, i.e.:

$$y_{temp} = g^k \bmod p \qquad \{2\}$$

Bob also calculates the shared secret using the following steps:

$$y_1 = y^k \bmod p = g^{x*k} \bmod p \text{ (see Lemma for proof)} \qquad \{3\}$$
$$y_2 = \text{SHA}(y_1) \qquad \{4\}$$
$$y_3 = (\text{low 80 bits of } y_2) \oplus (\text{high 80 bits of } y_2) \qquad \{5\}$$

Bob now sends y_{temp} to Alice. Bob digitally signs the transaction using his signature private key. Please note that it is not important whether Bob is an RSA user, a DSS user with different parameters than Alice, or a DSS user with the same parameters as Alice. This scheme will always work and provide a random session key. Alice now verifies Bob's digital signature and computes the following:

$$y_4 = y_{temp}^{\ x} \bmod p = g^{k*x} \bmod p \text{ (see Lemma for proof)} = y_1 \qquad \{6\}$$
$$y_2 = \text{SHA}(y_1) \qquad \{7\}$$
$$y_3 = (\text{low 80 bits of } y_2) \oplus (\text{high 80 bits of } y_2) \qquad \{8\}$$

Thus, Alice and Bob can share a secret limited by Alice's large modulus. The steps of calculating y_2 and y_3 are just an attempt to reduce the secret further in a random fashion. y_2 will be 160 bits. A predefined 128 bits of y_2 can be used for IDEA or predefined sets of 56 bits can be used for triple DES. y_3 will be 80 bits. A predefined 56 bits of y_3 can be used for DES.

Figure B-1 below summarizes the key distribution scheme. The scheme presented here can be easily extended to support the following requirements: both parties contribute to the secret, perfect forward secrecy, and/or both parties know each other has the shared secret. It requires Alice to also perform some of the functions Bob performs above: generate a random private key, calculate random public key based on equation 2, and transfer it to Bob in authenticated manner. Both Bob and Alice use these numbers in place to Alice's permanent keys for computing the shared secret.

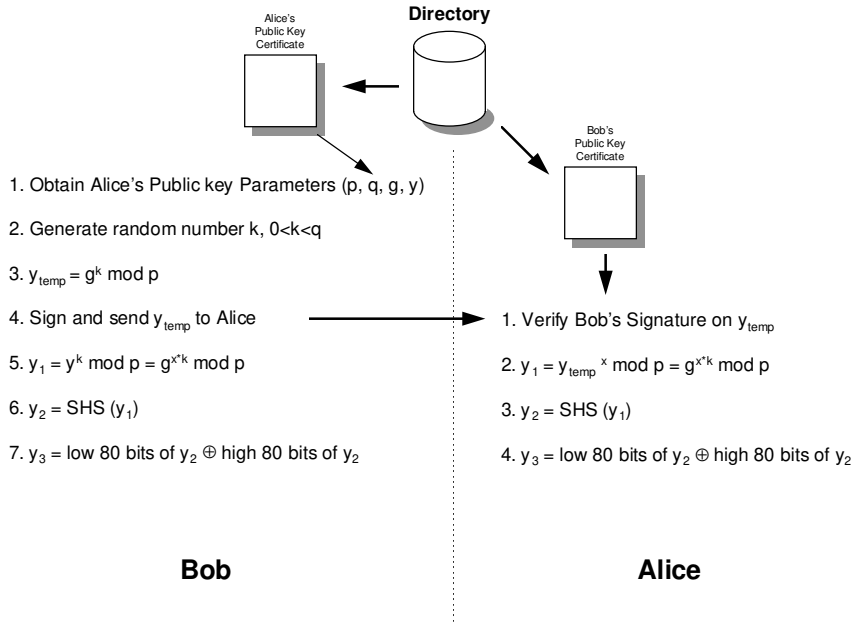

Figure B-1-- Key Distribution Protocol

Alice can always recover (recompute) the key if Alice's private key is escrowed. Bob may need to escrow "k" to recover (recompute) the key.

LEMMA: If $y = g^x \bmod p$, then $y^k \bmod p = g^{x*k} \bmod p$.

Proof:

$$g^x = y + n*p \quad \text{where } n >= 0 \qquad \{9\}$$

$$g^{x*k} \bmod p = (y + n*p)^k \bmod p = y^k \bmod p \qquad \{10\}^{19}$$

IMPLEMENTATION

The implementation of this key exchange protocol requires seven operations (equations {2-8} above). Equation {1} is identical to the public key computation used by the DSS.

