NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PUBLIC KEY INFRASTRUCTURE (PKI) INTEROPERABILITY: A SECURITY SERVICES APPROACH TO SUPPORT TRANSFER OF TRUST

by

Anthony P. Hansen

September 1999

Thesis Advisor: James Bret Michael
Second Reader: Timothy J. Shimeall

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1999	Master's Thesis

4. TITLE AND SUBTITLE
PUBLIC KEY INFRASTRUCTURE (PKI) INTEROPERABILITY: A SECURITY SERVICES APPROACH TO SUPPORT TRANSFER OF TRUST

5. FUNDING NUMBERS

6. AUTHOR(S)
Hansen, Anthony P.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT

Public key infrastructure (PKI) technology is at a primitive stage characterized by deployment of PKIs that are engineered to support the provision of security services within individual enterprises, and are not able to support the vendor-neutral interoperability necessary for large, heterogeneous organizations such as the United States Federal government. Current efforts to realize interoperability focus on technical compatibility between PKIs. This thesis defines interoperability as the capacity to support trust through retention of security services across PKI domains at a defined level of assurance and examines the elements of PKI interoperability using this more comprehensive approach.

The initial sections discuss the security services PKIs support, the cryptography PKIs employ, the certificate/key management functions PKIs perform, and the architectural elements PKIs require. This provides the framework necessary for discussing interoperability. Next, the two fundamental aspects of interoperability, technical and functional, are presented as well as their constituent elements and the existing barriers to interoperability. Finally, the proposed U.S. Department of Defense and Federal government PKI architectures are analyzed and recommendations made to facilitate interoperability.

14. SUBJECT TERMS
Cryptography, Public Key Infrastructure, Computer Security, Information Warfare Protect

15. NUMBER OF PAGES
167

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Approved for public release; distribution is unlimited

PUBLIC KEY INFRASTRUCTURE (PKI) INTEROPERABILITY: A SECURITY SERVICES APPROACH TO SUPPORT TRANSFER OF TRUST

Anthony P. Hansen
Lieutenant, United States Navy
B.S., University of Notre Dame, 1990

Submitted in partial fulfillment of the
Requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1999

Author: Anthony P. Hansen
Anthony P. Hansen

Approved by: James Bret Michael
James Bret Michael, Thesis Advisor

Timothy J. Shimeall
Timothy J. Shimeall, Second Reader

Dan C Boger
Dan Boger, Dean
Department of Computer and Information Sciences
and Operations

ABSTRACT

Public key infrastructure (PKI) technology is at a primitive stage characterized by deployment of PKIs that are engineered to support the provision of security services within individual enterprises, and are not able to support the vendor-neutral interoperability necessary for large, heterogeneous organizations such as the United States Federal government. Current efforts to realize interoperability focus on technical compatibility between PKIs. This thesis defines interoperability as the capacity to support trust through retention of security services across PKI domains at a defined level of assurance and examines the elements of PKI interoperability using this more comprehensive approach.

The initial sections discuss the security services PKIs support, the cryptography PKIs employ, the certificate/key management functions PKIs perform, and the architectural elements PKIs require. This provides the framework necessary for discussing interoperability. Next, the two fundamental aspects of interoperability, technical and functional, are presented as well as their constituent elements and the existing barriers to interoperability. Finally, the proposed U.S. Department of Defense and Federal government PKI architectures are analyzed and recommendations made to facilitate interoperability.

TABLE OF CONTENTS

LIST OF ACRONYMS AND ABBREVIATIONS

ACES	Access Certificates for Electronic Services
AES	Advanced Encryption Standard
ASD(C3I)	(United States) Assistant Secretary of Defense for Command, Control, Communications, and Intelligence
ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation One
CA	Certification Authority
CCEB	Combined Communications Electronic Board
CCITT	Consultative Committee for International Telegraph & Telephone
CINC	Commander in Chief
CMA	Certificate Management Authority
CMC	Certificate Management Over CMS
CMP	Certificate Management Protocol
CMVP	Cryptographic Module Validation Program
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
COI	Community of Interest
COTS	Commercial Off-The-Shelf
CP	Certificate Policy
CPMWG	Certificate Policy Management Working Group
CPS	Certification Practice Statement
CSE	Communications Security Establishment (of Canada)
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DMS	Defense Message System
DN	Distinguished Name or Directory Name
DNS	Domain Name System (or Service)
DNS Sec	Domain Name System Security
DoD	(United States) Department of Defense
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EE	End Entity
ECA	External Certification Authority
FIPS	(United States) Federal Information Processing Standard
FII	(United States) Federal Information Infrastructure
FPKI	(United States) Federal Public Key Infrastructure
GII	Global Information Infrastructure
GITS	(United States) Government Information Technology Services Board
GOTS	Government Off-The-Shelf
GSA	(United States) General Services Administration
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSEC	Internet Protocol Security
ISO	International Organization for Standardization
ISOC	Internet Society
ITU	International Telecommunications Union
IW	Information Warfare

KEA	Key Exchange Algorithm
KRA	Key Recovery Agent
LDAP	Lightweight Directory Access Protocol
LRA	Local Registration Authority
MIPS	Million Instructions Per Second
MISPC	Minimum Interoperability Specification for PKI Components
MISSI	Multilevel Information Systems Security Initiative
MIT	Massachusetts Institute of Technology
MMP	MISSI Manangent Protocol
NATO	North Atlantic Treaty Organization
NII	(United States) National Information Infrastructure
NIPRNET	Sensitive But Unclassified Internet Protocol Router Network
NIST	(United States) National Institute of Standards and Technology
NSA	(United States) National Security Agency
OCSP	Online Certificate Status Protocol
OID	Object Identifier
ORA	Organizational Registration Authority
PAA	Policy Approval Authority
PCA	Policy Creation Authority
PEM	Privacy Enhanced Mail
PGP	Pretty Good Privacy (Trademarks of Network Associates, Inc.)
PIN	Personal Identification Number
PKCS	Public Key Certificate Standard (RSA, Inc. standard)
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure (X.509)
PMA	Policy Management Authority
POP	Proof of Possession
RA	Registration Authority
RSA	Rivest, Shamir, and Adleman
SDN	Secure Data Network
SDSI	Simple Distributed Security Infrastructure
SIPRNET	Secret Internet Protocol Router Network
SMI	Security Management Infrastructure
S/MIME	Secure Multipurpose Internet Mail Extensions
SRA	Sub-Registration Authority
SSA	(United States) Social Security Administration
SSL	Secure Sockets Layer protocol
TCP	Transmission Control Protocol
TSA	Time Stamping Authority
WWW	World Wide Web

ACKNOWLEDGMENT

The author would like to especially thank Professor Bret Michael for his guidance and encouragement. In addition, Professor Tim Shimeall, Mr. Bill Burr, Mr. Gary Dahlquist, Mrs. Gloria Serrao, and Mr. Al Arsenault provided invaluable advice and direction. Without this assistance, this thesis would not have been possible.

A special thanks to Mary Ellen Hansen, our family, and our many friends for whose love, patience and prayerful support I will be eternally grateful.

I. INTRODUCTION

A. INFORMATION OPERATIONS AND PUBLIC KEY INFRASTRUCTURES

As the global economy rapidly shifts its principle means of wealth production from manufacturing to an information base, distributed computer networks have become increasingly ubiquitous and critical to both our economy and the operation of the U.S. Federal government. Protection of this critical information technology infrastructure was officially recognized on July 15, 1996 by Executive Order 13010 as a vital element of our nation's security. Information Operations (IO), and specifically IO defense, is the natural outgrowth out of this realization.

Protection of the critical information technology infrastructure requires a comprehensive, systems engineering approach. A total system philosophy is required to provide needed security services from the contributing disciplines of administrative security, physical security, personnel security, and information security. Since security can be compromised at any stage of system development, implementation, or use, each of these contributing disciplines must ensure security across the continuum from the strength of their theoretical underpinnings, through system design and implementation, to actual practice. Ultimately, however, perfect security is not possible. Additionally, security resources are always limited. As a result, comprehensive risk assessments must be performed within each security discipline and the results applied to an overall cost-benefit analysis for efficient, secure system development. In order to perform this analysis, the system must be bounded in a defined manner. This is especially important given the degree of interconnectivity

among today's distributed computer networks. These established boundaries define security domains.

Joint IO doctrine recognizes three general domains within the "information environment": the Global Information Infrastructure (GII), the U.S. National Information Infrastructure (NII), and the Defense Information Infrastructure (DII). Each hierarchical information infrastructure is composed of the system of interconnected computers, communications networks, databases, sensors, software, people and other support structures that serve the information and processing needs of the users within the U.S. Department of Defense (DoD), the United States and the world, respectively. The vast size and degree of physical interconnection among these information infrastructures makes their precise demarcation impossible. Rather, they serve as an intellectual construct over which supporting infrastructures may be mapped. (Joint Pub 3-13, 1998) Although these domains are necessarily DoD centric, the replacement of the DoD with any other sub-national organization and the U.S. with any other nation creates a more generic model that could be applied universally.

For the purposes of this thesis, I am going to interject an additional domain, the Federal Information Infrastructure (FII), between the DII and the NII. The FII represents the domain that subsumes the DII and serves the information and processing needs of the entire U.S. Federal government. Figure 1 depicts this DoD-centric, information environment model.

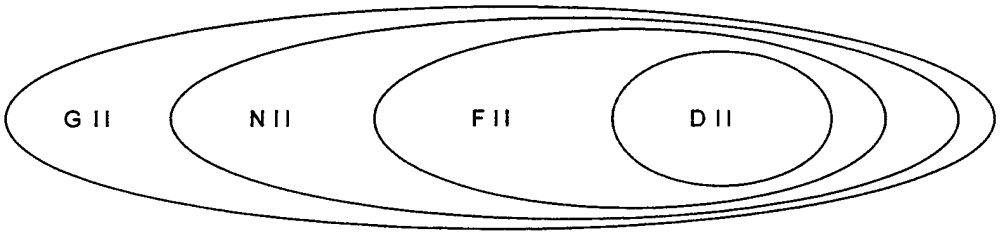

Figure 1. Concentric Domain Information Environment Model

The U.S. Federal government's information security strategy is based upon a "defense-in-depth" concept where multiple security technologies are employed to address various security vulnerabilities. (Hamre, J.J., 1999) The hope is to build overlapping layers of defense providing robust resistance to attack. The more critical the information system, the deeper and stronger the defensive architecture should be designed.

Computers and their users have a physical reality that can be uniquely defined as an identity, but establishing a positive identity, (or conversely creating anonymity) across a network is not a trivial problem. The use of information infrastructures in support of critical applications, however, whether commercial, governmental or military, demands a verifiable link to identity in physical reality in order to establish trust. This is the problem of authentication.

Although symmetric cryptography has long been a key national security technology vital to protecting secrecy*, the advent of public key cryptography promises an attractive general solution to this authentication problem and a means

* As discussed in chapter three, symmetric cryptography can also be used for digital signatures and authentication.

of transferring trust. For example, cryptographic digital signature protocols could provide a bridge between the physical and logical realms if a supporting public key infrastructure (PKI) can be established. Additionally, public key cryptography could make key exchange in support of traditional private key cryptosystems more efficient if a secure PKI can be built. Therefore, creation of a secure PKI is a critical defensive IO technology.

B. THE PRIMATIVE STATE OF PKI TECHNOLOGY

A PKI is the underlying system comprised of the applications, policies, standards and laws that governs the generation, storage, distribution and management of both cryptographic keys and digital certificates. PKIs are designed to support public key cryptography services and exist to propagate trust across computer networks by providing a defined set of security services with an established level of assurance.

PKI technology is still in its infancy. In fact, the current development of public key infrastructures parallels that of another critical infrastructure's development: the development of railroads within the United States in the 1800's.

The first railroads built in the U.S. during the early 1800's were independent, private enterprises, designed and justified exclusively to promote the commercial interests of local communities. Connecting two cities or providing links between natural and artificial waterways, they provided transportation and communication locally and were built in a manner that prevented the easy interchange of freight. During the 1830s, private companies embarked on expensive experiments trying different forms of motive power, cars and track. Gradually, railroads began forming regional networks

and started cooperating to adopt common standards for air brakes, couplers and wheels to permit the interchange of equipment between lines. The principle obstacle to interchange, a uniform gauge, was not overcome until the middle 1880s when the standard gauge was adopted throughout the country. By 1906, the nation's over two hundred independent railroads had been consolidated to the point that two thirds of the nation's total rail mileage was owned by only seven major lines and freight could be interchanged efficiently from coast to coast.

Today, dozens of vendors are racing into the marketplace hawking their proprietary "PKI solutions." Almost universally, they are designed to meet the needs of the "enterprise," a euphemism for a very localized, usually homogeneous, domain. Like the early railroads, their customers justify the expense of these products locally because different vendor's products do not work together. Although standards bodies have been working to address issues of technical interoperability for several years and have made significant progress, essential open standards for the technologies necessary to enable PKI interoperation are still under development. The few PKIs that have been implemented to serve a larger "regional" network, such as the Automotive Network Exchange (ANX), do so with a common vendor architecture, again designed to meet the needs of a very homogeneous user community. (ANX Service Overview, 1999)

Claims of "interoperability" are largely marketing hype that must be very narrowly defined. True interoperability is simply the capacity of a PKI to support trust by retaining its security services across domains at an established level of assurance. Naturally, the trivial case of interoperability occurs when the architecture and policy of both domains is identical. Unfortunately, this trivial

case of "interoperability" is what is sold by vendors' advertising today.

Like the railroads, however, the need to efficiently move "freight" between different vendors is universally recognized by industry, standards bodies, and governments as critical to the widespread use and growth of PKIs. The first step in establishing general interoperability, and the purpose of this thesis, is to define the essential elements of interoperability.

C. LITERATURE REVIEW

1. Significant Literature on Cryptography and Digital Signatures

The general subjects of cryptography and digital signatures are not new and are well developed in the literature. One of the best comprehensive references on the general subject of cryptography is Bruce Schneier's book *Applied Cryptography* which is now in its second edition. (Schneier, B., 1996) Similarly, Warwick Ford and Michael Baum's *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption* is an excellent tutorial on most of the issues associated with digital signatures. (Ford, W. and Baum, M., 1997)

2. Significant Literature on PKIs

The general subject of PKIs is not as mature. Marc Branchaud's Masters of Computer Science thesis at McGill University, Montereal entitled, "A Survey of Public Key Infrastructures" represents an early attempt to characterize the general elements of a PKI. (Branchaud, M., 1997) Branchaud identified two basic PKI operations, certification and validation, whose implementation "is the basic defining

characteristic of all PKIs."* He then developed a matrix of ten basic PKI characteristics that he used to survey four different PKI implementations. Branchaud's characterization of PKIs is focused on how they function, rather than why they exist, the security services they provide.

PKIs are a hot topic today in computer security periodicals and general descriptions of the technology abound. Mark Merkow's recent four part series entitled "Growing a Tree of Trust" published online by *internet.com* provides a good introduction to PKIs that a novice can easily understand. (Merkow, M., December 31, 1998, January 14, 1999, January 28, 1999, and February 18, 1999)

3. Significant Work on Technical Compatibility

Previous works addressing PKI interoperability have focused on their technical compatibility. Compatibility is necessary, but not sufficient for true interoperability. It seeks to ensure that two PKIs can do the same things so as to communicate and build trust paths. Despite almost a decade of effort by industry, governments and standards bodies, compatibility remains a very significant technical challenge today.

The U.S. Federal government, under the auspices of the National Security Agency (NSA), the National Institute of Standards and Technology (NIST), and the Technical Working Group (TWG) of the Federal PKI (FPKI) Steering Committee, has sponsored and published many of these efforts to define

---

* Branchaud defines certification as "the process of binding a public-key value to an individual, organization or other entity, or even to some other piece of information, such as a permission or credential" and validation as "the process of verifying that a certification is still valid." (Branchaud, M., 1997, p. 10)

the elements of technical compatibility in a general PKI. (Fillingham, D., 1996) (Burr, W., *et al.*, 1997) (Chokhani, S., 1997)

FPKI Technical Specification: Part D - Interoperability Profiles, defines the following eight "interoperability constraints":

- PKI Structure
- Signature Algorithms
- Key Distribution Algorithms
- Data Encryption Algorithms
- Data Formats
- Data Dissemination
- Policy
- Naming

(FPKI Technical Specification: Part D - Interoperability Profiles, 1995) It then profiled four PKIs being designed at the time using these constraints. Although this work addresses policy as an element of interoperability, it still focuses primarily on technical compatibility.

4. Comprehensive Vision for Interoperability Does Not Exist

A comprehensive vision of interoperability must move beyond technical compatibility to also address functional interoperability. Functional interoperability seeks to ensure that the technical elements do what is necessary to retain security services between domains. Issues of policy, legislation, and feasibility must be resolved before two domains can go beyond employing compatible protocols to supporting trust at a defined level of assurance. This thesis expands the discussion of interoperability within the U.S. Federal government to embrace this security service focus.

D. RESEARCH GOAL

The purpose of this thesis is to identify and define the elements necessary for vendor-neutral, standards-based PKI interoperability across a large-scale, heterogeneous enterprise such as the U.S. Federal government.

The security services based definition of interoperability presented above will be advanced by defining the two fundamental aspects of interoperability, technical and functional, and delineating their constituent elements. This framework will then be applied to PKIs in the DII and FII to analyze their capacity to support interoperability.

E. ORGANIZATION OF THESIS

Chapters II and III provide the theoretical basis and context necessary to understand PKI interoperability. Chapter II discusses the security and privacy services supported by cryptography while Chapter III is a general primer on cryptography, which the novice to the field will need to understand the functionality of PKIs.

PKIs are the means for certificate and key management. Chapter IV addresses the functional elements of this management and presents the data structures employed: certificates and certificate revocation lists.