[19] Given that n and k are integers, expansion of the polynomial $(y + n*p)^k$ will have only one term (y^k) that is not an integer multiple of p.

Equations {2}, {3}, and {6} can be implemented using the public key generation function of DSS. These equations are also the same as the "r" calculation in the DSS, except "r" requires computation of an additional modulo small prime "q". Equation {4} and {7} are SHA which is part of the DSS. Equations {5} and {8} are "exclusive or" operations.

APPENDIX C (TIS PCA Policy Statement)

TIS Commercial Assurance Policy

The following is excerpted from the information provided by TIS

```
Include the requested distinguished name, a statement to the
effect
that you can speak authoritatively for that distinguished
name,
confirmation that you have read and will abide by our PCA
policy, your
electronic mail address, and your phone number to:

        TIS PCA
        Trusted Information Systems
        3060 Washington Road
        Glenwood, Maryland 21738

Sending us a facsimile (+1 301 854 5363) in addition to
mailing us the
purchase order will expedite matters.

Distinguished Names

The use of distinguished names is new to the Internet at
large and it
is potentially confusing.  There are guidelines, including
RFC 1255
and various North American Directory Forum (NADF) documents,
for
selecting distinguished names.

The template for a corporation might look like the
following:

  Corporation name: Envious Green Computer Corp
  Headquarters: Oceanview, Kansas
  Major subdivision: Cylindrical Systems Group
  Department: Telescopic Systems Department

This, presumably, maps into the Certification Authority (CA)
distinguished name:

  /Country=US
  /State or Province=KS
  /Organization Name=Envious Green Computer Corp
```

/Organizational Unit Name=Cylindrical Systems Group
/Organizational Unit name=Telescopic Systems Department/

with each relative distinguished name (RDN) consisting of a single
attribute value assertion (AVA).

In order to use a distinguished name for your CA, it must be unique
and must faithfully identify the name of your organization. You must
be able to speak for the organization named by the CA distinguished
name.

You can add as many /OU=/ as necessary to identify your organizational
unit within a hierarchy in order to bring the distinguished name's
scope in line with your authority. For example,

/C=US/ST=CA/O=Wesayso Corporation/OU=Tree Pushing Division/OU=Big Tree Pushers/

PCA Policy Statement of Trusted Information Systems, Inc.
December 29, 1993

1. PCA Identity

The business identity and address is

Trusted Information Systems, Inc.
3060 Washington Road (Route 97)
Glenwood, MD 21738
USA

301-854-6889
301-854-5363 (fax)

The e-mail address for correspondence addressed to this PCA is

tispca-info@tis.com

Dr. Stephen D. Crocker is the company officer in charge.

This PCA uses the following distinguished name (DN):

/C=US

/ST=MD
/O=Trusted Information Systems PCA

Whenever the context is clear, Trusted Information Systems PCA will
also be known as TIS-PCA.

This information is effective now and will continue indefinitely.
Adjustments to this information will be made as experience is gained.

2. PCA Scope

TIS-PCA provides certification services to the educational, business,
and government communities worldwide. It is intended for those
organizations who wish to act in the role of Certification Authority
(CA) for the purpose of certifying the binding between a specific user
and his/her public key using identification criteria which are
reasonable and will encourage the use and availability of the public
key technology.

3. PCA Security

It is expected that each CA authorized by TIS-PCA will make a good
faith effort to protect its private key against disclosure. A CA
under the TIS-PCA domain will use reasonable and appropriate hardware,
software, physical, and procedural protection methods to protect its
private key. CAs authorized by TIS-PCA may choose to use special
purpose hardware for signing end-user certificates, but are not
obligated to do so.

TIS-PCA will exercise due care in ascertaining the identity of
organizations and individuals to whom it issues certificates.

4. Certification Policy

TIS-PCA reserves the right to judge the legitimacy and uniqueness of the name used in a certificate and the adequacy of the protection used by the CA. If, in the judgment of TIS-PCA, there is sufficient cause to believe the identity of the certificate holder is incorrectly or ambiguously represented by the distinguished name, TIS-PCA may refuse to issue the requested certificate. If the security of the certificate holder's private key is compromised or the binding between the distinguished name and the public key becomes invalid, TIS-PCA may may revoke the corresponding certificate.

Whenever TIS-PCA issues a residential certificate, the distinguished name in the certificate must unambiguously designate a specific individual. The distinguished name must include an address. The address may be residential, professional or electronic.

CAs authorized by TIS-PCA must institute policies and procedures for issuing certificates which provide reasonable assurance that the distinguished names in certificates it issues correspond to the identity of the certificate holder, and that all distinguished names are subordinate to the CA's distinguished name.

5. CRL Management

TIS-PCA requires that a CA organization issue scheduled CRLs on an advertised regular basis. No requirement is established on the minimum or maximum interval between CRL issuance. CAs are cautioned against overly short or long periods of time between CRL issuance, but

each CA must decide what is appropriate given their environment.

CAs should send new, already signed, CRLs to <tispca-crl@tis.com>.
Such CRLS will be stored in the PCA CRL archive server. CAs are
expected to issue a CRL fetch command shortly thereafter, via the CRL
archive server, to ensure that their CRL was properly received.
TIS-PCA will perform regular backup and system maintenance on the CRL
archive host(s), however the CAs are responsible for ensuring that
their CRLs are reaching the CRL archive server. TIS-PCA reserves the
right to revoke the CA certificate for any CA which fails on an
ongoing basis to make a current CRL (even if empty) available.

6. Naming Conventions

All DNs issued to end-users must be subordinate to the CAs DN. However
it is expected that the enforcement of this provision will be done in
end-user PEM applications.

TIS-PCA reserves the right to inspect the records of each CA authorized by TIS-PCA to check for compliance with the naming rule.

7. Business Issues

CAs operating with certificates issued by TIS-PCA may not have
certificates issued by other PCAs unless the CA is in the process of
transitioning from one PCA to another and has so notified both TIS-PCA
and the other PCA.

If a CA's private key is lost, compromised or otherwise unavailable,
the CA's certificate will be revoked and a new one (with a new public

key) will be signed by the TIS-PCA.

CAs will be issued a certificate by TIS-PCA only upon completion of a licensing agreement and payment of fees. The licensing agreement embodies the policies set forth in this document. The fee schedule may change from time to time, but changes in fees will not apply to license agreements in force.

8. Other

THIS PCA PROVIDES CERTIFICATION SERVICES ON AN "AS-IS" BASIS WITHOUT WARRANTY. USE OF THIS CERTIFICATE SERVICE IS AT THE SOLE RISK OF THE INDIVIDUAL CAs. The only legal agreement necessary between TIS-PCA and any subordinate CAs will in general be limited to affirmation of this PCA's lack of warranty.

Some (or all) public key cryptosystems in the United States are patented. By providing certificate service, TIS-PCA is NOT granting any organization the right to use public key technology. It is the responsibility of each CA to ensure that they are using products which are properly licensed or to otherwise obtain the right to use public key technology. Within the United States and Canada, a UNIX-based reference implementation of a privacy enhanced MH mail system, TIS/PEM, is available from TIS.

APPENDIX D (CROSS CERTIFICATION PROCESS)
CROSS CERTIFICATION PROCESS

This appendix contains the process used by a CMA from one domain to cross certify another CMA from another domain.

The appropriate representatives of the two CMAs will meet face to face to perform one or more of the following exchanges and agreements:

- exchange each other's public key
- establish agreements on liability
- establish agreements on compromise notification
- establish agreements on compromise recovery procedures, including rekeying
- establish agreements on auditing, if at all, each other's operation periodically
- establish agreements on name spaces, i.e. name spaces for they will honor certificates from each other
- establish policy mappings, i.e., agree on which policies in one domain are acceptable as equivalent for policies in another domain and vice versa
- establish rules for directory chaining and authentication protocol for directory access
- sign the agreement and provide each other a copy

APPENDIX E
CERTIFICATE POLICY TAXONOMY

The Federal PKI has defined a taxonomy that specifies the contents of a "certificate policy", the term used in ISO DAM for version 3 certificates under the X.509 Standard. The intent of the certificate policy is to allow the user of the certificate[20] to decide how much trust to place in the binding of the entity's identity and its public key. A lot of factors go into making that decision, including the CA policy and procedures for authentication of end entities, CA operating policy, procedures and security controls, end entity policy and procedures for handling private keys, etc. Even the liability assumed by the certificate issuers and end entities play a role in the degree of trust. If a CA or end entity is willing to assume liability for certain situations, a user of the certificate need not be worried about consequences of those situations.