Chapter V will detail the architectural elements of a PKI. PKI architecture will be approached first as a set of operational components and secondly as a set of supporting infrastructures. Overall PKI architectures will then be characterized by a set of general design characteristics including interoperability.

Having established the roles, functions and components of a PKI, Chapter VI will present the elements of interoperability by dividing them into two subsets,

technical interoperability and functional interoperability. Finally, Chapter VI will present some of the barriers that stand in the way of interoperability.

Chapter VII contains a discussion of the application of the elements of interoperability to the developing DoD and Federal PKI infrastructures. This chapter will present the various architects, the architectures they are developing, and the capacity of the architectures to support interoperability.

Chapter VIII will summarize my conclusions and recommendations.

II. INFORMATION SECURITY THEORY

A. TRUST

In general, information systems exist for the express purpose of supporting decision making. Claude Shannon, the father of Information Theory, described information abstractly by examining a communications channel between a source and a destination. Shannon's contribution was to define information in terms of the uncertainty on the part of the destination as to what the source would send. If the data sent by the source is known by the destination *a priori*, then no information is communicated. Conversely, when the data communicated over the channel is completely uncertain to the receiver, then the channel transfers only information. (Sloane, N.J.A. and Wyner, A., 1992)

Computers process data. When this data transfers content that is unknown to the user, the data becomes information. Yet information alone is not enough to support decision-making. Questions concerning the information's completeness, accuracy, timeliness and attribution may affect a person's willingness to use information to make decisions. Trust is the subjective assessment of these and related factors by the decision-maker to determine the confidence he will place in the information. When information and trust are united, the fusion results in knowledge. It is important to note that the objective answers to these questions are not germane to knowledge; only the relying party's perceptions and confidence affect trust.

Since decisions are made based upon the knowledge available to the decision-maker, the ability of information

systems to perform their function as decision support tools is predicated upon their capacity to be trusted.

Trust, as a subjective assessment made whenever information is used, is influenced by a variety of factors which vary based upon the person making the trust assessment. Common factors that affect trust include privacy, operational necessity, risk assessment, and security services provided. Public key infrastructures (PKIs) are said to support or transfer trust because they facilitate the provision of security and/or privacy services with an established level of assurance. An information system's user, the relying party, is still free to distrust it in the presence of an environment characterized by risk, even when the information system is supported by the services of a PKI. After all, PKIs do not change the adage that "garbage in" results in "garbage out." PKIs can, however, provide the objective security services that serve as a prerequisite for trust.

B. FUNDAMENTAL SECURITY AND PRIVACY SERVICES

Security services are those functions of an information system that facilitates trust by mitigating risk to a defined level of assurance. Although there is some diversity of terminology within information security theory, the consensus, and international convention defined by the European Community in 1991, defines three interdependent information security characteristics: confidentiality, integrity, and availability. (Brinkley and Schell, 1995, p. 41) Other supporting security services include authentication, access control, non-repudiation, time-date stamping, and key recovery.

Privacy is a related concept that also supports trust. It is often confused with confidentiality because

confidentiality can support privacy. Privacy is the process of making and enforcing agreements about the further dissemination and intended use of information or its source when a party not under the control of the information's owner accesses it. Anonymity is a privacy service.

The nine fundamental privacy and security services presented below are defined by the author for the purpose of this thesis, but these definitions are consistent with their common usage in the information and computer security fields. Whenever possible, synonymous terminology is also presented for completeness.

1. Confidentiality

Confidentiality, sometimes also called secrecy, is that aspect of information security that prevents the unauthorized disclosure of information. A classical example of a mechanism used to provide confidentiality is the opaque envelope sealed to protect a letter. Confidentiality is the traditional security service that is provided electronically by cryptography. Other electronic techniques, such as steganography, may also be used in certain applications to provide confidentiality.

2. Integrity

Integrity is defined as the prevention of unauthorized modification of information. Integrity can protect the accuracy and reliability of information. The use of ink or type that cannot be erased and hand-written signatures are classical mechanisms for providing integrity to a letter. Digital signatures, described in detail below, are a cryptographic means to ensure integrity.

3. Availability

Availability is that characteristic of an information system that prevents the unauthorized withholding of information or resources. Availability is said to be "non-computable" because an algorithm cannot be developed to establish it. (Brinkley and Schell, 1995, p. 44) As a result, cryptography cannot directly support availability.

In fact, cryptography can impair availability because it slows data access and access can be prevented should cryptographic keys be lost, destroyed or altered. Key escrow and recovery are services supported by some cryptosystems that guard against the risks of cryptography to data availability.

4. Authentication

Authentication is the process of establishing the identity and/or verifying the claimed identity of agents. It is important to recognize that an agent in this context is not necessarily a person. In general, an agent may be a program or another computer system. Additionally, authentication must be a two-way street; the system must both authenticate its users and authenticate itself to the users. This is called a "trusted path" since both sides are authenticated. False login screens, for instance, are an attack against authentication because the system is not authenticated to the user.

Means of authentication vary dramatically, each method having a different probability of accurately verifying identity. All authentication systems, however, are probabilistic; no matter how technically sophisticated, none is foolproof. As a stochastic element of an information system, authentication is subject to both Type I error, false positive authentication, and Type II error, false

negative authentication. (Devore, J., 1995, p. 307) Determining the probability of each of the error types, and selecting an authentication algorithm that provides the appropriate level of assurance, is a critical aspect of computer security design.

Authentication methods fall into the classical taxonomy of something the user has, something the user knows, or something the user is. Smart cards, badges, challenge and reply calculators, and dongles are all examples of token-oriented systems. These systems transform authentication into a physical security problem; any user that has a valid token is authenticated. Password schemes, the most common authentication method, are an example of something the user knows. Cryptographic authentication systems are also fundamentally knowledge based. Security within knowledge based authentication systems rests entirely on the confidentiality of the "knowledge," the password, or cryptographic key. Knowledge and possession authentication schemes are general and can be designed for non-human users. Systems that measure a person's physical characteristics, what the person is, fall into the field of biometrics, and therefore are not general. Examples include fingerprint readers, retinal scanners, pressure sensitive signature pads, and various body geometry measurement methods.

Recently this taxonomy has been extended to include "something the user does." The profiling of a person's actions is being used as a means of authentication and means of identifying abnormal usage to trigger audit. The use of a person's typing profile, the rhythm and force of keystrokes, to authenticate a user is an example of something the user does.

This taxonomy is a simplification; most practical systems combine elements to improve security. A smart card

or a retinal scanner, for instance, may also require a password. Finally, recognize that each authentication system is susceptible to two general types of failure, method failure and implementation failure. When designing an authentication scheme, it is important to consider the probability of each of these failures. Digital signatures, therefore, are a general user, knowledge-based authentication system whose confidence must be examined both in terms of the strength of the method and the strength of the implementation.

5. Access Control

Access control is the process of restricting the use of an information system's resources based on assigned rules of privilege. Access control is predicated upon authentication. Privilege rules cannot be enforced without the identification of the entities requesting the use of controlled information system resources.

6. Non-Repudiation

Non-repudiation is the security service that prevents a party to a transaction or communication from falsely denying the nature of their participation in the communication or transaction. Non-repudiation allows for an established means of proving participation and an efficient resolution of disputes. Since disputes may have to be resolved via the legal system, non-repudiation is sensitive to the rules of evidence within a jurisdiction and legislative interpretation. Proof of receipt is a special case of the non-repudiation security service that prevents a party from falsely denying receipt of something he accepted.

7. Time-Date Stamping

Time-date stamping is the security service that assigns a recognized time and date to a data structure to indicate when changes to the data structure's content were frozen. Time-date stamping is characterized by its degree of synchronization with the established time standard and the degree of temporal granularity asserted. Non-repudiation is often predicated upon authentication, integrity, and time-date stamping. Time-date stamping can also support audit requirements for access control.

8. Key Recovery/Key Escrow

Key recovery and key escrow are security services that allow an authorized party to retrieve the cryptographic keys used for data confidentiality and establish access to the plaintext data. Key recovery is a unique security service to cryptographically-provided confidentiality and is established to support data availability in the case of key loss. Key recovery and key escrow do not support digital signature systems since they would undermine the integrity and non-repudiation characteristics of digital signatures.

9. Anonymity

Anonymity is the privacy service that protects the identity or other attributes of an entity from being disclosed during a transaction or communication when the entity does not desire its release.

C. INTERELATIONSHIPS AMONG SECURITY AND PRIVACY SERVICES

Notice that confidentiality, integrity, and availability are defined using the term "unauthorized." How can an information system, such as a computer or computer network, determine who or what is authorized to access data,

modify data or be given access to system resources? The answers to this basic question, authentication and access control, are fundamental to all three characteristics of information security and pervade all aspects of computer security.

Without authentication and access control, the characteristic of availability would be diametrically opposed to those of confidentiality and integrity, and computer system security would not be possible. With authentication and access control, availability can be supported probabilistically, but not guaranteed.

At the theoretical level, computer scientists differentiate between "computable" and "non-computable" characteristics. Confidentiality and integrity, which can be designed absolutely into a system, are said to be "computable." Availability, which cannot be designed absolutely, is said to be "non-computable." For authentication to be computable, algorithms must be developed that allow the computer to establish identity and/or verify the claimed identity of users.

Naturally, authentication and anonymity are diametrically opposed. Identity is either established or it is not disclosed. Non-repudiation and anonymity, however, can be simultaneously supported in systems where privilege is not linked to identity. This is critical to electronic commerce applications that are required to emulate both the anonymous and atomic natures of cash exchanges.

D. SECURITY CHARACTERISTICS OF SIGNATURES

Conventional handwritten signatures are a recognized legal means of establishing authentication and legally transferring trust. A digital signature is a cryptographic

protocol that minimally produces the same security functions as a conventional signature: authentication, integrity, and non-repudiation. Some digital signature protocols are stronger than a regular signature because they provide confidentiality as well.*

Lawyers identify four functions of a conventional handwritten signature that should also be reflected in a digital signature: evidence, approval, ceremony, and efficiency. (American Bar Association, 1998)

The concept of non-repudiation encompasses both evidence and approval. Non-repudiation requires that the signature is attributable to the signer and provides evidence to this fact. Therefore, digital signatures must be very difficult to forge. Secondly, a signature must demonstrate the intent of the signer to give the document legal effect, approval. A signature must not be easily erased or transferred (reused) by either the sender or the receiver after it is affixed. Digital signatures may incorporate a time stamp to support non-repudiation. This is an attempt to create a digital analog to the "mailbox rule".**

Although separated for legal distinction, intent, ceremony, and efficiency are closely interrelated. Ceremony, or the cautionary function, is the solemnity of signing a document that prevents inadvertent signature.

---

* Added confidentiality features extend the metaphor and are frequently referred to as "digital envelopes."

** The "mailbox rule" is the legal principal from common law that is applied to contracts where the parties are not using a substantially instantaneous means of communication. It establishes the legal effectiveness of the contract on the time of dispatch, rather than the time of receipt. (Ford, W. and Baum, M., 1997, p. 41)

Ceremony protects intent. This is an important aspect of implementation since affixing an inadvertent "default" signature on an electronic document seems far more plausible than an inadvertent physical signature. Efficiency is the result of intent and attribution. Efficiency is that characteristic of a signature making it readily identifiable to the general viewer that attribution and intent exists. As a result, efficiency suspends the general viewer's predisposition to question the validity of the document's content and saves time.

Digital signature integrity, prevention against modification of a message after signature, is often referred to as the cryptographic sealing function. This "seal" does not necessarily imply confidentiality, but many protocols provide this additional security too.

III. CRYPTOGRAPHY AND DIGITAL SIGNATURES

A. BACKGROUND

A true understanding of PKIs is not possible without a basic understanding of cryptography. Similarly, understanding digital signatures in their context as cryptographic protocols is a prerequisite to understanding public key certificates, the basis for PKIs. This chapter provides a general overview of cryptography to introduce terms, concepts, and methods that are integral to a PKI; readers familiar with the fields of cryptography and digital signatures are encouraged to go directly to the next chapter.

Cryptography, whose etymological origin comes from the Greek words *kriptos*, meaning "hidden," and *graphos*, meaning "writing," literally means "hidden writing." Since cryptography is an ancient and well-established discipline with an existing vernacular, the reader is directed to the glossary for definitions of terms.

Classically, encryption algorithms involve two fundamental operations, substitution and permutation. Most modern encryption algorithms combine these two functions, and are known as product ciphers. Claude Shannon identified two methods for the hiding of information, "confusion," which is achieved by substitution, and "diffusion," which is achieved by permutation. (Abrams and Podell, 1995, p. 353) Confusion is that characteristic of an encryption algorithm that makes it difficult to predict the effect on the ciphertext of changing a single character of the plaintext. Good confusion is provided by a complex functional

relationship between the plaintext/key pair and the ciphertext, making it difficult to break this relationship. Diffusion is the characteristic of the encryption algorithm that spreads the information contained in the plaintext over the entire ciphertext. Good diffusion makes it necessary for the attacker to intercept a large quantity of the ciphertext in order to infer the algorithm.

Pfleeger cites Shannon's 1949 work, "Communication Theory of Secrecy Systems," as proposing five characteristics of a good cipher:

- The amount of secrecy needed should decide the amount of labor appropriate for the encryption and decryption.
- The set of keys or the enciphering algorithm should be free from complexity.
- The implementation process should be as simple as possible.
- Errors in ciphering should not propagate and cause corruption of further information in the message.
- The size of the enciphered text should be no larger than the text of the original message. (Pfleeger, 1989, pp. 63-64)

The first principle is one of economy; since perfect security is not possible, the value of the information to be protected should dictate the limit to which resources are applied to protect it. The second principle is less straightforward. It states that complexities restricting possible plaintext or making the key too complex, difficult to transmit, store, or remember, should be avoided. The remaining three principles are apparent.

A brute force attack attempts to break the cipher by trying all the possible solutions. The enormous computational power of modern, parallel processing supercomputers or vast networks of personal computers requires that an encryption algorithm be much stronger against a brute force attack. This is especially important when the encryption algorithm itself is public. Public algorithms make the confidentiality of the key the sole source of the system's security. Historically, cryptographic algorithms that depend primarily on the confidentiality of the algorithm itself, rather than the confidentiality of the key, have been easier to break, and are therefore not recommended.

Cryptographers have implemented two approaches to this brute force threat environment, using a one-time pad and basing their algorithms on "hard" mathematical problems.

B. ONE TIME PAD

A one-time pad, a stream cipher with a completely random key stream, is the only cryptographic system mathematically proven to be immune to cryptanalytic attack. This system establishes perfect confidentiality because a one-time pad has perfect confusion; all decryptions are equally likely. As a result, it is impossible for an attacker to statistically identify any plaintext decryption as better than every other decryption. The key for a one-time pad is the key stream; it must be as long as the message to be encrypted. As the name implies, the key can only be used once, since reuse would result in a trivial cryptanalytic problem. The one time pad, however, violates Shannon's second principle; the key is complex. As a

result, its use is limited to special circumstances where privacy is essential, and the overhead of key generation, key length, and key transfer can be justified. Therefore, the one-time pad is not practical for digital signature applications.

C. COMPLEXITY

In order to create enough confusion to thwart a high-speed computer's brute force attack, cryptographers have used mathematically "hard" problems. These are problems whose fastest known solution algorithm runs (is deterministic) in exponential time, or worse, relative to the size of its input. In complexity theory, these problems are said to be NP-complete, and it has been proven that if a deterministic solution can be found for any NP-complete problem in polynomial time relative to the input size, then one can be found for every NP-complete problem. (Pfleeger, 1989, p.81) Despite significant mathematical research in this field during the latter half of this century, no deterministic solution in polynomial time has been found for NP-complete problems. As a result, cryptographers have chosen three problems whose complexity is known to be at least NP-complete, and therefore the problem believed to be intractable, upon which to base the security of their cryptographic algorithms. These are the integer factorization problem (IFP), the discrete logarithm problem (DLP Z_p), and the elliptic curve discrete logarithm problem (ECDLP).

1. Integer Factorization Problem (IFP)

The IFP is the easiest to state: given a composite number n that is the product of two large prime numbers, p

and q, find p and q. This is the problem upon which the Rivest, Shamir and Adleman (RSA) cryptosystem is based. The widespread use of this algorithm has inspired the development of a variety of special and general purpose factoring algorithms, such as the number field sieve. The number field sieve is the fastest known algorithm for factoring integers having at least 120 decimal digits. (Certicom Corp., September 1997) These easily parallelized algorithms have made an attack on RSA more feasible for small key sizes.

2. Discrete Logarithm Problem (DLP Z_p)

The DLP Z_p is more difficult to express. Given a large prime number p, let Z_p denote the set of integers over which addition and multiplication can be performed modulo p: {0, 1, 2, ..., p-1}. There exists a non-zero element of Z_p, n, such that each non-zero element of Z_p can be written as a power of n. Given p, n and another non-zero element of Z_p, m, find the unique integer, l, where $0 <= l <= p-2$, such that $m = n^l \pmod{p}$. The unique integer, l, is known as the discrete logarithm of m to base n. This is the problem upon which the U.S. Government's Digital Signature Standard (DSS), the ElGamel system and the Diffie-Hellman algorithm are based. Implementations of attacks on the DLP Z_p problem lag behind those of IFP. The best known algorithm for solving the DLP Z_p is also the number field sieve. Conceptually, finding logarithms of a k-bit prime modulus p is roughly as difficult as factoring a k-bit composite number n using the number field sieve. (Certicom Corp., September 1997)

3. Elliptic Curve Discrete Logarithm Problem (ECDLP)

The ECDLP is also mathematically complex. If q is a prime power, let F_q denote the finite field containing q elements. Given an elliptic curve, E, defined over the field F_q, a point, p, which is an element of $E(F_q)$ of order n, and a point, r, which is an element of $E(F_q)$, determine the integer, l , where $0 <= l <= n-1$, such that $r = lp$. This is the hard problem upon which a DSS analog, the Elliptic Curve Digital Signature Algorithm (ECDSA), has been developed. The best known attack on ECDLP is the Pollard rho-method. (Certicom Corp., September 1997).

Tables 1, 2, and 3 compare the computing power, measured in millions of instructions per second (MIPS) years, necessary to attack all three "hard" problems. Table 1 provides a recent historical record of the state of the factoring art. It shows that in twelve years the size of the largest factorable integer has almost doubled. Table 2 depicts the computer power necessary to solve the ECDLP for various key sizes. Table 3 gives the same information for either the IFP or the DLP Z_p. Comparing Tables 2 and 3, the developers of ECDSA conclude that ECDLP is an order of magnitude more complex per bit of key size than either the IFP or the DLP Z_p.

Year	Number of decimal digits	Number of bits	MIPS years
1984	71	236	0.1
1988	106	352	140
1993	120	399	825
1994	129	429	5000
1995	119	395	250
1996	130	432	750

Table 1: Historical Data on the Integer Factorization Problem.

Source: (Certicom Corp., September 1997)

Field size (in bits)	Size of *n* (in bits)	MIPS years
163	160	9.6×10^{11}
191	186	7.9×10^{15}
239	234	1.6×10^{23}
359	354	1.5×10^{41}
431	426	1.0×10^{52}

Table 2: Computing Power Required to Compute Elliptic Curve Logarithms with the Pollard Rho-Method.

Source: (Certicom Corp., September 1997)

Size of integer to be factored (in bits)	MIPS years
512	3×10^{4}
768	3×10^{8}
1024	3×10^{11}
1280	3×10^{14}
1536	3×10^{16}
2048	3×10^{20}

Table 3: Computing Power Required to Factor Integers Using the General Number Field Sieve.

Source: (Certicom Corp., September 1997)

D. SINGLE (PRIVATE) KEY CRYPTOGRAPHY

In general, cryptosystems can be modeled using the following simple mathematical notation:

Let: A = Alice (the standard "A" sender in cryptographic literature)

B = Bob (the standard "B" receiver in cryptographic literature)

P = Plaintext message

C = Ciphertext message

E_k = Encryption key

D_k = Decryption key

E = Encryption function

D = Decryption function

Then: $C = E(E_k, P)$ {encryption}

$P = D(D_k, C) = D(D_k, E(E_k, P))$ {decryption}

In conventional cryptosystems, also known as private key, single key or symmetric cryptosystems, $E_k = D_k$.* Although this simple model uses an equal sign to denote equality between the encryption key with the decryption key, in practice this equation is not necessarily identical. Any cryptosystem where the encryption key, E_k, and decryption

* A private key cryptosystem should not to be confused with the private key of a public key cryptosystem to be discussed later.

key, D_k, are easily derived from one another (i.e., complements) is recognized as having equality between the keys and is considered to have a single "logical key."

Single key cryptosystems require a secure, out-of-band process to exchange the encryption/decryption key between Alice and Bob. An "out-of-band" process in this context is one where the key being exchanged is not being used to secure the exchange.

E. TWO (PUBLIC) KEY CRYPTOGRAPHY

In 1976, Whitfield Diffie and Martin Hellman of Stanford University released, "New Directions in Cryptography," the first published discussion of the concept of public key cryptography. (Diffie, W. and Hellman, M.E., 1976.) Although Diffie, Hellman, and Ralph Merkle are generally credited for discovering public key cryptography, recently declassified documents released by the British Government Communication Headquarters (GCHQ) indicate that GCHQ employees first developed the concept in the early 1970's. The declassified documents credit Malcom Williamson with discovering an algorithm very similar to Diffie-Hellman in 1974. (Wayner, December 1997) Two key, public key or asymmetric key cryptosystems all describe this new branch of cryptography characterized by two different keys, E_k and D_k.

In these systems, the public key, E_k, and the private key, D_k, operate as inverses. The public key gets its name because it is distributed so that everyone has easy access to it. The public key is used for encryption and anyone can use it. The private key is necessary for decryption and must be kept secret to ensure confidentiality. The strength of these systems rests on the choice of E_k and D_k such that

knowledge of E_k and C does not reveal D_k. The system follows the same general cryptographic model notation presented above.

Notice that $P = D(D_k, E(E_k, P))$ establishes confidentiality, but it does not support authentication. Anyone can have access to E_k, so its use does not authenticate. In order to establish authentication, Alice must be able to encrypt with her private key. Hence, if the roles of the keys are reversed, E_k is the private key and D_k is the public key, $P = D(D_k, E(E_k, P))$ authenticates, but does not support confidentiality.

Public key implementations of the NP-complete problems detailed above, however, have a significant limitation. They require that the message length be smaller in character length than the modulus used. As modulus size is increased, encryption/decryption calculation times increase significantly. Even when messages are broken into smaller blocks, the slower algorithms associated with public key systems present an important engineering problem. Slow encryption is less likely to be used and may be more dangerous than no encryption. Often the engineering answer to this problem is a hybrid approach. Public key systems are used for key exchange and a symmetric cryptosystem is used for bulk encryption.

F. HASHING AND DIGESTS

Prior to discussing the use of public key cryptography for digital signatures, the mechanism of a digital signature's integrity, the hash function, must be examined. Employed by both the sender and the receiver, a hash function, or cryptographic sealing function, is an algorithm

that maps a variable length message into a much shorter, fixed-length representation. The resulting representation is known as a digest. Given the computational complexity of encryption using public key algorithms, encrypting a digest rather than the entire document creates a much smaller signature without extensive overhead. The ideal hashing function for use in digital signature applications must have a low probability of collision and exhibit sensitivity.

A good hash function should produce a unique message digest, but this is mathematically impossible since the general case allows for more potential messages than the number of possible (smaller) message digests. Two messages hashing to the same digest causes a collision. Since collisions cannot be eliminated entirely, the hash function must be designed to minimize the probability of their occurrence. For hash functions that produce a near random digest, this probability is a function of the size of the digest and the number of bit sequences representing meaningful messages.

The requirement for sensitivity complicates the design of hash functions with low probability of collision. In order to establish integrity, the hash function necessarily must produce a digest that changes by at least one bit whenever a single bit of the message is changed and allows no conspiracy of multiple changes in the original message to result in the same digest. In practice, hash functions are highly sensitive, producing very different digests as a result of changing a single bit. The strength of this sensitivity and the low probability of collision between meaningful messages are the basis upon which any additional

digital signature security features to authentication, such as non-repudiation and integrity, originate.

The digest of a good hash function can be thought of as a "fingerprint" for the message. A message's digest is an intrinsic characteristic of the message, uniquely identifying it from other messages. The same message will always hash to the same "fingerprint" digest. Like a fingerprint, a hash can be easily determined; no special key is required.

Two hashing functions in widespread use and considered reasonably secure are Message Digest Algorithm 5 (MD5) and Secure Hash Algorithm 1 (SHA-1).* MD5, developed by RSA Data Security, Inc., produces a 128-bit digest from an arbitrary length data stream. Since 2^{128} is vastly larger than the number of different messages likely to ever be exchanged in the world, MD5 has a low probability of collision.

SHA-1 is a United States Government standard developed by the National Institute of Standards and Technology with the assistance of the National Security Agency. It also takes an arbitrary length data stream, but produces a longer 160-bit digest. Both of these algorithms have the additional advantage of producing digests that are small enough to be efficiently encrypted using public key algorithms.

---

* MD5's predecessors from RSA Data Security, Inc., are MD2 and MD4. MD2 has no known weaknesses, but it is slower than MD5. MD4 has published flaws and is not recommended for use.

G. DIGITAL SIGNATURES

1. Single Key Signatures

Although a conventional cryptosystem provides authentication as long as the key remains private between Alice and Bob, it cannot establish integrity and nonrepudiation without an arbitrated protocol. Extending the mathematical notation of the model:

Let: T = Arbiter (trusted third party)

A_k = Single (private) key shared by A and T

B_k = Single (private) key shared by B and T

C_{AT} = Ciphertext message sent from A to T

C_{TB} = Ciphertext message sent from T to B

Then: $C_{AT} = E(A_k, P)$

$P = D(A_k, C_{AT})$

$C_{TB} = E(B_k, (A, P, E(A_k, P)))$

$A + P + E(A_k, P) = D(B_k, C_{TB})$

In theory, this scheme provides each of the digital signature's security features, but it does so at the expense of considerable overhead and requires that significant trust be placed in the hands of the Arbiter, T. Alice encrypts P using A_k, the single key she shares with the Arbiter. The Arbiter establishes that P is from Alice by decrypting C_{AT}.*

* At this point, no mechanism exists except the trustworthiness of T to protect P from being altered. This vulnerability could be eliminated

The Arbiter then encrypts everything with B_k and sends it to Bob. Bob is able to decrypt C_{TB} using B_k, and therefore authenticate Alice due to his trust in the arbiter, and read P. Bob must then store P and $E(A_k, P)$ to protect against dispute. Should Alice attempt repudiation of the signature, Bob can provide the Arbiter with P and $E(A_k, P)$. Although Bob is never able to read $E(A_k, P)$, the Arbiter is able to read it and resolve the dispute.

Since a non-arbitrated single key system scales exponentially, implementing an arbitrated symmetric key digital signature protocol across a distributed computer network produces a Herculean key-management problem. This alone makes this approach infeasible. Even if the problem of finding a trusted arbiter is solved, the arbiter is a potential system bottleneck. For every message it processes, the arbiter must find A_k, decrypt C_{AT}, find B_k, and encrypt C_{TB}. Attempting to distribute the load over multiple arbiters only multiplies the complexity of the key management problem. Each user would require a separate key for each additional arbiter. As a result, symmetric key systems are occasionally used for authentication within private networks, but are not generally used for digital signatures.

---

using a hashing function. Furthermore, confidentiality could be added if Alice and Bob shared a third private key and Alice encrypted P with it prior to her encryption of C_{AT}. This, however, would add significantly to the key management burden if the network were large.

2. Public Key Signatures

Using public key cryptography, Alice's private key can be used for encryption as well as decryption and her public key can be used for decryption as well as encryption.

Let: A_{Ek} = Alice's public key

A_{Dk} = Alice's private key

H = Hashing function

P_T = Plaintext message with attached date/time stamp

C_{Asign} = Alice's digital signature

Then: $C_{Asign} = E(A_{Dk}, H(P_T))$

$C = P_T + C_{Asign}$

$P_T = C - C_{Asign}$

$H(P_T) = D(A_{Ek}, C_{Asign})$

This algorithm produces a basic public key digital signature. Bob receives C from Alice and is able to read the unencrypted plaintext and date/time stamp, P_T, per the established protocol. Bob can then compute P_T's digest, $H(P_T)$, and compare it to the $H(P_T)$ he decrypts using Alice's public key. If they are identical, authentication, integrity, and non-repudiation are all established without the aid of an arbiter.

This digital signature protocol does not establish confidentiality, but as can be seen below, with another

step, this functionality can be easily added when P is small.

Let: B_{Ek} = Bob's public key

B_{Dk} = Bob's private key

Then: $C_{Asign} = E(A_{Dk}, H(P_T))$

$C = E(B_{Ek}, P_T + C_{Asign})$

$P_T + C_{Asign} = D(B_{Dk}, C)$

$H(P_T) = D(A_{Ek}, C_{Asign})$

As mentioned previously, if P is long, then Bob's public key pair would be used to transfer the symmetric key for a more efficient symmetric key algorithm like the United States Government's Data Encryption Standard (DES)*.

The problem of key management within a public key system is not trivial, but it is much easier to manage than the single key case. Since each user has one set of keys, a public key system scales arithmetically rather than exponentially. In order for a public key system to transfer trust reliably, however, a means to certify the validity of public keys must be established. How does Bob protect himself from the possibility that an unscrupulous user might fraudulently post a public key and represent it as Alice's public key? How does Alice go about revoking her public key if the confidentiality of her private key is compromised? A

* Chapter five discusses stronger symmetric algorithms, which are alternatives to DES.

PKI is a system established to address these issues. The next chapter will discuss these and the other basic issues of key management within the context of PKIs.

IV. KEY AND CERTIFICATE MANAGEMENT

A. KEYS, CERTIFICATES AND THE PKI

As alluded to in the previous chapter, the security of both modern symmetric and asymmetric cryptographic algorithms is predicated upon the security of their keys. Although some governments, and perhaps other large organizations, have the resources and user base necessary to make practical the use of algorithms that are themselves secret, the clear majority of the users of cryptography use publicly available algorithms in commercially manufactured applications.* For these users, a compromise of the private key necessarily compromises the intended security services supported by the algorithm. Even if the algorithm itself is secret, the security of the cryptosystem is designed to reside with the keys. As a result, a compromise of the private key will likely lead to the subsequent compromise of the secrecy of the algorithm too. Hence, the security of cryptographic keys is of ultimate criticality.

It is important to emphasize at this point what may be intuitively obvious to most readers: the security of private keys cannot be proven. Stated another way, it is not possible to prove that a private key has not been compromised. Given this reality, it is clearly necessary to provide a system to ensure the security of cryptographic keys and reduce the risk of compromise. This critical process is known as key management.

---

* Bruce Schneier's article, "Why Cryptography Is Harder Than It Looks," provides an excellent argument for why most users should choose open, commercial algorithms. (Schneier, B., 1996)

The general problem of key management is beyond the scope of this thesis.* Nevertheless, it is important to examine some of its basic precepts, particularly those associated with public key cryptography, since it is the role of a PKI to support key management.

Although the problem of key management within a public key system is not trivial, it is less complex than the symmetric key case described in the previous chapter. Since each user has one set of keys, a public key system scales arithmetically rather than exponentially. This is the characteristic of public key systems that makes their use within distributed computer networks so attractive for authentication, key exchange, and digital signatures.

In order for a public key system to transfer trust reliably, however, a means to certify the validity of public keys must be established. How does Bob protect himself from the possibility that an unscrupulous user might fraudulently post a public key and represent it as Alice's public key? How does Alice go about revoking her public key if the confidentiality of her private key is compromised?

The mechanism used to securely certify the validity of a public key, the digital certificate, was first proposed by Loren Kohnfelder in his 1978 bachelor's thesis in electrical engineering from the Massachusetts Institute of Technology. (Kohnfelder, L. M., 1978) A digital certificate, or "public

* Symmetric key management addresses issues such as generation of unique, algorithmically strong keys, key storage, secure retrieval, key distribution, key replacement and key retirement with the goals of creating and maintaining integrity and confidentiality between the communicants. Asymmetric key management addresses many of these same issues, but secure distribution is of the public key and secure storage is of the private key.

key certificate," is a formatted data structure that binds a public key to a subject using the digital signature of a trusted third party, known as a certification authority (CA). In general, the subject of a certificate can be any entity named in the certificate. Subjects can be persons, organizations, computers,* software agents or some attribute such as an account authorization or computer resource access. CAs operate within the context of a security policy detailing the process for gathering the names and public keys of the users it represents,** periodically changing these keys, handling the possibility of compromised or lost private keys, and securely distributing its own public key. This policy is articulated in a public document known as a certificate practice statement (CPS). The next chapter will discuss the functions of CAs in detail.

PKIs exist to support the security services that engender trust and are designed to manage cryptographic keys and digital certificates toward this end. Interoperability among PKIs is therefore rooted in common management functions and data structures. As such, these are the aspects of key management that will be addressed.

---

* Computers' "names" may be represented by DNS names, email addresses, IP addresses, or any other network label. It need not be uniquely associated with a specific hardware unit.

** Users in this context are known as "subscribers."

B. KEY AND CERTIFICATE MANAGEMENT FUNCTIONS

1. Registration

Registration is the process of determining the identity or validity of a subject so that it can be certified. This function is occasionally performed directly by the CA, but it is more commonly delegated to a subordinate authority that exists expressly for the function of registration. Not surprisingly, an authority so designated is called a registration authority (RA). The registration procedures within a given CA's domain are delineated in its CPS. (Chokhani, S. and Ford, W., 1999)

Registration is inherently an "out-of-band" process. This means that the process of verifying the identity or validity of the subject cannot be supported by the PKI itself. As a result, registration is often the most expensive single component of the total operational costs associated with a large scale PKI.

2. Creation and Issuance

The process for the creation of cryptographic keys, called key generation, is both algorithm and implementation specific. Asymmetric key pair generation within a PKI may be performed, as established in the CPS, either by the CA or by the subscriber in his local environment. In either case, the process must protect both the confidentiality of the private key and the uniqueness of the key pair. Since the strength of a strong cryptosystem is in its keys, the generation of cryptographically strong keys is the genesis of real security.

The process of creating and issuing a digital certificate, sometimes called certification, occurs after registration. Once the CA has verified the subject through

the registration process, it digitally signs a certificate that minimally contains the subject, the subject's public key, and the public key of the CA.

The certificate is "issued" when it is distributed to its subject and made available to the relying parties in the PKI. This is usually done by posting the certificate in some public forum, often electronically through a directory service. If the CA generates the asymmetric key pair for the subject, the issuance of the subject's private key must be done through some out-of-band process.

3. Storage

Generally speaking, the length and randomness of a cryptographic key makes their memorization and manual input impractical. As a result, cryptographic keys must be stored in some format that protects their confidentiality and integrity. Although this may often distill to a physical security problem, the practice of employing a digital storage medium can also make this an information security issue. Commonly, keys are stored electronically as either an encrypted file on a magnetic media or within dedicated hardware that is designed to provide tamper protection. Smart cards and other similar tokens are a popular media today for the storage of cryptographic keys.