In summary, a certificate policy should encompass all the factors that may be pertinent to the degree of trust placed in the certificate by the user. Figure E-1 represents the Federal PKI hierarchical certificate policy taxonomy that encompasses the factors pertinent to the degree of trust in a certificate. The following paragraphs describe the taxonomy. An ASN.1 specification for the certificate policy is also provided at the end of this section.

[20] Typically, the user is a signature verifier for signature certificate or key agreement party for key exchange certificate.

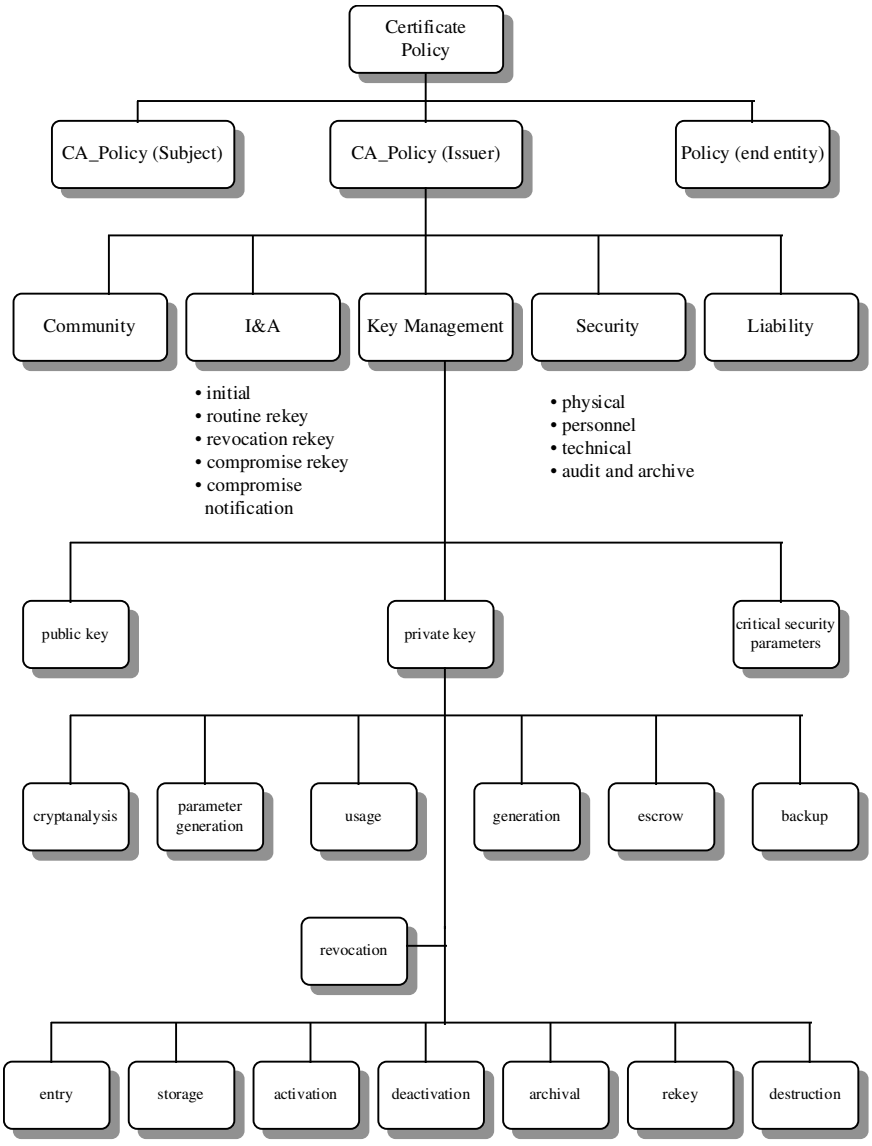

Figure E-1 Hierarchical Policy Taxonomy

At the top level (level 1), the certificate policy consists of three policies, one for issuer CA, one for subject CAs, and one for subject end entities. Of the three policies, the issuer CA policy is mandatory, the subject CA policy is recommended if there is at least one subject CA, and the end entities policy is recommended. The CA policies are composed using the same taxonomy. A CA policy consists of the following five elements: community, identification and authentication policies, key management policies, security policy, and liability policy. The subject end entities policy is composed using the following elements: key management, security, and liability.