4. Revocation

Certificates are not valid indefinitely. Revocation, as its name implies, is the process of invalidating a digital certificate and informing relying parties that the certificate is no longer valid. There are two categories of revocation: planned and unplanned.

The more frequently a cryptographic key is used, the more ciphertext is potentially available to an attacking

cryptanalyst. Good key management requires that keys be changed regularly to mitigate this risk. Certificates standards generally have a planned expiration date that is contained in their validity period fields. Planned revocation, sometimes called expiration, occurs when the certificate's life exceeds the validity period and is a normal stage of a certificate's life cycle. Since the certificate itself contains its expiration date, no special mechanism is needed to inform relying parties of planned revocation.

In contrast to planned revocation, unplanned revocation is an exceptional event that can be precipitated by a wide variety of unforeseen events. If the data contained in any of a certificate's fields becomes invalid or if the subject's private key is known or suspected to be compromised, the certificate may be subject to unplanned revocation. Other common causes for revocation include, but are not limited to, name changes, disassociation with a sponsoring organization, loss of the private key, and changes in access privileges.

Two principle mechanisms exist for a CA to inform relying parties of the unplanned revocation of its certificates, the certificate revocation list (CRL), and the on-line certificate status protocol (OCSP). The CPS of a CA will indicate which technique(s) the CA uses. (Chokhani, S. and Ford, W., 1999)

A CRL is a digitally signed data structure issued by a CA of its certificates that have been revoked before their planned expiration. CRL entries, revoked certificates, are removed after their planned expiration date. A CRL must contain the time of its issuance, usually in the form of a time-date stamp, and commonly contains the projected time of

issuance for the next CRL. This gives users an indication of which CRL is the most current. The integrity of a CRL is protected by the CA's digital signature so that it can be posted to any open directory where potential relying parties can access it. CRLs can be distributed by CAs in either a "push" mode where it is sent to all subscribers, or a "pull" mode where relying parties request the CRL as part of the certificate validation process.

The OCSP concept depends upon a certificate-status responder operated by a CA. A certificate-status responder provides upon request a digitally signed update of the certificate's status, either valid or revoked, to relying parties. OCSP is inherently a "pull" mechanism.

5. Re-key/Update

As stated above, good key management requires that keys be changed with a regular periodicity to reduce the cryptanalytic threat to any one key. In turn, the expiration of a key requires that any certificates containing it also be revoked. Naturally, when revocations are planned, a routine process can be established to generate new keys and certificates with minimal impact on the overall PKI. This process is especially critical when the key is that of a CA since all of a CA's certificates must be updated when the CA's certificate expires. Re-key or update of certificates is performed at some point established in the CA's CPS at the end of a certificate's lifetime, but prior to expiration.

When dealing with symmetric keys, "key update" is a specific key management function where both Alice and Bob create a new key by passing an existing shared key that is not believed to be compromised through a one-way function to

produce the same updated key. The only catch is that the confidentiality of the updated key produced in this way is only as secure as the previous key. Symmetric key update is used because it provides the added security of a higher key-change periodicity without the expense of an out-of-band secure exchange of new keys between Alice and Bob.

The use of "re-key" and "key update" within asymmetric key systems is less rigorous. Depending upon the policy of the CA, these terms often are distinguished from "re-issuance" because re-registration is not required. New keys are generated, the old certificate is used for registration to save the out-of-band expenses, and the CA issues a new, "re-keyed" certificate. This distinction is not universal, however, as some CAs will require re-registration of current subscribers whenever a new certificate is issued, but still call it an "update" or "re-key" when the subscriber's certificate is issued close to planned revocation.

6. Key Escrow/Key Recovery

As the use of cryptography for confidentiality continues to proliferate, the legitimate need under certain exceptional circumstances to recover data that has been encrypted has become an important security feature of some systems. Although data recovery is essentially a key management issue, it is also often sold as a security service and called "data availability."

Reasons for data recovery vary. Some common scenarios include the loss or inadvertent destruction of the original key, actions pursuant to a lawsuit discovery order, and law enforcement investigations under search warrant. When asymmetric key cryptography is used to establish a symmetric session key for confidentiality, either by key exchange or

by key agreement, two primary techniques exist for data recovery: key escrow and key recovery.

Key escrow is conceptually simple. When an asymmetric key pair is generated for either key exchange or key agreement, a copy of Alice's private key is made and stored securely. Depending upon the implementation, this copy may be stored by a trusted third party such as an employer, a CA, or a government agency. Additional security may be obtained by splitting the copy and storing it with multiple trusted third parties so no one third party has access to the key without complicity of the others. Alternatively, Alice may store it herself. Independent of who stores Alice's private key, duplicate storage is known as escrow.

Key recovery approaches data recovery by establishing access to the actual session key. One simple technique is for Alice to designate a trusted third party known as a key recovery agent (KRA). (Key Recovery Tutorial, 1999) When Alice establishes a session key with Bob, she also sends it to the KRA by encrypting it with the KRA's public key. Other more sophisticated protocols may also be employed for key recovery, but they all attain access to the symmetric confidentiality key.

When private key systems are used and the key is exchanged by an out-of-band process not involving asymmetric key cryptography, data recovery can only be based upon access to the symmetric key. The distinction between "escrow" and "recovery" is lost here, and either term may be used synonymously.*

---

* The reader is referred to (Denning, D. E. and Branstad, D. K., March 1996), (Denning, D. E., February 26, 1997) and (Schneier, B., 1996. p.97-100, 181-182) for a more extensive discussion of the topic.

7. Archival

Traditional paper signatures generally have a long lifetime, limited only by the endurance of the paper and ink. Since digital signatures are physically transient and totally dependent upon certificates, the certificates must be stored for the intended lifetime of signed documents in a secure manner. The long-term storage of certificates, even past the lifetime of the certificate, for the express contingency of establishing non-repudiation in the future is known as certificate archival. Naturally, a CA's policy will establish the parameters for archival.

8. Key Destruction

Destruction of private cryptographic keys is an often-overlooked aspect of key management. Secure destruction of keys, which usually means the complete physical destruction of the medium upon which the keys are stored, is important to ensure that old keys are not exploited.

C. DATA STRUCTURES AND STANDARDS

1. Certificate Types

Digital signatures are grouped into three principle types according to their subject and function: identity certificates, attribute certificates, and cross-certificates.

a) Identity Certificates

As the name denotes, identity certificates bind their subject's identity to their public key. Although the general concept of identity itself is philosophically complex, in this context the concept of identity is

simplified to the naming of the subject. The subject of an identity certificate is the name of an entity that either has a physical existence, such as a person or a computer, or software that has a "virtual" existence by dint of its ability to act as a proxy for an entity that has a physical existence. In either case, the subject of the certificate is the named entity that holds and uses the private key associated with the public key in the certificate. Examples of entities within a PKI might include CAs, RAs, KRAs, and end entities (EEs)*. Identity certificates answer the "who" question regarding the actors in distributed computer networks.

b) Attribute Certificates

An attribute certificate's subject is not an entity. Instead, the subject of an attribute certificate is some characteristic or set of characteristics of an entity that describes what the entity can do rather than who it is. Attribute certificates might be used where authority or privilege must be communicated, but privacy is also desired. A vendor of certain products may not need to know the identity of a purchaser who is placing an order electronically, but he does need to know that the entity placing the order has the authority to commit the money to pay for it. Attribute certificates provide both non-repudiation and privacy in this case.

---

* End entities are subscribers to a CA that are not RAs or other CAs. A more comprehensive discussion of the architectural elements of PKIs, including end entities, will be presented in the next chapter.

c) Cross-Certificates

A cross-certificate is a specialized certificate class whose subject is another CA. Cross-certificates allow for trust to be communicated between independent domains. When a CA issues a cross-certificate it tells its subscribers that the signature of another CA can be trusted with an acceptable level of assurance. The specifics are delineated in the CA's policy. The issuance of a cross-certificate by a CA to another domain is known as a "forward certificate" to subscribers within the issuing CA's domain. When other domains issue cross-certificates to the CA that issued the forward certificate, these are known as "reverse certificates" in the original CAs domain. When two CAs exchange cross-certificates so each domain expresses its trust in the other, a forward and reverse certificate pair exists. Together, the two certificates are called a "cross-certificate pair". (Burr, W.E., September 1998, p. 7)

2. Certificate Standards

Fundamentally, public key certificates are standardized, digital data structures, and they are the basis upon which PKIs are built. Several standards bodies have contributed to the establishment of various digital certificate standards, but the two most important international standards bodies to PKI development are the International Telecommunications Union (ITU) and the Internet Engineering Task Force (IETF). The international standard for public key certificates published by the ITU is X.509. The IETF currently has two Working Groups in its Security Area which are dedicated to Public Key Infrastructure standards development, the Simple Public Key Infrastructure (SPKI) and Public Key Infrastructure (X.509)

(PKIX) working groups. The Security Area also has other working groups, such as the Domain Name System Security (DNS Sec), the IP Security Protocol (IPSEC), and the Open Specification for Pretty Good Privacy (Open PGP) working groups, dedicated to the development of Internet security protocols that are public-key-cryptography enabled. Their key and certificate management functionality, however, is limited to support of the protocol and is not intended to function as PKI.

*a) Pretty Good Privacy (PGP)**

PGP is an electronic-mail security program first created in 1991 by Philip Zimmermann who freely distributed it as Version 1.0 throughout the Internet. PGP rapidly became one of the most popular public key cryptography applications for digital signature and confidentiality. The IETF issued RFC 1991, "PGP Message Exchange Formats," in August of 1996 which provided standardization for PGP Versions 2.X. (Atkins, D., Stallings, W., and Zimmermann, P., 1996) In November of 1998, the IETF issued RFC 2440, "OpenPGP Message Format," to update its standards for PGP Versions 5.X. (Callas, J., *et al.*, 1998)

As a stand-alone software application intended for individual use, PGP employs a relatively simple key-management scheme. PGP employs the "PGP key" which is functionally similar to an identity certificate. PGP keys, which are always created by their end user rather than a third party CA, contain an e-mail address as their subject, the associated public key, and a degree-of-trust attribute.

---

* "PGP" and "Pretty Good Privacy" are now registered trademarks of Network Associates, Inc.

The unique characteristic of a PGP key is that it can be signed by multiple relying parties, each of whom assigns his own degree-of-trust. This system works on a small scale where users know each other, but does not scale well.

Although PGP is mentioned here due to its popularity and widespread use, it is a secure messaging application rather than a general use PKI. As a result, it will not be addressed more fully. Readers interested in PGP are referred to *The Official PGP Users Guide*. (Zimmermann, P., 1995)

b) Simple Public Key Infrastructure (SPKI)

The IETF SPKI Working Group, building on the prior work performed on the Simple Distributed Security Infrastructure (SDSI) protocol, has specified its own digital certificate data structure. (Ellison, C., *et al.*, 1999) Subjects of SPKI certificates can be either names or explicit authorizations and are bound either directly to a public key or indirectly to the hash of a public key.

SPKI certificates fall into one of the following three categories: identity, attribute, or authorization. The subject of an identity certificate is a name that is bound, or "mapped," to a public key or its hash. This binding need not be unique, a name may be bound to many keys by the same certificate issuer. Attribute certificates bind an authorization to a name, and authorization certificates map an authorization to a key, or its hash.

Although SPKI supports three certificate types, it is designed and optimized around the philosophy that the primary purpose of a certificate is not authentication, but to tell an application that a keyholder is authorized for some privilege. (Ellison, C., *et al.*, 1999) This

authorization-primacy philosophy is derived from the assertion that naming is difficult and of limited functional utility. SPKI proponents argue that establishing globally unique names outside of a localized domain can become technically challenging on an Internet scale.* Careful engineering is necessary to avoid name collisions on a global scale and the process is manifestly complicated by political and sovereignty issues. Furthermore, computer applications rarely need the spelling of a name, the only real information a name alone denotes, in order to make a decision. Instead, applications need what a name connotes, a set of authorizations, to execute their programming; a name has meaning to a computer only when the name is associated with an authorization to proceed in a predefined manner. In fact, using names needlessly compromises privacy if a key and an authorization are all that is required to make a decision as how to execute. This philosophy makes SPKI substantially different than name-based PKIs such as the market dominate X.509 v3.

Although SPKI is an important academic exercise whose technical merit is superb, its failure to achieve significant vendor/market acceptance as a PKI standard has marginalized its importance as a contributor to PKI interoperability. As a result, a more comprehensive discussion of SPKI is beyond the scope of this thesis.

---

* The characteristics of a good global name, long enough (so that it is unique in the name space and typographical errors do not easily cause a collision) and random (so that its structure does not itself reveal private information) are also the characteristics of a public key or its hash. Hence, SPKI uses the public key, or its hash, when it needs a globally unique identifier.

c) X.509

The ITU-T X.509 certificate standard* was first specified as part of its X.500 directory standard in 1988. X.500 is a standard for a global, distributed database of named entities, such as people, computers, peripherals, etc., where each entry can be referenced via a unique identifier called a Distinguished Name (DN). The intended function of an X.509 certificate was to bind a subject's DN to a public key. The creators of X.500 envisioned using X.509 certificates for authentication prior to making directory node modifications. This original X.509 standard is now known as version 1 (X.509 v1).

In 1993 when X.500 was revised, two new fields were added to the X.509 v1 certificate to facilitate directory access control. The resulting certificate became version 2 (X.509 v2). The IETF's efforts to develop the Privacy Enhanced Mail (PEM) protocol and its supporting PKI using X.509 v1 and v2 certificates identified the need for additional modifications to the data structure.

In response to these new requirements, ANSI X9 in cooperation with ISO/IEC/ITU developed another revision that included a provision for optional extension fields. The use of these extensions was left unspecified, allowing other standards to specify extension types based upon their requirements. Organizations are also permitted to register extension types. This new, flexible data format allowing customization of the data structure for a wide range of applications was standardized as version 3 (X.509 v3) in June of 1996.

---

* Also ISO/IEC/ITU 9594-8.

Table 4 depicts the X.509 v3 certificate with the information fields protected by the issuing CA's signature shown in the shaded area. Note that the algorithm identifier and parameters fields of the signature itself are not protected.

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

Table 4 - X.509 v3 Certificate

Source: (Burr, W.E., September 1998, p. 6)

Naturally, a data structure with as much complexity and ambiguity as X.509 v3 must be further specified, creating specific extension types and rules for their use, before it is "standardized" enough for vendor-neutral, technically compatible implementations. This process of providing further specification is called "profiling" the standard.

In October of 1995, the PKIX Working Group of the IETF was formed with the express purpose of profiling the emerging X.509 v3 standard to meet the security requirements of the Internet. The basic Internet X.509 v3 profile, RFC 2459, is extensive, providing definitions for extension types, standardized data values for these extension types, certain sub-fields of the basic X.509 v3 information fields, and rules for their employment. Although it went through eleven drafts prior to becoming RFC 2459, the basic Internet X.509 v3 profile, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," has been embraced internationally by governments and industry as the standard certificate format for most PKI implementations. (Arsenault, A., and Turner, S., 1999)

3. Certificate Revocation Lists

The ITU defined only one mechanism in X.509 for revoking certificates, the CRL. The current version of this data format is the X.509 v2 CRL illustrated in Table 5.

Signature	algorithm identifier	algorithm used to sign CRL
	parameters	any parameters needed
Issuer	name of CRL issuer (X.500 "distinguished name," a sequence of Relative Distinguished Names that uniquely identify a directory object)	
This update	Time	update time stamp
Next update	Time	optional time of next update
Revoked certificates	list of revoked certificates	
CRL extensions (optional)	criticality flag	if "true" extension must be processed
zero or more extensions	extension parameters	
Issuer's signature		

Serial number	serial number of revoked certificate (unique for the issuer)	
Revocation date	Time	
CRL entry extensions (optional)	criticality flag	if "true" extension must be processed
zero or more extensions	extension parameter	

Table 5 - X.509 v2 Certificate Revocation List

Source: (Burr, W.E., September 1998, p. 9)

a) Advantages

The CRL is the only fully standardized mechanism for distributing revocation information to relying parties. A CA can easily manage the periodicity of CRL issuance to provide the data freshness necessary to meet the security

requirements of its subscribers without the costs and performance limitations associated with a responder architecture. Because a CRL is a digitally signed data structure, its integrity is protected, and it can be posted on multiple, low-assurance directories. This saves cost and avoids single points of failure.

b) Disadvantages

The limitations of CRLs include:

- Since CRLs are issued on a regular, periodic schedule, they may not contain the freshest possible information about certificate status. The delay between revocation and dissemination to relying parties is known as latency. Latency can be addressed by reducing the periodicity of CRL issuance, but this requires more communications bandwidth and reduces overall efficiency as relying parties are required to download a new CRL more frequently.
- As revocations become more frequent, CRLs become larger and more inefficient. This can be compensated for in part by shortening the validity period of certificates so they are removed from the CRL sooner, but this in turn requires more frequent certificate updates. (Not an unattractive feature to the CA, however, if it bills for certificate update.)
- Since CRL distribution uses untrusted repositories, and is intended to support caching, mirroring, and shadowing, it would be difficult for CA to charge relying parties for access to CRLs, thus limiting it as a source of revenue. As a result, CRLs do not support contractual privity between CAs and relying parties, which may necessitate special legislation to establish CA liability.
- CRLs (particularly indirect CRLs) are complex structures that add software-processing complexity to applications for relying parties.

c) Optimizations

Techniques that can improve CRL efficiency include:

- The use of Delta CRLs, which are shorter CRL lists that only include new revocations since some base CRL, and CRL distribution points, which is a technique for dividing the CRL data space into more manageable chunks, may be used to minimize the amount of information needed to validate a single certificate.
- Shorten validity periods.
- Cache CRLs locally once released
- Use OCSP when the application requires frequent revocation as is often the case with attribute certificates.

4. Online Certificate Status Protocol (OCSP)

As an alternative to CRLs, the PKIX working group has introduced OCSP to the IETF in June of 1999 as RFC 2560, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP."

a) Advantages

The on-line approach to revocation dissemination has a variety of advantages:

- Each response provided by the certificate status responder can be generated from the most current data available to the CA, creating potentially fresher status data.
- Less bandwidth may be required to send revocation data since each OCSP response contains less information than what is contained in a CRL.
- Each individual response may be processed faster since less data is exchanged.
- Although OCSP responses could be signed and cached, the client requires less memory for the storage of

revocation information; this may be critical for memory-limited applications such as mobile phones.

- Client software necessary for the processing of OCSP may be more compact and less complex than might be required to process CRLs.
- OCSP provides a potential cost recovery model where responder clients might bill relying parties on a straightforward, per response basis. If relying parties pay for status information from a CA responder, they can establish a contractual relationship with the CA and hold the CA legally liable within the context of this contractual exchange.

b) Disadvantages

Naturally, OCSP has its weaknesses too. Among the disadvantages to OCSP are:

- The transition from a repository that does not need to be trusted because the CRL is digitally signed to an on-line, trusted responder with high-availability and integrity requirements adds significant technical and security risks to the repository function. This could make the responder an attractive target to someone wishing to undermine the security of the system.
- The responder is inherently a centralized function and therefore a potential bottleneck/single point of failure.
- Potential responder performance enhancing solutions are likely to sacrifice some of the potential data freshness that makes OCSP so attractive

c) Optimizations

The primary optimization for OCSP is to cache responses locally for a designated period of time defined by the allowable certificate latency in the CAs policy. The penalty here is obviously to response freshness. A CA may also use multiple responders to add redundancy to the system

and reduce bottlenecks, but the penalty is the cost of its operation.

V. ARCHITECTURAL ELEMENTS OF PUBLIC KEY INFRASTRUCTURES

A. PKI OPERATIONAL COMPONENTS

A PKI generally has the following six operational components that perform its core functions.

1. Policy Management Authority (PMA)

Each PKI domain is "owned" by an entity, an individual, group or organization, known as its policy management authority (PMA). As the title indicates, the PMA determines the policies and procedures under which the other components of the PKI operate and is responsible for their enforcement. (Burr, W.E., September 1998) It is the PMA that designates a CA and has approval authority over its certificate practice statement (CPS). Authority for cross certification of other domains would likewise rest with the PMA. Often in a small scale PKI, the PMA will also function as the principal CA in the domain. As a consequence of ownership, the PMA accepts liability for the domain. The means and degree to which this liability is transferred is established by policy.

2. Certificate Authority (CA)

As introduced in Chapter IV, digital certificates are created by a trusted party known as a certification authority (CA) who is responsible to the PMA for the validity of the certificate subject/public key binding from certificate issuance to revocation. The CA establishes a security policy whose implementation is articulated in its CPS for gathering the public keys of the users it represents (subscribers), periodically changing these keys, handling the possibility of compromised or lost private keys, and securely distributing its own public key. CAs do this by

maintaining a database of subscribers' valid public keys, issuing cryptographicly sealed public key certificates to subscribers, and maintaining a publicly available means of announcing notices of unplanned certificate revocations, such as a CRL. Additionally, the CA serves as the intersection point for transferring trust between domains. Therefore, CAs are the principal agents within a PKI responsible for the key/certificate-management functions discussed in Chapter IV of creation/issuance, revocation, and re-key/update.

Depending upon the scale and implementation of a PKI, CAs may be used for other related and optional functions. CAs may perform other key/certificate management functions such as registration, key creation, key escrow, archival, and key destruction. A CA could also serve ancillary functions such as operating a time stamping authority (TSA) or a certificate status responder. Instances where CAs would take on these additional roles are limited, but it is important to realize that nothing prohibits a CA from adopting such additional responsibilities. The economies of scale in smaller domains often encourage an expanded CA role.

The public key of a CA is itself generally protected by a digital signature. When a CA distributes its public key with a self-signed certificate, its trustworthiness must be accepted axiomatically. For relying parties who directly trust this self-signed certificate, this CA serves as a trust anchor.* Authentication of a trust anchor's public

* A trust anchor is sometimes referred to as a "most trusted certificate authority" or a "trust root." It is not synonymous with "root CA," although a root CA is often also a trust root.

key is only possible through a secure, out-of-band process. When two CAs share the same PMA and their domain is organized with a hierarchical structure, the lower level CA whose certificate is signed by the higher level CA is known as a subordinate CA. If a trust anchor is the ultimate, topmost CA within a hierarchically structured PKI domain and it issues certificates to subordinate CAs, it is called the root CA. A further discussion of the organizational structures of PKIs is contained below in the section on PKI topography.

Today, CA services are either provided within an enterprise as an in-house support function, or contracted out to a third party. The largest commercial CA service provider in the U.S. is VeriSign, Inc.

3. Registration Authority (RA)

As introduced in Chapter IV, a registration authority (RA), sometimes called a local registration authority or an organizational registration authority, is an optional component that performs the registration function of certificate management. Like CAs, RAs can be organized into hierarchical structures where necessary to serve the physical distribution of the subscriber base.

RAs use out-of-band administrative techniques to establish the proof of possession (POP) for a subject or verify permissions for requested privileges on behalf of a CA. As a result, oversight of RAs is often provided by either, or both, the CA it serves and the PMA.

4. Repository

A repository provides the directory services necessary to support the storage of a CA's certificates, CPS, and CRL, and may be responsible for the key/certificate-management

function of archival. Repositories need not be trusted entities since the data structures they post are digitally signed by a CA. Repositories may also operate an OCSP certificate responder in some PKIs.

Like RAs, repositories are optional entities within the PKI that support CAs directly. There is no limit to the number of repositories that support a CA or to the number of different CAs that a repository might support. Similarly, oversight of a repository may be provided by either, or both, the CAs and PMAs it serves.

5. Certificate Path Validating Clients/Applications

PKI-enabled applications must be able to establish a validated certificate path back to a trust anchor. A trust anchor is a CA whose public key is trusted by a relying party due to an out-of-band certification process. Establishing a validated certificate path, or certificate path processing, is the foundation of a PKI's ability to convey trust.

Consider the example of Alice (the sender) and Bob (the receiver), where Bob wishes to validate Alice's certificate. If Alice's certificate is signed by Bob's trust anchor, path processing is trivial; Bob is able to validate Alice's certificate directly. In general, however, a CA that is not even in the same domain as one of Bob's trust anchors may sign Alice's certificate. If this is the case, Bob must be able to establish a path, a list of CAs of arbitrary length which successively supports either forward certification from Bob's trust anchor to Alice's CA or reverse certification from Alice's CA to Bob's trust anchor. Figure 2 depicts a simple certificate path where "CA1" is Alice's trust root. The software that discovers this path and processes the digital signatures of the CA certificates on the path is an essential element of a PKI.

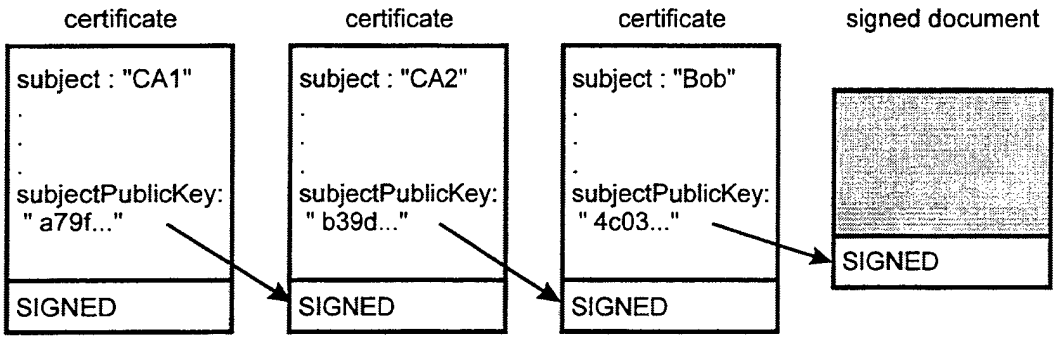

Figure 2. Certificate Path Processing

Source: (Burr, W.E., September 1998, p. 15)

6. End Entities(EE)

An end entity (EE) is a certificate holder in a PKI that is not a CA or RA. The vast majority of a CA's subscribers should be end entities.

A relying party is any entity that seeks to establish trust using a PKI. Although most EEs may at one time be a relying party, a relying party need not have a certificate. As a result, a relying party does not need to be part of the PKI.

B. SUPPORTING INFRASTRUCTURE

Like any infrastructure, a PKI depends upon a host of components that play critical supporting roles, but are not fundamentally operational components.

1. Standardized Data Structures

Any PKI requires that its data structures be standardized. As discussed in Chapter IV, the two primary data structures in a PKI are the certificate itself and the CRL. Given that X.509 v3 has become the *de facto* standard

for commercial PKI implementations, it will be the only standard discussed here.

a) PKIX X.509 v3 Certificates

Although PKIX has profiled the X.509 v3 certificate standard, individual PKI implementations can selectively employ the PKIX extension types based upon the applications they wish to support and remain PKIX compliant.

Table 6 depicts the PKIX X.509 v3 extensions that will be used in the U.S. Federal PKI (FPKI) and is an example of how the standard PKIX Internet profile is designed.

Extension	Used By	Use	Critical (see Note)
Key and Policy Information			
authorityKeyIdentifier	all	identifies the CA key used to sign this certificate	No
keyIdentifier	all	unique with respect to authority.	
authorityCertIssuer	all	identifies issuing authority of CA's certificate; alternative to key identifier	
authorityCertSerialNumber	all	used with authorityCertIssuer	
subjectKeyIdentifier	all	identifies different keys for same subject	No
keyUsage	all	defines allowed purposes for use of key (e.g., digital signature, key agreement...)	Yes*
privateKeyUsagePeriod	all	for digital signature keys only. Signatures on documents that purport to be dated outside the period are invalid.	Opt.
certificatePolicies	all	policy identifiers and qualifiers that identify and qualify the policies that apply to the certificate	Opt.
policyIdentifiers	all	the OID of a policy.	
policyQualifiers	all	more information about the policy	
policyMappings	CA	indicates equivalent policies	No
Certificate Subject and Issuer Attributes			
subjectAltName	all	used to list alternative names	Opt.

		(e.g., rfc822 name, X.400 address, IP address,...)	
issuerAltName	all	used to list alternative names	Opt.
subjectDirectoryAttributes	all	lists any desired attributes	Opt.
Certification Path Constraints			
basicConstraints	all	constraints on subject's role & path lengths	Yes*
cA	all	distinguish CA from end-entity cert.	
pathLenConstraint	CA	number of CAs that may follow in cert. path; 0 indicates that CA may only issue end-entity certs.	
nameConstraints	CA	limits subsequent CA cert. Name space.	Yes*
permittedSubtrees		names outside indicate subtrees are disallowed	
excludedSubtrees		indicates disallowed subtrees	
policyConstraints	all	constrains certs. issued by subsequent CAs	Yes*
policySet	all	those policies to which constraints apply	
requireExplicitPolicy	all	All certs. following in the cert. path must contain an acceptable policy identifier	
inhibitPolicyMapping	all	prevent policy mapping in following certs.	
CRL Identification			
cRLDistributionPoints	all	mechanism to divide long CRL into shorter lists	Opt.
distributionPoint	all	location from which CRL can be obtained	
reasons	all	reasons for cert. inclusion in CRL	
cRLIssuer	all	name of component that issues CRL.	

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

Table 6. Standard X.509 v3 certificate extensions for the FPKI

Source: (Burr, W.E., September 1998, p. 8)

b) *X.509 v2 CRLs*

The extensions for X.509 v2 CRLs can also be profiled. Table 7 depicts the CRL extensions that will be used in the FPKI.

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

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

Table 7. Standard X.509 v2 certificate revocation list extensions for the FPKI

Source: (Burr, W.E., September 1998, p. 10)

2. Cryptographic Algorithm Standards

Chapter III introduced the concepts behind the cryptographic algorithms used in PKIs. Commercial PKI implementations support one or more algorithms for each of the four principle cryptographic functions of digital signature, data confidentiality, key exchange, and hashing. Although a comprehensive listing of commercially available algorithms or a detailed description of the individual algorithms presented below are beyond the scope of this

thesis, the major algorithms being implemented today for each cryptographic function are presented for completeness.

Vendors of cryptographic algorithms subscribe to a variety of standards including:

- U.S. Federal Information Processing Standards (FIPS) developed by NIST for the Federal government
- The X9 family of American National Standards Institute (ANSI) standards developed to support the financial services industry
- Public Key Cryptography Standards developed by RSA Laboratories in cooperation with other vendors
- Requests for Comments (RFC) developed by the IETF for the Internet

a) Digital Signature

Most digital signature algorithms are asymmetric. The three principal digital signature algorithms being used in PKIs are the U.S. Digital Signature Algorithm (DSA), RSA, and the Elliptic Curve Digital Signature Algorithm (ECDSA).

The DSA was originally specified in FIPS 186 and, until recently, was the only digital signature algorithm used by the U.S. Government for unclassified applications. On December 15, 1998, FIPS 186 was superceded by FIPS 186-1. This allowed RSA, as specified in ANSI X9.31, or DSA to be used by the U.S. Government for digital signature.

RSA is the commercially dominant signature algorithm owned by RSA Data Security, Inc. Most commercial implementations of RSA conform to the specification in PKCS-1. It is important to realize, however, that RSA implemented under ANSI X9.31 is not compatible with PKCS-1 RSA.

The patent on RSA will expire on September 20, 2000. This is likely to result in an expansion of its use since standards organizations prefer to adopt algorithms that are already proven in an operational setting and are not encumbered by intellectual property rights.

Standards for ECDSA are specified in ANSI X9.62 and PKCS-13. NIST is also looking at updating the FIPS to include ECDSA.

b) Data Confidentiality

Symmetric algorithms are used for data confidentiality and constitute the largest functional block of commercially implemented algorithms.

The U.S. Government standard for data confidentiality, the Data Encryption Standard (DES) as expressed in FIPS 46-2, was first adopted in 1977 and is no longer considered to be strong enough for many applications. Triple DES, as specified in ANSI X9.52 and draft FIPS 46-3, is considered an interim solution where DES's 56-bit key size is insufficient. NIST is in the process of selecting an algorithm from the five Advanced Encryption Standard (AES) Round 2 finalists to replace DES.* The AES algorithm is projected to be selected in 2000 and incorporated into a FIPS by 2001. (AES Development Effort, 1999) The NSA developed, 80-bit SKIPJACK stream cipher algorithm was declassified on June 24, 1998 and is now specified as the Escrowed Encryption Standard (EES) in FIPS 185. Its controversial key escrow features implemented in its Law Enforcement Access Field (LEAF) make its commercial acceptance unlikely.

The International Data Encryption Algorithm (IDEA), released in 1991, is a highly respected algorithm described as "the best and most secure block algorithm available to the public at this time." (Schneier, B., 1996, p. 319) IDEA is patented in the U.S. and Europe by the Swiss company Ascom-Tech AG, but a license fee is not

* The five AES Round 2 finalists are MARS, RC6, RIJNDAEL, Serpent and Twofish. (AES Development Effort, 1999)

charged for non-commercial use. IDEA is now the confidentiality algorithm used by PGP.

CAST is a block cipher developed in Canada by Carlisle Adams and Stafford Tavares, but they claim that the name is not derived from their initials. Other than brute force, there is no known way to break CAST. (Schneier, B., 1996, p. 335) CAST is supported by Entrust Technologies, Inc., a major PKI vendor who submitted CAST with a 256-bit key as an AES candidate. IETF RFC 2144 governs the Internet use of CAST with a 128-bit key.

RC2, RC4, RC5, and RC6 are RSA Data Security, Inc. ciphers that are commonly used on the Internet. RC2, a block cipher, and RC4, a stream cipher, allow variable key lengths up to 2048 bits. These two algorithms have been maintained as trade secrets, but are not patented. The U.S. export versions, however, are limited to a 40-bit key, making them vulnerable to a brute force attack. RC5 is a patented block cipher that allows for variable block size, key size, and number of encryption rounds whose Internet use is governed by RFC 2040. RC6 is an evolutionary improvement on RC5 that RSA Data Security, Inc. has submitted as a candidate for the AES and agreed to license royalty free should it be selected.

c) Key Exchange/Agreement

Public key algorithms are used for key exchange and agreement. Important algorithms include Diffe-Hellman, Key Exchange Algorithm (KEA), RSA, ELGalmal, and Elliptic Curve algorithms. Diffie-Hellman, the first published public key algorithm, is a commonly used key agreement algorithm. Originally patented, it is now freely available since the U.S. patent expired on April 29, 1997. ANSI X9.42 and PKCS-3 are both specifications for use of Diffe-Hellman. NSA's KEA was declassified June 24, 1998, and its use is

expected to grow outside of the U.S Federal government. RSA, ElGamal, and Elliptic Curve algorithms all have the advantage of being useful for both digital signature and key exchange. ANSI X9.63 and PKCS-13 are both Elliptic Curve specifications for key exchange.

d) Hashing

The two principle hashing algorithms in use, SHA-1 and MD5, are described in detail in Chapter III. Secure Hash Standard (SHA-1) is used as part of the Digital Signature Standard (DSS) as set forth in FIPS 180-1. MD5 is an RSA Data Security, Inc. algorithm documented in RFC 1321.

3. Cryptographic Module Standards

The infomation theoretic level of security provided by any cryptographic algorithm is totally dependent upon the correct implementation of the algorithm. Whether implemented in hardware, software, or firmware, independent verification of both the algorithm and the cryptographic module that executes it provides additional assurance that the vendor's product really does provide the security it advertises. Development of an open evaluation criterion is fundamental to this assurance.