E.1 Community

The community element consists of zero or more tuples. Each tuple consists of subject type (CA or end entity), number of subjects, and a brief characterization of the community, e.g., employees, citizens, residents, etc.

E.2 Identification and Authentication

The Identification and Authentication (I&A) policy consists of I&A policy for initial subject registration, I&A policy for routine subject rekey, I&A policy for subject key after revocation such as reorganization, change of address, etc., I&A policy for subject rekey after key compromise, and I&A policy for compromise notification.

Some or all of these I&A policies could be same or different for the subject type (i.e., subject CA or end entity). Thus, there could be up to ten I&A policies. Each of these I&A policies is defined using the same ASN.1 structure. An I&A policy consists of subject type(s), identification policy and authentication policy.

Identification Policy
The identification policy consists of uniqueness flag, meaningfulness flag, and identification technique (e.g., X.500 distinguished name, RFC-822 electronic mail address, URL, etc.).

Authentication Policy
The authentication techniques include, biometrics (thumb print, ten finger print, face, palm, and retina scan), driver's license, passport, credit card, company badge, government badge, articles of incorporation, company seal, and notarized document.

E.3 Key Management

The key management element is further refined into public key, private key, and other critical security parameters such as personal identification number (PIN), password, or private key decryption or activation token. The private key management policy is defined using the following subelement: cryptanalysis, parameter generation, usage, generation, escrow, backup, entry, storage, activation, deactivation, archival, rekey, and destruction. The critical security parameters policy consists of the following elements: generation, escrow, backup, entry, storage, archival, rekey, and destruction. The public key management policy consists of the following elements: archival, rekey and revocation.

Cryptanalysis
This subelement relates to the ability to discover private key and is expressed in terms of MIPS years. For example 200 MIPS years means it will take a machine that computes one million instructions per second, two hundred years to discover the key. It will take a

machine that computes two hundred million instructions per second, one year to discover the key.

Parameter Generation
This subelement consists of several components. A Boolean flag defines whether the parameters are common with other subjects or unique to the subject. A next field defines the FIPS 140-1 level or independent testing of the system that generates the parameters. Another Boolean flag defines if the quality checks are performed on the parameters at the time of certificate generation. If the checks are performed, the entity (issuer CA or ORA) is identified as the quality check agent.

Usage
This subelement defines the usage of the public/private key pair. It has the same syntax and semantics as the key usage field in the standard version 3 extensions.

Generation
This subelement defines how the public private key pair is generated. It contains the entity performing the key generation (issuer, subject, or ORA), media (a Boolean flag to indicate whether generation is performed in hardware or in software), the FIPS 140-1 level or independent testing of the system that generates the key pair, and key control as a pair of integers n, m, where n<=m. The semantics of this field is that the key is split into m shares and any n out of m shares will reconstitute the key.

Escrow, Backup, and Archival
These three subelements follow the same syntax. The structure consists of the action agent (issuer CA, subject, ORA), media, form (plaintext, encrypted, split, checksum), and recipient (issuer CA, subject, ORA, parent organization, commercial third party, or government body).

Entry
This subelement is defined in terms of action agent (issuer CA, subject, ORA), and form (plaintext, encrypted, split, checksum).

Storage
This subelement is defined in terms of media and form.

Activation
This subelement is defined in terms of action agent (subject, issuer, ORA), action to perform activation (login, power on, supply PIN, insert token/key, automatic), and duration of the activation (one time, indefinite, or time period in seconds)

Deactivation

This subelement is defined in terms of action agent, action to perform deactivation (logout, power off, remove token/key, automatic, time expiration).

Rekey
This subelement defines the maximum validity period (in days) for the key. It also defines how the routine rekey is done (manual, certification by old key).

Destruction
This subelement defines how the expired or revoked keys are destroyed. It defines the action agent (issuer CA, subject, ORA), and the action taken (token surrender, token destruction, key overwrite)

Revocation
This subelement defines the public key revocation policy. It consists of CRL issuance frequency in hours, revocation notification grace period in hours, compromise notification grace period in hours, and key loss grace period in hours.