The NIST and the Communications Security Establishment (CSE) of the Government of Canada have jointly developed the Cryptographic Module Validation Program (CMVP) under the auspices of FIPS 140-1 for the express purpose of providing this service. Independent labs, accredited by NIST/CSE under the National Voluntary Laboratory Accreditation Program (NVLAP) to perform FIPS 140-1 validation testing, receive fees from vendors to evaluate vendors' cryptographic modules. The independent lab evaluates a cryptographic module under FIPS 140-1 and submits a detailed report of the results of their testing to NIST and CSE. Based upon the

report, NIST and CSE issue a joint certificate to the vendor rating the security level attained and publish the cryptographic module on its list of FIPS 140-1 validated modules. This certificate also entitles the vendor to use the FIPS 140-1 trademarked seal in their advertising in a manner analogous to the Underwriters Laboratories safety seal. Although U.S. Government organizations are required to use FIPS 140-1 validated cryptographic modules for sensitive, but unclassified applications, participation in the program is voluntary.

4. Certificate Management Protocol Standards

As the name implies, certificate management protocols define the specific procedures for electronic certificate management related interactions between operational components of a PKI. For example, the procedure for an EE to electronically inform its CA that its private key has been compromised and request the unplanned revocation of its certificate would be dictated by a certificate management protocol. Because vendors and governments have developed their own proprietary certificate management protocols and adopted various competing conventions, the effort within PKIX to standardize certificate management protocols for Internet use has led to the emergence of multiple "standards."

Certificate management protocol standards are evolving following two tracks within PKIX. Both tracks separate certificate management protocols into two constituent parts, message format standards and message transmission protocol standards. PKIX has reached rough convergence on the message formatting, with both certificate management protocol standards incorporating the certificate formats in the now expired Internet Draft, "Internet X.509 Public Key Infrastructure Certificate Management Message Formats."

(Adams, C. and Meyers, M., 1998) Disagreement on the appropriate protocols for transferring these messages has resulted in the development of two standards that are not interoperable.

a) Certificate Management Protocol (CMP)

RFC 2510, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," was the first proposed PKIX certificate management protocol. (Adams, C. and Farrell, S., 1999) It employs a transfer protocol developed specifically for certificate management. The major PKI vendor Entrust Technologies, Inc., who largely developed the protocol, uses CMP in its products. The IBM JONAH project offers a freeware implementation of CMP.

b) Certificate Management Over CMS (CMC)

PKIX Internet Draft, "Certificate Management Messages over CMS," is the governing document for CMC. (Meyers, M., Liu X., Fox, B., and Weinstein, J., 1999) CMC uses the PKCS 10 certificate-request syntax and uses an existing transfer protocol, Secure Multipurpose Internet Mail Extensions (S/MIME), rather than developing a custom protocol. At the time of this writing, CMC is endorsed by VeriSign, Microsoft, Netscape, and Cisco Systems.

5. Operational Protocols

Operational protocols provide provisions for the distribution of certificates and CRLs using established protocols such as Lightweight Directory Access Protocol (LDAP), Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and X.500.

a) X.500

X.500 is the directory service architecture for which X.509 certificates were developed. X.500 directories are not in widespread use.

b) Lightweight Directory Access Protocol (LDAP)

LDAP is the most widely implemented PKI operational protocol in commercial products. RFC 1777, "Lightweight Directory Access Protocol," RFC 2559, "Internet X.509 Public Key Infrastructure Operational Protocols -- LDAPv2," and RFC 2587, "Internet X.509 Public Key Infrastructure LDAPv2 Schema," provide the basis for LDAPv2 use in PKIs. (Yeong, Y., Howes, T., and Kille, S., 1995) (Boeyen, S., Howes, T., and Richard, P., April 1999) (Boeyen, S., Howes, T., and Richard, P., June 1999)

The new version, LDAPv3, is documented in RFC 2251, "Lightweight Directory Access Protocol (V3)," and supports all the protocol elements of LDAPv2. (Wahal, M., Howes, T., and Kille, S., 1997)

c) FTP and HTTP

RFC 2585, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP," provides guidance for using FTP and HTTP for distribution of certificates and CRLs. (Housley, R. and Hoffman, P., 1999)

6. Legislation

Like any infrastructure, PKIs must operate within a legal framework that can profoundly influence its nature. Security services, such as non-repudiation, and privacy services, such as anonymity, are not merely technical services. These services may be required to meet legal requirements, which are both technical and procedural.

Additionally, these legal requirements often vary among jurisdictions. Legislation is also used to define or limit the exposure of a PKI's operational components to liability. Finally, legislation is used to enforce policy through government oversight and licensing.

The example of Utah's digital signature legislation is illustrative. In 1996, Utah became the first state to enact comprehensive digital signature legislation, and establish legal guidelines for the security policy of a CA. Under Utah law, a CA instituting these guidelines can be licensed by the State, thereby exempting it from liability. (Pinsky, L., 1997) (Utah Digital Signature Program, 1998)

Each of the United States has now enacted some form of digital signature legislation with the notable exceptions of Massachusetts, Michigan, New Jersey, New York, Pennsylvania, and South Dakota. (McBride, Baker, and Coles, 1999)

The U.S. Congress has discussed national-level digital signature legislation for several years, but action is still pending. Legislation has been held up by the fierce, ongoing debate over the national cryptographic policy issues of export restrictions and law enforcement access to encrypted communications.

7. Domain Naming

Directory technologies and certificates can require globally unique naming rules to function properly. X.500 Designated Names require such a construct. When two or more entities share identically the same name, a naming collision is said to have occurred since the entities can no longer be uniquely distinguished by name. A discussion of the techniques that can be used to prevent naming collisions is beyond the scope of this thesis. Nevertheless, the service or protocol that provides globally unique naming is an

important element of the supporting infrastructure to some PKIs.

8. Time and Time-Date Stamping

Time and date are critical factors when establishing non-repudiation, but they are also important to a variety of other computer applications. For example, time and date are critical to electronic security transactions since price is a function of time. As a result, time assignment can be an infrastructure of its own within distributed networks that supports PKIs.

The essential defining characteristics for any time assignment infrastructure are synchronization and temporal granularity. Since time elapses continuously and at a constant rate, at least in non-relativistic frames of reference, the accuracy of time is measured relative to its synchronization with a standard and in units of finite granularity. The time standard in the U.S. is provided by the U.S. National Timing Authority (NTA), the NIST.

The PKIX working group is in the initial stages of developing Time Stamp Protocols (TSP). Time stamping is a PKI-related security service where a trusted third party, known as a Time Stamp Authority (TSA), digitally signs a document after appending a time stamping token (TST). The TSP certifies the existence of the document prior to the date and time contained in the TST. TSP also defines another trusted third party, called a Temporal Data Authority (TDA), that appends a temporal data token (TDT) to documents prior to applying its digital signature. A TDT provides supplementary evidence that the date and time stamped was not stamped prior to the time attested by associating a reference to a non-deterministic temporal public event. For example, the value of a popular securities index such as the Dow Jones Industrial Average is

both public and arguably non-deterministic. If the Dow Jones Industrial Average quote was included in the TDT, a relying party can know with the probability that this index cannot be predicted that the time stamp was not issued prior to its stated value.

As PKI technology is implemented, time stamping may evolve from a PKI related service into a component element. If this occurs, time standards and whatever non-deterministic event is selected for TDTs will become important elements of PKI architectures.

C. DEFINING PKI ARCHITECTURAL CHARACTERISTICS

1. Topology

The structural relationships between subscribers and the CAs within a domain can vary in complexity, but they can be broken down into three fundamental infrastructure topologies: hierarchical, mesh, and trust list. Figure 3 depicts the layout of these three topologies.

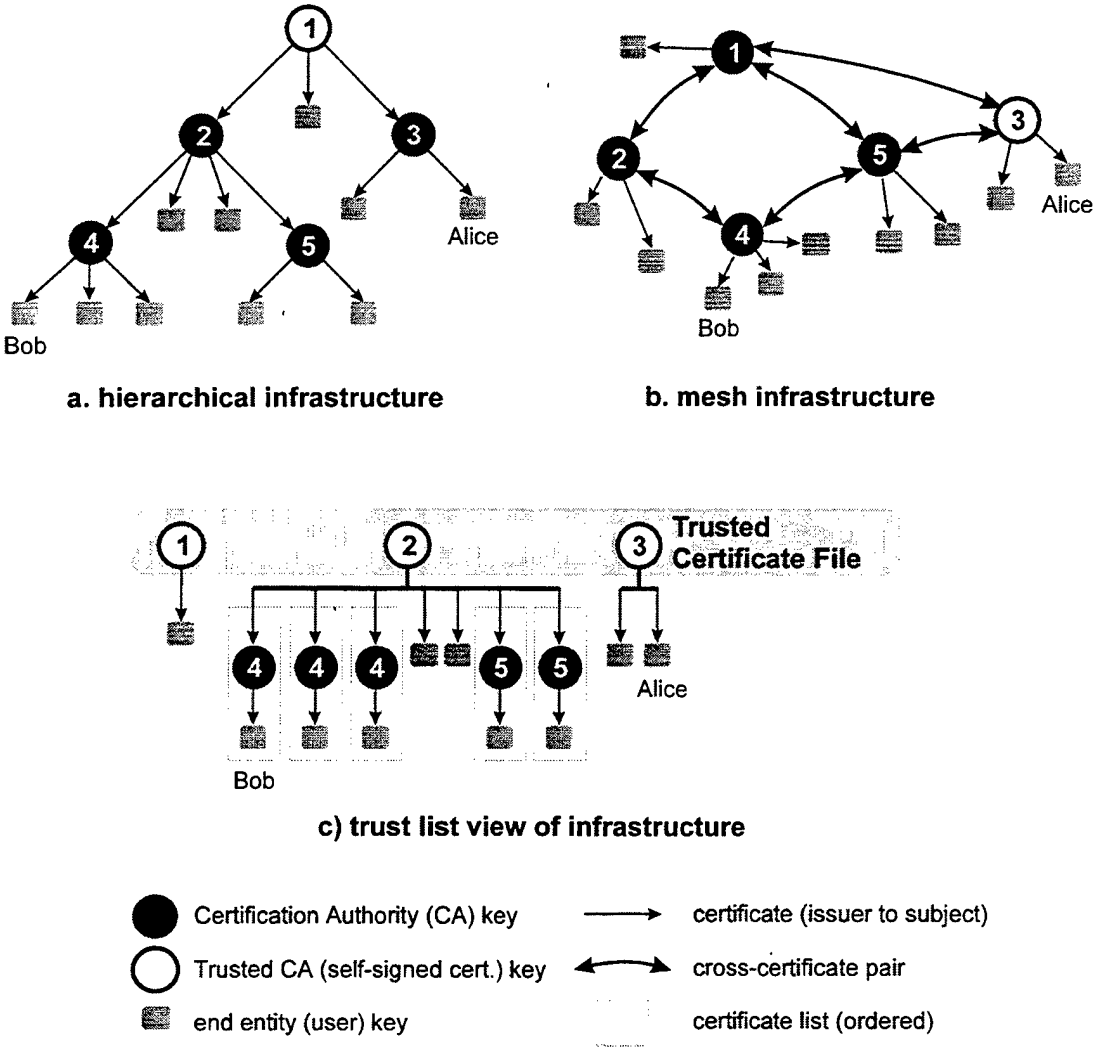

Figure 3. Basic PKI Topologies

Source: (Burr, W.E., September 1998, p. 16)

a) Hierarchical

In a hierarchical topology, a central CA issues a self-signed certificate and is the trust anchor upon which all subordinate CAs in the PKI ultimately validate their certificates. The hierarchy may be very flat, with one

central CA serving numerous satellite CAs. In a flat hierarchy directly subordinate CAs have EEs as subscribers. Alternatively, a hierarchy may have multiple layers forming an organizational pyramid of CAs beneath the root CA.

The advantage of a hierarchical architecture is centralized control. Key management policies are much easier to enforce. In addition, the effect of a compromise of any CA's key can be determined precisely. Orderly certificate revocation and re-issuance is possible. Hierarchically structured organizations may also find that mapping a hierarchical PKI to their existing organization aligns trust relationships in a natural manner. These infrastructures can also greatly simplify processing of certificate paths, and potentially speed up the validation process.

Unfortunately, centralized control is also a potential liability, because it represents a single point of failure. Compromise of the trust anchor's private key is catastrophic to the entire domain and requires an out-of-band distribution of a replacement public key for recovery. Additionally, a hierarchical infrastructure may not reflect the trust relationships that exist in some organizations. For example, when parties are all peers, a mutually respected higher authority may not exist. Hierarchies are often poorly suited to meet the needs of open commerce where parties may not subscribe to the same central authority.

Early efforts to develop PKI technologies often employed a hierarchical model. The failed Privacy Enhanced Mail (PEM) protocol illustrates the problem. PEM used a strict hierarchical architecture with its trust anchor administered by the Massachusetts Institute of Technology

(MIT) administering its trust anchor. Although this hierarchical topology was easier to engineer, the experience demonstrated that it is unrealistic for everyone to ultimately trust one entity (even MIT). Whenever possible, the PKI's structure should mirror the existing trust structure. Hence, the hierarchically structured architecture has its role, but other structures are necessary.

b) Mesh

A mesh topology has no centralized trust anchor. Instead, CAs certify one another forming an interdependent structure of peers known as a shared-trust system, or a "web of trust." Each CA is the trust anchor for its subscribers.

This topology more closely reflects the bilateral relationships that exist between independent organizations or businesses. Additionally, the impact of the compromise of any individual CA is far more localized, and is not necessarily catastrophic to the overall infrastructure. Finally, the ability of CAs to cross certify directly with other domains can significantly shorten frequently used inter-domain certificate paths.

Although a mesh structure scales easily across distributed networks, all CAs within the infrastructure have equal responsibility for policing the network. Enforcement, or rather self-enforcement, of key management "policy" across the infrastructure is voluntary. Since the effect of compromise is not structurally segregated, certificate revocation and re-issuance can be very complicated. The absence of centralized control and responsibility may not be

appropriate for transferring legal liability or high-trust applications.

The more complex the mesh structure, the more difficult it can be in the general case to discover certificate paths. In fact, as the mesh complexity grows, the infrastructure may not be able to maintain a guarantee of path discovery even if a valid path exists. Certificate path processing may "time-out" before a valid path is developed in a manner somewhat analogous to the time-to-live of a Transmission Control Protocol (TCP) packet.

c) Trust List

The two major commercial Internet browsers and their associated E-mail tools support only very limited certificate-path processing. Instead, they maintain a trust list, a local file containing the certificates of the trust anchors they recognize. In most cases, if a certificate is not signed directly by one of the CAs whose certificate is stored in the trust list, it cannot be automatically verified. Although some of these applications can retrieve a certificate from a repository when directed by the user, they do not have the capability to automatically discover a multi-step certificate path, nor do they support cross-certificates. They cannot process certificate policies or policy constraints extensions and have no means to automatically check for certificate revocation. Instead, their ability to sign, encrypt, decrypt, and verify is optimized to support such Internet protocols as S/MIME and Secure Socket Layer (SSL). The nearly ubiquitous market share of these products, however, has led to the "logical" infrastructure that they can support, the "browser-oriented" or trust list infrastructure.

Naturally, these products come pre-configured with the certificates their manufacturers accept, and users are free to add or delete certificates as they see fit. Today, no authentication or other security features are used to protect this list. Depending on the integrity features of the operating system upon which the browser is running and the computer's network connectivity, external agents may very well be able to modify the file. Such an implementation is a potential key management nightmare. Users must be very PKI savvy to understand the threat and configure their operating systems and browsers to use a trust list in a secure manner. Nevertheless, the dominant marketshare of these products makes the infrastructure they support relevant to any discussion of interoperability.

2. Scalability

The ability to expand a PKI to efficiently handle additional users is a qualitative metric of its architecture known as scalability. Given the enormous growth rate of the Internet, the scalability of a given PKI architecture may be critical to its ability to operate with other domains of indeterminate size.

3. Interoperability Models

The following three interoperability models are used to attain the capacity for a PKI to support trust across domains by retaining its security services at an established level of assurance:

a) Assimilation

As discussed in the introductory chapter, the trivial case of interoperability is assimilation. In this model, domains function under a single policy and are usually organized with identical architectures. If a domain wishes to establish interoperability, it becomes a

functional element of that domain. If the PKI is hierarchical, this requires assumption of the trust anchor of the assimilating domain. The obvious advantage of assimilation is the maintenance of centralized control and the ensuing key management benefits. Additionally, certificate path-validating clients and applications need not be altered to support assimilation.

Although the assimilated PKI's PMA completely subjugates its authority, it may maintain ownership of its domain and the capacity to divest itself. Assimilation, therefore, is still interoperability since the domains are not completely fused into as single domain and represent independent ownership.

b) Cross Certification (CA to CA)

The cross-certificate is the fundamental interoperability data structure. When a CA directly cross certifies the CA of another domain or the two CAs establish a cross-certificate pair, certificate paths can be processed between the domains. As long as the appropriate policy is created and enforced for the use of this certificate path, security services can be maintained with assurance across the domain boundary.

c) Third Party Bridge

It is possible to create a special CA whose principle function is to cross certify with CAs representing independent domains. The function of this special CA is to manage interoperability between domains. It is known as a bridge CA (BCA) because it provides a certificate path "bridge" between independent domains. Subscribing CAs to the BCA issue a single cross-certificate to the bridge. The BCA can then issue certificates to multiple other domains as established in its policy thereby substantially simplifying

cross certification for subscribing CAs. A bridge CA could also provide additional services that aid interoperability such as protocol translations, centralized repository mirroring, or even partial certificate-path processing.

4. Policy

a) Security and Privacy Services Supported

The set of security and privacy services supported by a PKI is the most important and general defining element of its architecture. Figure 4 depicts a PKI-centric view of some of the security services that a PKI can be designed to support.

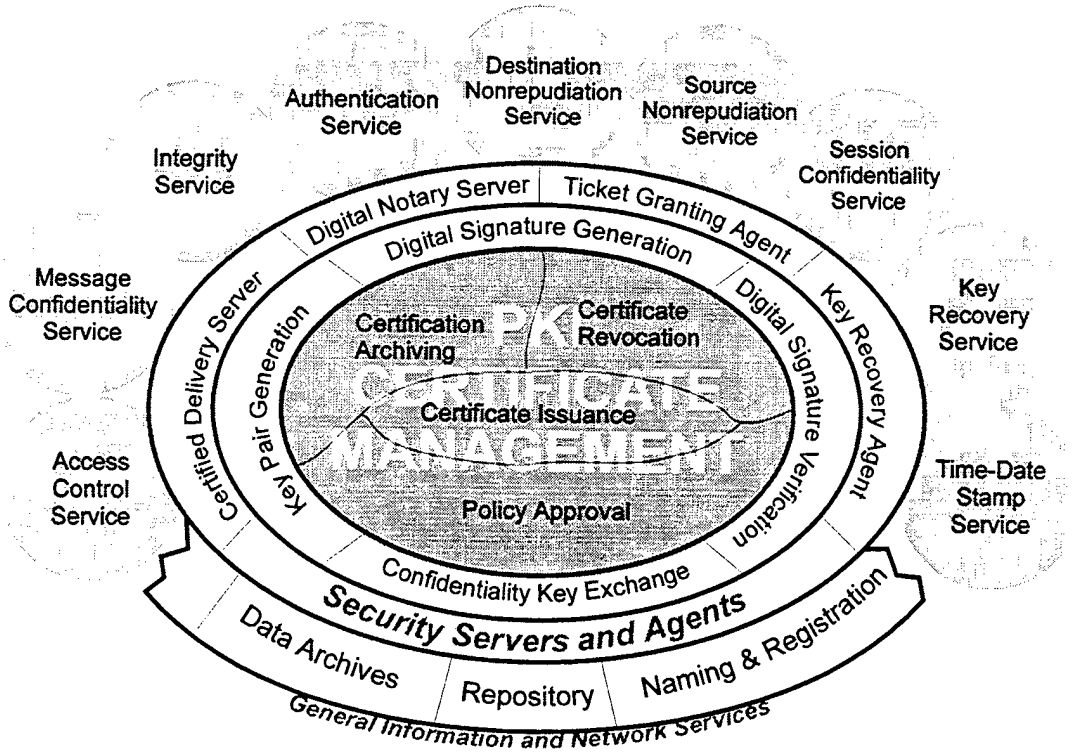

Figure 4. PKI Enabled Security Services

Source: (Burr, W.E., September 1998, p. 3)

b) Assurance Level

The physical and information security measures taken to support higher levels of assurance within a PKI drive up the cost of its implementation and operation. Paying for more assurance than is necessary wastes money. Conversely, accepting more risk than is prudent may cost more over the term of operation than the cost of higher assurance. Therefore, a PMA must conduct a risk management analysis and make a corresponding cost versus security decision to establish its assurance policy.

c) Certificate Latency

Certificate latency is the period of time between unplanned revocation of a certificate and the time at which notification of revocation is available to relying parties. Naturally, the longer a PKI's certificate latency, the more likely its security can be compromised. Although certificate latency can be minimized, providing near instantaneous latency is very expensive and cannot be totally eliminated. As a result, determination of an acceptable level of latency is a fundamental PKI policy decision driven by cost and the risk of latency compromise the PMA is willing to accept.

d) Cost Recovery Models

All PKIs have financial costs associated with their deployment and operation. Often these costs are incurred as overhead in enterprises that require the support of a dedicated PKI and therefore own and operate it themselves. Some enterprises are willing to outsource PKI services and contract with another enterprise to provide this service. Common payment models include payment by subscribers per issued certificate, payment by relying party per certificate use, or payment for specialized services such time stamping. The selection of the method for cost recovery is an important policy decision because the PKI must be structured in a manner to facilitate billing. Since the billing process may have profound implications on privacy as well as interoperability, it constitutes a fundamental policy decision.

VI. PUBLIC KEY INFRASTRUCTURE INTEROPERABILITY

A. CONTEXT

The previous five chapters provide the context for examining interoperability within large, heterogeneous enterprises such as the U.S. Federal government. PKI interoperability is defined functionally in terms of the security and privacy services a PKI supports and the level of assurance provided. This view of interoperability consists of two co-dependant dimensions: technical interoperability and functional interoperability.

B. TECHNICAL INTEROPERABILITY

Technical interoperability is the capacity of a PKI to establish and validate certificate paths across domains with an established level of assurance. All security and privacy services supported by a PKI are predicated upon the common requirement to trace a certificate path back to the trust anchor of the relying party. In general, interoperability becomes an issue when the certificate path must cross domains. This occurs when the trust anchor of the relying party, Bob, is not within the PKI domain of the sender, Alice.

Technical interoperability is a necessary, but not sufficient condition for PKI interoperability. It tests the compatibility of PKI domains. If two PKI domains are compatible, they can do the same things. Just because they can do the same technical things, however, does not necessarily imply that the procedures they both perform have the same meaning within both domains. As a result, the existence of a certificate path and the ability to process

it consistently across domain boundaries is merely a prerequisite to PKI interoperability.

Establishing technical interoperability is a major engineering challenge since PKI technology is so early in its development. The first step in realizing it, however, is to define its constituent elements.

1. Open, Standards-based Architecture

Multi-vendor technical interoperability requires the deployment of an open, standards-based PKI architecture, with interoperability as the fundamental design requirement of the standards.

a) Compatible Data Structures and Profiles

Certificates and CRLs, as the fundamental data structures containing information in a PKI, must be standards compliant to be processed correctly by validating agents. Certificates contain assurance data, such as policy identifiers and constraints, and technical interoperability data, such as domain and cryptographic algorithm identification, whose use varies between certificate profiles. Hence, limiting the number of standards for profiling certificates and CRLs is essential to interoperability. Similarly, as the number and complexity of profiles grows, careful attention must be paid to maintaining compatibility among diverse profiles.

b) Standard Cryptographic Algorithms

Each PKI establishes a suite of cryptographic algorithms for confidentiality, digital signature, hashing, and key distribution/exchange. Chapter V introduced the significant and most commonly used algorithms for each of these four applications. These cryptographic algorithms

must be open and standardized to facilitate multi-vendor implementation. The use of proprietary algorithms can impede technical interoperability.

Key escrow or recovery schemes may be incorporated as a fundamental component of a cryptographic algorithm's implementation. Even if data recovery is provided by a protocol external to the algorithm, compatibility of key escrow and recovery systems may be required to achieve technical interoperability.

Open, compatible standards for implementing these algorithms may not be enough; it is not practical for a validating agent to implement all algorithms. Instead, market forces will likely select a subset of the available algorithms to realize technical interoperability.

c) Standard Certificate Management Protocols

As discussed in Chapter V, CMP and CMC, PKIX's two certificate management protocol standards, are not compatible. Hence, validation agents must either implement both protocols, which is impractical, or sacrifice technical interoperability. PKIX participants expect market forces to select one of these standards.

d) Standard Operational Protocols

PKI technical interoperability is facilitated by selection of one of the four PKIX standardized operational protocols, X.500, LDAP, HTTP, or FTP. Today, LDAP appears to be the most commonly implemented operational protocol for repositories.

2. Standards Compliant Certificate-Path-Validation Agents

Certificate path processing is a function performed by the certificate path validation agent. As a result, technical interoperability is largely a function of the capability of the relying party's certificate-path-validation agent.

Certificate-path-validation agents are likely to be implemented as a component of a primary application, such as a browser or an email client, in order to support security and privacy services. Since these services may not be viewed as a primary feature of the product, it is very unlikely that vendors will implement more than a few standards from each of the general families listed above. Although standards compliance in implementations is fundamental, prudent algorithm and protocol selection can be equally as critical to general technical interoperability.

3. Verified Implementations

Properly implementing strong cryptography requires significant expertise and discipline. Secure public key cryptography is predicated upon the intractability of the algorithm, the security of private keys, and truly random key generation. A vendor's claim of standards compliance is not sufficient; validation of the implementation is necessary for assurance. This is equally true for the operational PKI elements and cryptographic algorithms themselves.

Two principal means of validation support public key cryptography: peer review and FIPS 140-1. The academic cryptography community, cryptographic consulting firms, and standards organizations review the intractability assumptions behind algorithms. This type of peer review requires open documentation, which can be entangled by the

need to protect trade secrets. FIPS 140-1, which was introduced in Chapter V, seeks to validate the security of cryptographic modules using a metric based upon four, tiered security levels. It requires rigorous testing by independent NIST/CSE certified laboratories.

Although the labs of cryptographic consulting firms will verify implementations for a fee, PKIX does not establish verification procedures to independently confirm standards compliance. Creation of a commercial PKI standards verification mechanism, however, would significantly improve both technical interoperability and security.

C. FUNCTIONAL INTEROPERABILITY

Functional interoperability is the capacity of a PKI to retain its security and privacy services at an established level of assurance, provided a verifiable certificate path can be developed. Without functional interoperability, technical interoperability is moot. Just because two PKI domains can perform the same functions digitally does not in itself inspire trust. For example, a verified certificate chain is useless if the subscribers in other domains fail to adequately protect their private keys.

Functional interoperability goes beyond system design and implementation to encompass human, economic, and operational factors that are sometimes overlooked by engineers. Like security itself, functional interoperability can be characterized and tested, but not objectively proven.

1. Policy

Policy is the single most significant contributing element to functional interoperability. Policy establishes

the security and privacy services to be supported, the degree of assurance required, and the procedures to be implemented in order to bring theory into practice.

2. Legislation

Legislation has the capacity to either facilitate or dramatically complicate functional interoperability. Legislation's potential to complicate PKI interoperability will be addressed later within the discussion of the challenges to realizing interoperability.

PKI legislation sometimes incorporates provisions for the licensing of CAs or other operational elements of a PKI. This can encourage functional interoperability by requiring standards for operation that enforce security and privacy assurance as a condition for licensing. Additionally, the conditions for maintenance of a license may include submission to compliance audits performed by governmental regulatory agencies or independent auditing bodies.

Finally, should PKIs follow the model of other utilities and infrastructure industries, the infrastructure and its use could become heavily regulated by government. Alternatively, PKIs could become a government-sponsored monopoly in some jurisdictions. This government oversight could foster functional interoperability by enforcing policy with strong centralized control.

3. Feasibility

The feasibility of a PKI is the likelihood or probability that a PKI will be realized and operate as intended by its designers. Security is usually a supporting function to the primary role of a system. As a result, security is perceived by some users as an impediment to the rendering of services by the system. These same users will sometimes deliberately place personal convenience ahead of mandated security procedures and inadvertently undermine the

primary services of the system as a result. Alternatively, operator ignorance or the complexity of security procedures can also end in compromise, even without malice on the part of the operator. As with all security systems, PKIs should be designed and implemented with due regard for feasibility.

a) User Interface/Ease of use

The success of any computer technology designed for widespread consumer use is closely linked to its human interface and the training necessary for an application's operation. This is especially true for complex security products that may be bypassed in favor of convenience if they are not made operationally apparent. PKI-enabled products must be easy to use, or they may not be used.

Non-use is the ultimate breakdown in functional interoperability. It guarantees that the intended security and privacy services are not supported. In fact, PKI non-use can be worse than not having a PKI since the presence of an unverified signature may create a false perception of security.

b) Speed of response

The average computer user is unwilling to wait for more than a few seconds while the computer is processing a task before becoming impatient. Certificate processing must be performed fast enough so that the user is not forced to wait until the processing is complete before continuing with the use of the PKI-enabled application. If users perceive the processing time as excessive and an impediment to the application's utility, it may result in non-use decisions due to inconvenience. Alternatively, users may choose to bypass certificate validation possibly creating a false sense of assurance and compromising the credibility of PKI services.

c) PKI Reliability and Availability

Recall that availability is a fundamental aspect of information security that is non-computable. Still, sound engineering practices can result in a robust, highly reliable PKI design that supports high-availability and gracefully degrades under attack. Denial-of-service attacks are an assault on a PKI's availability. The ability of a PKI to withstand such attacks, as well as environmental emergencies and component failure due to wear, is a measure of its reliability. Reliability is a technical interoperability issue to the extent that it supports certificate path creation across domains, but it is also a functional interoperability issue because reliability supports a user's assurance in cross-domain security and privacy services.

d) Economic viability and Cost effectiveness

The marketplace often decides which technologies become actual elements of our infrastructures. Technical "solutions" that do not sell are just ideas; in order for a technology to become a solution, it must sell itself and inspire correct use. The total lifecycle costs of a PKI, including development, implementation, operation, and disposal, must be justified in terms of the security and privacy benefits it provides or the infrastructure will not be economically viable. Because the marketplace will determine what PKI technology and security services are available, economic viability and cost effectiveness are also components of functional interoperability.

Cost effectiveness is also a critical area of functional interoperability because efforts to reduce components of the lifecycle cost may quickly compromise the effectiveness of the PKI. For example, if registration, an

expensive out-of-band process, is not performed with an adequate level of assurance in a misguided attempt to reduce costs, the PKI will be compromised even if it functions correctly from the perspective of technical interoperability.

e) Generality

PKI generality, the ability of a unified PKI architecture to support multiple applications, supports functional interoperability. If domains become more general, the variation between domains decreases. It follows that if the variation between domains can be minimized, this facilitates functional interoperability.

D. CHALLENGES TO REALIZING INTEROPERABILITY

1. Proprietary Profit Motive (First-to-Market Strategy)

The future market for PKI products is anticipated to be enormous, so competition between vendors for market share is intense. PKI vendors recognize the critical need for interoperability, but also want to establish a competitive advantage that will allow their products to capture the dominate position in the marketplace. As with any industry that is sensitive to standards, vendors want to attain standards compliance without compromising proprietary technology that may distinguish their products in the marketplace.*

Closely coupled to proprietary profit motive is the impetus to be first to market with a product that distinguishes itself. Commercial vendors in the computer

---

* Product differentiation is a key means to sway consumer purchasing and usage behavior regarding a product.

technologies often rush their products to market before they are mature. In their zeal to attain marketshare, they are willing to compromise such things as quality and interoperability with the philosophy that they can correct these problems in the next version.

2. Political Forces

A comprehensive discussion of the political issues that affect PKI interoperability is beyond the scope of this thesis. Instead, the remainder of this section will outline some the significant areas of political concern that may need to be resolved in order to attain interoperability.

a) Sovereignty Issues

The Internet is an international computer network whose novelty and growth rate has greatly outpaced government regulation.* As the use of the Internet and other distributed computer networks becomes a part of our daily lives, issues of sovereignty arise. Who has legal jurisdiction over the Internet? Who has the authority to tax electronic commerce and how can this be enforced across physical jurisdictions? What impact does the location of hardware have on legal status if it is controlled remotely? Can we trust a foreign CA? Governments realize that infrastructure standards can be used as a tool to advance political hegemony, so they may try to impede PKI interoperability as means to protect or advance their power and influence. These and countless other issues affecting PKI interoperability cannot be resolved, however, without

---

* The inherent delays involved with public debate and the democratic process of enacting legislation makes it difficult for legislative bodies to prudently reign in a target that moves as fast as the computer technologies.

first addressing the fundamental issues of political sovereignty in cyberspace.

b) Export Restrictions

The U.S. controls the export of strong cryptography under legislation that characterizes cryptography as a munition. Some of the other industrialized nations also place controls on export of cryptographic products or their use domestically. Use of strong cryptography is necessary to provide high assurance of security and privacy services using PKI technology. Current legislation before the U.S. Congress may ease export restrictions, but this will continue to be an obstacle when viewed from an international perspective.

c) Key Escrow for Law Enforcement

As mentioned above, the key escrow and recovery services supported by a PKI have the potential to impede technical interoperability. Law enforcement agencies within the administrations of several nations, including the United States and the United Kingdom, have attempted to establish a means of protecting their access to digital information through proposed policies on key escrow. This has met with significant opposition from privacy advocates and other national governments, which believe that it will be used by these nations to conduct economic espionage. As a result, key escrow/recovery is tightly coupled to other privacy concerns.

d) Privacy

The potential of a PKI to provide strong authentication to transactions that have not historically provided this service and the policies behind the

transference of transactional information are of great concern to privacy advocates. Currently, European Union countries have stronger legal requirements for protection of electronic privacy than the United States. (Ha, K.O., 1998) This has resulted in the current effort of PKIX to create a profile, known as a Qualified Certificate, that will support European Union privacy requirements.

e) Public Confidence

Any general use PKI will be dependent upon public confidence to engender trust. Implementers of PKI technology must be careful not to "oversell" their product to ensure that they do not compromise public confidence in the technology as a whole.

VII. PUBLIC KEY INFRASTRUCTURE INTEROPERABILITY WITHIN THE DEPARTMENT OF DEFENSE AND THE U.S. FEDERAL GOVERNMENT

Collectively, the previous chapters have developed a schema for the evaluation of PKI interoperability. This chapter will apply this framework to PKIs within the U.S. Federal government.

The U.S. Federal government is a large, heterogeneous organization whose diverse agencies are at various stages of implementing PKIs. Initial PKI pilot implementations within the Federal Information Infrastructure (FII) were designed by individual agencies and departments to meet the internal requirements of their individual enterprises without the benefit of an overall FII architecture. The Department of Defense (DoD) is a prime example. Its subset of the FII, the Defense Information Infrastructure (DII), has deployed two principal PKIs that are distinct from the designs being pursued by other agencies such as the General Services Administration (GSA).