E.4 Security Policy

The security policy consists of the physical, personnel, and technical security controls. The policy also includes the audit and archive requirements.

Physical Controls
This subelement defines the physical controls on the processing system as applicable, e.g., the CA, the end entity system, end entity token etc. It is represented by a bit string for the following (lock and key, physical possession, cypherlock, locked facility, electronic locks, access list, guarded facility)

Personnel Clearances
This subelement defines the clearances granted to the operations personnel. It defines the clearance level (none, sensitive, confidential, secret, top secret) and the clearance granting authority name

Technical Controls
This subelement defines the technical controls on the system. It defines the Computer Security criteria (TCSEC, Canadian, ITSEC, or CC) and the attendant rating as a text string. It also defines the Cryptography Criteria (FIPS 140-1) and the attendant rating as a text string.

Audit Policy
This subelement consists of multiple tuples of audit events and number of days they are kept. Audit events consist of request to create a certificate, request to revoke a

certificate, creation of a certificate, revocation of a certificate, issuance of a certificate, and issuance of a CRL.

Archive Policy
This subelement consists of multiple tuples of archive events and number of years they are kept. Archive events are same as the audit events, i.e., request to create a certificate, request to revoke a certificate, creation of a certificate, revocation of a certificate, issuance of a certificate, and issuance of a CRL.

E.5 Liability Policy

The liability policy consists of zero or more tuples. Each tuple consists of a liability condition and the damages covered. If the liability condition is true, the damage covered are the maximum liability. A liability condition is expressed using predicate logic using the "and" function. The damage is described in dollars.

```
CertificatePolicyDefinitions {??module number??}
-- For example this module could be something like --
-- CertificatePolicyDefinitions {joint-iso-ccitt ds(5) module(1)
certificatePolicyDefinitions(?26?) ?2?}

DEFINITIONS
IMPLICIT TAGS ::=
BEGIN

-- EXPORTS All --

IMPORTS
   KeyUsage, AltNameType
      FROM CertificateExtensions {joint-iso-ccitt ds(5) module(1)
certificateExtensions(26) 2};

ID     ::=    OBJECT IDENTIFIER


CertificatePolicy          ::=    SIGNED {SEQUENCE {
   issuerPolicy            [0]    CAPolicy,
   cAPolicy                [1]    CAPolicy OPTIONAL,
   entityPolicy            [2]    EntityPolicy OPTIONAL, }}

CAPolicy                   ::=    SEQUENCE {
   community                      Community,
   iDandAuth                      IDandAuth,
   keyManagement                  keyManagement,
   security                       Security,
   liability                      Liability }

Community                  ::=    SET OF {CommunityDescription}
CommunityDescription       ::=    SEQUENCE {
   subjectTypes            ::=    SubjectTypes,
   numberOfSubjects        ::=    INTEGER,
   characterization        ::=    Characterization
Characterization           ::=    CHOICE {
   subjects                       Subjects
   SubjectDescription             IA5STRING    }
Subjects                   ::=    BIT STRING   {
   employees               (0),
   citizens                (1),
   residents               (2),
   contractors             (3),
   anyone                  (4)    }
SubjectTypes               ::=    BIT STRING {
   issuerCA                (0),
   endEntity               (1),
   subjectCA               (2),
   oRA(3)                  }

EntityPolicy               ::=    SEQUENCE {
   keyManagement                  keyManagement,
   security                       Security,
   liability                      Liability }


IDandAuth                  ::=    SEQUENCE     {
   initialID                      SEQUENCE OF {IAPolicy},
   rekeyRoutine                   SEQUENCE OF {IAPolicy},
   rekeyRevocation                SEQUENCE OF {IAPolicy},
   rekeyCompromise                SEQUENCE OF {IAPolicy},
```

```
   compromiseNotification SEQUENCE OF {IAPolicy}    }
IAPolicy                  ::=    SEQUENCE     {
   policySubjects                 SubjectTypes, --which groups a policy applies
to
   iDPolicy                       IDPolicy,
   authPolicy                     AuthPolicy   }
IDPolicy                  ::=    SEQUENCE     {
   uniqueNames                    BOOLEAN DEFAULT TRUE, -- whether names are
guarenteed unique
   meaningfulNames                BOOLEAN DEFAULT TRUE, -- whether names are
meaningful to humans
   idNameTechnique                AltNameType  }
--The followin AltNameType would be imported from the X.509 v3 amendments
--AltNameType              ::=    CHOICE {
-- builtinNameForm         ENUMERATED   {
--                         rfc822Name   (1),
--                         dNSName      (2),
--                         x400Address  (3),
--                         directoryName       (4),
--                         ediPartyName (5)    },
-- otherNameForm           OBJECT IDENTIFIER   }
AuthPolicy                ::=    BIT STRING   {
   driversLicense          (0),
   passPort                (1),
   creditCard              (2),
   companyBadge            (3),
   governmentBadge         (4),
   articlesIncorp          (5),
   companySeal             (6),
   notarizedDoc            (7),
   biometricHand           (8), --Dimensions of hand
   biometricThumb          (9), --Single Thumbprint or other single digit print
   biometricFingers        (10), --Set of ten fingerprints
   biometricPalm           (11), --Palmprint
   biometricVoice          (12),
   biometricFace           (13),
   biometricRetina         (14)   }


KeyManagement             ::=    SEQUENCE {
   publicManagement               PublicManagement
   privateManagement              PrivateManagement
   parameterManagement            ParameterManagement }
PrivateManagement         ::=    SEQUENCE {
   cryptanalysis                  INTEGER,     --MIP Years to break using
current methods
   parameterGeneration            ParameterGeneration,
   usage                          KeyUsage,
   generation                     Generation,
   escrow                         Staging,
   backup                         Staging,
   entry                          Entry,
   storage                        Storage,
   activation                     Activation,
   deactivation                   Deactivation,
   archival                       Staging,
   rekey                          Rekey,
   destruction                    Destruction  }
ParameterManagement       ::=    SEQUENCE {
   generation                     Generation,
   escrow                         Staging,
   backup                         Staging,
```

```
   entry                        Entry,
   storage                      Storage,
   archival                     Staging,
   rekey                        Rekey,
   destruction                  Destruction  }
PublicManagement          ::=   SEQUENCE {
   archival                     Staging,
   rekey                        Rekey,
   revocation                   Revocation   }
ParameterGeneration       ::=   SEQUENCE     {
   commonality                  BOOLEAN DEFAULT FALSE -- Whether parameters
are common (not unique)
   generationSystem             GenerationSystem,
   qualityChecks                BOOLEAN DEFAULT FALSE, --Defined quality
checks performed on parameters
   qualityChecker               SubjectTypes }
GenerationSystem          ::=   ENUMERATED   {
   none                         (0),   --No Validation performed
   fIPS140Level1          (1),  --U.S. FIPS 140-1 Level 1 Validation
   fIPS140Level2          (2),  --U.S. FIPS 140-1 Level 2 Validation
   fIPS140Level3          (3),  --U.S. FIPS 140-1 Level 3 Validation
   fIPS140Level4          (4),  --U.S. FIPS 140-1 Level 4 Validation
   independEval           (5),  --Independent Test and Evaluation performed
   standardEval           (6)   }, --Standards body test and evaluation
--The following KeyUsage would be IMPORTed from the X.509 v3 amendments
--KeyUsage                ::=   BIT STRING   {
-- digitalSignature       (0),
-- nonRepudiation         (1),
-- keyEncipherment        (2),
-- dataEncipherment       (3),
-- keyAgreement           (4),
-- keyCertSign            (5),
-- offLineCRLSign         (6)   }
Generation                ::=   SEQUENCE     {
   generator                    SubjectTypes,
   resourceHardware             BOOLEAN DEFAULT FALSE, -- whether hardware
used (not software)
   generationSystem             GenerationSystem,
   splitControls                SEQUENCE {
                                INTEGER      --number of shares required to
reconstitute info
                                INTEGER      }} --number of shares into which
info is split
Staging                   ::=   SEQUENCE     {
   actionAgent                  SubjectTypes,
   mediaHardware                BOOLEAN DEFAULT FALSE, -- Whether stored on
hardware (not software)
   form                         Form
   recipient               BIT STRING   {
                           issuer       (0),
                           subject      (1),
                           ORA          (2),
                           parentOrg    (3),
                           thirdParty   (4),
                           govtBody     (5)    }      }
Form                      ::=   BIT STRING   {
   plaintext               (1),
   encrypted               (2),
   splitKey                (3),
   checksum                (4),
   other                   (5) }
Entry                     ::=   SEQUENCE     {
```

```
   actionAgent                  SubjectTypes,
   form                         Form   }
Storage                 ::=     SEQUENCE     {
   mediaHardware                BOOLEAN DEFAULT FALSE, -- Whether stored on
hardware (not software)
   form                         Form   }
Activation              ::=     SEQUENCE     {
   actionAgent                  SubjectTypes,
   actionRequired           BIT STRING   {
                            login               (0),
                            powerOn             (1),
                            supplyPIN           (2),
                            insertTokenOrKey    (3),
                            automatic           (4)    },
   duration                 CHOICE {
                            ENUMERATED   {
                                  oneTime       (0),
                                  indefinite    (1)    },
                            INTEGER      }      }      --Time period in seconds
Deactivation            ::=     SEQUENCE     {
   actionAgent                  SubjectTypes,
   actionRequired           BIT STRING   {
                            logout              (0),
                            powerOff            (1),
                            timeExpiration      (2),
                            removeTokenOrKey    (3),
                            automatic           (4)    }      }
Rekey                   ::=     SEQUENCE     {
   maxValidity                  INTEGER,     --maximun validity period in
days
   rekeyProcess                 ENUMERATED   {
                                manual              (0),
                                certifiedByOldKey   (1)    }      }
Destruction             ::=     SEQUENCE     {
   actionAgent                  SubjectTypes,
   actionTaken                  BIT STRING   {
                                tokenSurrendered    (0),
                                tokenDestroyed      (1),
                                keyOverwrite        (2)    }      }
Revocation              ::=     SEQUENCE
   cRLFrequency                 INTEGER,     --time between CRL issues in
hours
   compromiseNotification INTEGER,     --max time before notifications issued
in hours
   lostKeyNotification          INTEGER,     --max time before notifications
issued in hours
   revocationNotification INTEGER      }      }--max time before notifications
issued in hours



Security                ::=     SEQUENCE     {
   physicalSecurity             PhysicalSecurity,
   personnelSecurity            PersonnelSecurity,
   technicalSecurity            TechnicalSecurity,
   auditSecurity                AuditSecurity,
   archiveSecurity              ArchiveSecurity     }
PhysicalSecurity        ::=     BIT STRING   {
   lockAndKey           (0),
   physicalPossesion    (1),
   cypherLock           (2),
   lockedFacility       (3),
```

```
   electronicLocks        (4),
   accessList             (5),
   guardedFacility        (6)    }
PersonnelSecurity         ::=    SEQUENCE     {
   clearanceLevel                BIT STRING   {
      none                       (0),
      sensitive           (1),
      confidential        (2),
      secret                     (3),
      topSecret           (4),
      higher                     (5)    },
   clearanceGrantor               CHOICE {
                                  otherName    [0]    TYPE-IDENTIFIER,
                                  rfc822Name   [1]    IA5String,
                                  dNSName      [2]    IA5String,
                                  x400Address  [3]    ORAddress,
                                  directoryName       [4]    Name,
                                  ediPartyName [5]    IA5String    }      }
TechnicalSecurity         ::=    SEQUENCE     {
   computerSecurityCriteria       ENUMERATED   {
                                  tCSEC        (0),
                                  canadian     (1),
                                  ITSEC        (2),
                                  CC           (3)    },
   computerSecurityRating IA5STRING,
   cryptographyRater              GenerationSystem,
   cryptographyRating             IA5STRING    }
AuditSecurity             ::=    SEQUENCE OF {AuditEvents}
AuditEvents               ::=    SEQUENCE     {
   events                         Events,
   timeRetained                   INTEGER      }      --time in days
Events                    ::=    ENUMERATED   {
   requestCreate          (0),
   requestRevoke          (1),
   createCertificate      (2),
   revockCertificate      (3),
   issueCertificate       (4),
   issueCRL               (5)    },

Archive                   ::=    SEQUENCE OF(AuditEvents)


Liability                 ::=    SET OF {LiabilityConstraints}
LiabilityConstraints      ::=    SEQUENCE     {
   liabilityConditions            LiabilityConditions,
   damagesCovered                 DamagesCovered}
DamagesCovered            ::=    INTEGER  -- Damage limit in U.S. Dollars (Not
very international)
LiabilityConditions       ::=    SET OF (Conditions)
-- Conditions to be defined later
Conditions                ::=    NULL


END
```