A. REQUIREMENT FOR INTEROPERABILITY

The U.S. Federal government's need for PKI interoperability is driven by the complex set of customers it serves. A simple customer model for agencies of the U.S. Federal government recognizes five primary customer relationships (FPKI Steering Committee, 1998, p. 10):

- Internal Government: interactions within and between Federal agencies and departments
- Government-Government: interactions with local, state, and foreign governments
- Government-Citizen: interactions with private citizens to collect information and taxes and provide government services
- Government-Industry: interactions with industry to ensure compliance with Federal laws, regulations, and standards

- Government-Business: interactions with businesses to obtain goods and services

Each of these general interactions are supported electronically and require security and privacy services that could be supported more efficiently by PKI interoperability. Given the complexity of the linkages among the government's customers, realizing interoperability at this level is at the cutting edge of PKI technology.

Thus, the Federal government, and the DoD in particular, constitutes a microcosm for examination of PKI interoperability. Although in this thesis PKI is treated from a DoD-centric perspective, the general elements of PKI interoperability analysis could be extended to any other large, heterogeneous organizations.

B. DOD PKI POLICY AND IMPLEMENTATION STRATEGY

1. DoD PKI Architecture Engineers

The roles and responsibilities for the DoD PKI principles are established in *Public Key Infrastructure Roadmap for the Department of Defense*. (PKI Roadmap for the DoD, May 6, 1999)

a) Assistant Secretary of Defense for Command, Control, Communications and Intelligence [ASD(C3I)]

The ASD(C3I) is the Chief Information Officer for the DoD and the ultimate Policy Management Authority (PMA) for DoD PKIs. Although the ASD(C3I) retains the policy signature authority aspect of the PMA role, he delegates many of the functional aspects to the DoD PKI Steering Group. The DoD Steering Group is chaired by the DoD PKI Program Manager and is comprised of representatives from the National Security Agency (NSA), the Defense Information

Systems Agency (DISA), and the Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence [OASD(C3I)].

b) National Security Agency (NSA)

The NSA is responsible for providing a DoD PKI Program Manager and is the implementation authority for DoD PKIs. NSA is specifically tasked with development of the security criteria and architectures, as well as security testing and validation, for DoD PKIs. Additionally, NSA is responsible for the research and development of the key management aspects of DoD PKIs.

c) Defense Information Systems Agency (DISA)

DISA is responsible for providing a Deputy Program Manager to be co-located with the Program Management Office. DISA is the designated lead agency for integration of centralized components such as CA servers and directory services.

d) DoD PKI Program Management Office (PMO)

The DoD PKI PMO is responsible for coordination with the CINCs, Services, and Agencies to ensure the DoD PKI meets their requirements, and is the procurement agent for all centrally operated DoD PKI components. The DoD PKI PMO is specifically tasked with interoperability:

> The DoD PKI PMO will ensure that the DoD PKI is able to interoperate securely both within DoD and externally with Federal, NATO, partner nation, and business partners. The PMO must address technical challenges, as well as ensure that the necessary policies, practices, and procedures are in place to advance interoperability. (PKI Roadmap for the DoD, 1999, p. 21)

2. Information Valuation

The value of information within the DoD is defined in terms of its sensitivity (e.g., unclassified, sensitive or classified), its criticality (e.g., mission critical, mission support, and administrative), and its monetary value.

The *Public Key Infrastructure Roadmap for the Department of Defense* defines the three mission categories of criticality as follows:

- Mission Critical: Systems handling information determined to be vital to the operational readiness or mission effectiveness of deployed and contingency forces in terms of content and timeliness. It must be absolutely accurate and available on demand (may include classified information in a traditional context, as well as sensitive and unclassified information).
- Mission Support: Systems handling information that is important to the support of deployed and contingency forces. It must be absolutely accurate, but can sustain minimal delay without seriously affecting operational readiness or mission effectiveness (may be classified, but is more likely to be sensitive or unclassified).
- Administrative: Systems handling information that is necessary for the conduct of the day-to-day business, but does not materially affect support to deployed forces or the readiness of contingency forces in the short term (may be classified, but is more likely to be sensitive or unclassified). (PKI Roadmap for the DoD, 1999, pp. C-2&3)

3. DoD PKI Policy Classes

In order to provide both cost-effective and appropriate security services given the DoD's assessment of information's value and the risk environment, the DoD has established four hierarchical assurance-based PKI classes. This policy is articulated in *U.S. DoD X.509 Certificate Policy* (U.S. DoD X.509 Certificate Policy, 1999).

DoD PKI Classes are numbered from two through five to map directly to their Canadian counterparts. Canada has four PKI assurance levels: rudimentary, basic, medium, and high. Since the DoD does not have a need for an equivalent PKI with a Canadian rudimentary level of assurance, the DoD has not defined a "Class 1" PKI. Canadian basic, medium, and high levels of assurance roughly correspond to the DoD Classes 2, 3, and 4, respectively. The DoD Class 5, the highest assurance level, is designed to protect classified National Security Information and does not have a Canadian PKI assurance counterpart.

a) Class 2 (Formerly Basic)

A DoD Class 2 assurance PKI is intended to support applications that process information of low value (e.g., unclassified, Non-Mission-Critical, and small monetary value) or to protect system-high information in a low- to medium-risk environment (e.g., SIPRNET). Registration for a Class 2 certificate need not be in person, and cryptography can be software based. The DoD's Class 2 applicability guidance is as follows:

- Digital signature services for mission support or administrative data on any network;
- Key exchange for privacy of system high data in an encrypted network or for confidentiality of low value information on unclassified networks;
- Non-repudiation for small value financial applications such as travel claims or credit card purchases. (U.S. DoD X.509 Certificate Policy, 1999, p. 10)

b) Class 3 (Formerly Medium)

The DoD Class 3 assurance PKI is intended to support applications that process information of a medium value in a low- to medium-risk environment. Class 3 enabled

applications typically require identification of an entity as a legal person, rather than just a member of an organization. Registration for a Class 3 certificate must be in-person, and cryptography can be software based. The DoD's Class 3 applicability guidance is as follows:

- Digital signature services for mission critical and national security information on an encrypted network;
- Key exchange for the protection of communities of interest (COIs) and low valued compartmented information on an encrypted network;
- Non-repudiation for medium value financial or electronic commerce applications such as payroll, some contracting, vehicle purchases, etc. (U.S. DoD X.509 Certificate Policy, 1999, p. 10)

c) Class 4 (Formerly High)

The DoD Class 4 assurance PKI is intended to support applications that process information of a medium to high value in any risk environment. Class 4 enabled applications not only require identification of an entity as a legal person, but also require that private key material be stored in a cryptographic hardware token. In-person registration is required. The DoD's Class 4 applicability guidance is as follows:

- Digital signature services for unclassified mission critical or national security information in an unencrypted network;
- Key exchange for confidentiality of high valued compartmented information on encrypted networks, and COIs or classified data over an unencrypted network on a case by case basis (e.g. FORTEZZA For Classified (FFC));
- Protection of information crossing classification boundaries (e.g. sending information from NIPRNET to SIPRNET);
- Non-repudiation for large financial or electronic commerce applications. (U.S. DoD X.509 Certificate Policy, 1999, pp. 10-11)

d) Class 5

The DoD Class 5 PKI is intended to support applications that process high-value information (e.g. classified) in a high-risk environment, such as an open, unprotected network. Class 5 PKIs require the use of NSA-approved Type I cryptography. The DoD's Class 5 applicability guidance is as follows:

- Key exchange for confidentiality of classified information over an unprotected network such as NIPRNET;
- Digital signature services for authentication of subscriber identity and credentials in support of providing access to classified information over an unprotected network such as NIPRNET when used with appropriate encryption;
- Digital signature services for authentication of key material in support of providing confidentiality services for classified information over an unprotected network such as NIPRNET. (U.S. DoD X.509 Certificate Policy, 1999, p. 11)

4. Implementation Plan

The DoD does not currently support a Class 2 PKI, nor does it have any plans to do so in the future. Instead, the ASD(C3I) has decided to use the Class 3 PKI for all Class 2 applications, making its detailed specification and interoperability moot. Consequently, Class 2 policy requirements will not be further developed.

The DoD has currently deployed both a Class 3 and a Class 4 PKI. The Deputy Secretary of Defense, John J. Hamre, directed the DoD to embark on an ambitious PKI implementation schedule in a memorandum dated May 6, 1999 requiring all DoD users to be issued a minimum of a Class 3 certificate no later than October of 2001. (Hamre, J.J., 1999) This memorandum also requires the DoD to migrate the Class 3 PKI to Class 4. It directs the issuance of Class 4

certificates for all unclassified applications by January of 2002 and the replacement of all Class 3 certificates with Class 4 upon their revocation after this date.

C. DOD CLASS 3 PKI ARCHITECTURE AND INTEROPERABILITY

The DoD has deployed a Class 3 PKI based upon a Netscape Certificate Server 1.0. This section will outline the current Class 3 architecture and examine its capacity for interoperability.

1. Security Services

The DoD Class 3 PKI is designed to provide authentication, integrity, and non-repudiation via digital signature and to provide both key recovery and data confidentiality via key exchange. Because Class 3 certificate policy is designed to establish identity, it can be used to support access control, but it does not support anonymity. The Class 3 PKI does not directly support time-date stamping.

2. Topology

The DoD Class 3 PKI has a relatively shallow, strictly hierarchical structure. The trust anchor for the entire PKI is a single root CA whose private key is held at the NSA vault in Finksburg, Maryland. Its direct subscribers are CAs in the PKI; it does not issue certificates to end entities (EEs). Currently, there are four CAs in the PKI all operated by DISA, two at each of the DISA Defense Mega-Centers in Chambersburg, Pennsylvania and Denver, Colorado. RAs and Local RAs are distributed wherever needed within DoD Agencies, the Services, and among the CINCs, and connect to DISA's Defense Mega-Centers via the NIPRNET.

3. Data Structures

The Class 3 PKI supports two digital signature certificate types, each of which conforms to the X.509 v3 FPKI certificate profile. The first certificate is intended for use in conjunction with key recovery and is designed for use in establishing a session key for data confidentiality. The second certificate is designed to support non-repudiation, and will not be used in conjunction with key recovery.

The Class 3 PKI uses X.509 v2 CRLs for the distribution of unplanned revocation information. Class 3 policy requires a weekly CRL periodicity and a certificate latency of less than 24 hours from the notification of compromise.

4. Cryptographic Algorithms

The Class 3 PKI implementation employs 1024 bit RSA and SHA-1 as its digital signature, key exchange, and hash algorithms. Given the current weakness of DES, the Class 3 PKI uses Triple-DES until the AES algorithm is available.

Although key recovery is a security service required by policy, the Class 3 PKI's cryptographic algorithms do not provide a technical means of providing key recovery or escrow. Hence, escrow has to be performed manually and incorporated into procedure.

5. Certificate Management and Operational Protocols

The Class 3 PKI was implemented while CMC and CMP were still being developed. It uses PKCS-10 Certificate Request Syntax Standard and SSL for certificate management functions. LDAP is the Class 3 PKI's operational protocol.

6. Repository

The two Defense Mega-Centers in Chambersburg, Pennsylvania and Denver, Colorado operate two independent, but mirrored Netscape directory service systems.

7. Interoperability Model

Cross-certificates are not currently supported by the DoD Class 3 PKI. As a result, it employs the assimilation interoperability model.

The DoD has framed assimilation through its designation of "external" certificate authorities (ECAs). External certificate authorities are financed and operated by a PMA outside of the DoD, and provide certificate services under the DoD's policy to non-DoD EEs. They are required, however, to be certified by the DoD root and cannot serve as trust anchors for their domains. In fact, if they had an established subscriber base prior to being accredited as an ECA, they are required to re-issue their certificates under the DoD root. Any certificates that are not re-issued will not be permitted to interoperate with the DoD PKI. (Guidelines for External Certification Authority Interoperability with the DoD PKI (Draft), 1999)

8. Technical Interoperability Issues

Early adopters of an emerging technology accept significant technical risk, and this is especially true for the DoD Class 3 PKI implementation. Although the DoD has wisely embraced commercial off-the-shelf (COTS) technology and a commitment to open standards with its Class 3 PKI, the rapid pace of PKI standards change will continue to make technical interoperability difficult to maintain.

The DoD uses FIPS 140-1 approved Level 2 cryptographic modules for the Class 3 PKI as a matter of policy. Although this supports functional interoperability by providing assurance, it presents a technical barrier to potential ECAs that currently employ common non-FIPS approved cryptographic algorithms. Similar technical barriers to potential ECAs, such as different existing data structures, certificate

management protocols, or operational protocols may also complicate technical interoperability.

The assimilation model of interoperability, however, negates the significance of many of these potential technical barriers. If a PMA is willing to accept assimilation of its PKI into the DoD's Class 3 PKI in order to attain interoperability, then acceptance of the Class 3 technical requirements is unlikely to serve as a deterrent.

9. Functional Interoperability Issues

From the perspective of functional interoperability, assimilation can be a very risky interoperability model that presumes the hegemony of DoD over other PKIs. Assimilation is attractive to the DoD because its capacity for centralized control fits well into DoD's cultural ethos and supports the highest level of assurance of any interoperability model. This level of security may prove to be too expensive, however.

Although certain defense contractors and close military allies may be willing to operate ECAs, assimilation does not model the peer relationships that DoD has with other Federal departments, NATO countries, partner nations, and many business that are not primarily defense contractors. The sovereignty concerns of other nations and political friction within the Federal government between departments make assimilation very unattractive to these peer entities. In order to make assimilation palatable to these "peers," at a minimum DoD is likely to have to pay indirectly or directly for their participation. This expense, both politically and fiscally, may preclude interoperability.

D. CLASS 4 INTEROPERABILITY

The DoD's interim Class 4 PKI is based upon the FORTEZZA Crypto Card that is being integrated into the

Defense Message System (DMS). The FORTEZZA program grew out of the NSA's Multi-Level Information Systems Security Initiative (MISSI) in the early part of this decade. As with other government off-the-shelf (GOTS) products, FORTEZZA is designed around U.S. Government standards as opposed to commercially developed standards.

1. Security Services

The DoD Class 4 PKI implementation is designed to provide authentication, integrity, non-repudiation, key escrow, and data confidentiality. FORTEZZA's certificate policy is designed to establish identity and can be used to support access control for privilege management, but it does not support anonymity. FORTEZZA does not employ a supporting time-date stamping architecture.

2. Topology

The DoD Class 4 PKI has a deeper topology than Class 3, but it is still a hierarchical structure. The trust anchor for FORTEZZA is a root CA, known as the Policy Approval Authority (PAA), whose private key is held at the NSA vault in Finksburg, Maryland. The PAA signs the certificates of subordinate CAs, known as Policy Creation Authorities (PCAs) and is the only authority that can sign cross-certificates. PCAs sign the certificates of third tier CAs, and are responsible for distribution of CRLs within their domain. Third tier CAs, called Certification Authorities, actually subscribe EEs. Registration of EEs is performed for Certification Authorities by RAs called Organizational Registration Authorities (ORAs). FORTEZZA also requires globally unique assignment of Distinguished Names (DNs). ORAs interface with a specialized RA, known as a Sub-Registration Authority (SRA), that create X.500 directory entries during registration on behalf of the ORA and ensure the global uniqueness of DNs.

3. Data Structures

FORTEZZA is in the process of upgrading from X.509 v1 certificates to X.509 v3 certificates. The X.509 v3 certificates comply with the FPKI certificate profile, support multiple policies, and are intended to support both access control and privilege management. Command privileges, such as the authority to release messages, and multi-level access control mechanisms, such as security clearance levels, are implemented using X.509 v3 certificates.

The Class 4 PKI uses X.509 CRLs for the distribution of unplanned revocation information. Class 4 policy requires a daily CRL periodicity and a certificate latency of less than six hours from the notification of compromise.

4. Cryptographic Algorithms

FORTEZZA employs only FIPS-approved cryptographic algorithms. Cryptographic hashing is performed using SHA-1 (FIPS 180-1) and digital signature is accomplished with DSA (FIPS 186). FORTEZZA uses KEA for key exchange and the recently declassified SKIPJACK algorithm (FIPS 185) for data confidentiality.

5. Certificate Management and Operational Protocols

FORTEZZA's certificate management protocol, the MISSI Management Protocol (MMP), is very similar to its successor, CMP, which was developed by the IETF.

DMS and FORTEZZA both employ X.500 directory services, so the Class 4 PKI operational protocol is X.500. NSA's Secure Data Network 702 (SDN.702) specification articulates the FORTEZZA's interface with X.500 directories.

6. Repository

The FORTEZZA Prototype X.500 Directory System provides directory services for the Class 4 PKI repository. The

FORTEZZA Prototype X.500 Directory System is an upgraded version of an X.500 Chromatix, Inc. product, Softpages, which is designed to work in tandem with DMS and support strong authentication.

7. Interoperability Model and Issues

FORTEZZA was originally intended as an enterprise solution; it was not engineered for interoperability. FORTEZZA was custom designed to meet the high-assurance needs of the DoD before significant commercial PKIs existed. As a result, FORTEZZA's interoperability is limited.

When FORTEZZA is upgraded to X.509 v3 certificates, it will support direct cross certification from the U.S. PAA to other nations' PAA. This cross certification is designed to allow signature interoperability with allied nations that the U.S. equipped with FORTEZZA. Once the PAA has issued a signature cross-certificate to another domain, it must be distributed to the affected users before interoperability can be realized. Cross certification does not, however, support data confidentiality outside of a PAA's domain. As a result, the upgrade of FORTEZZA to support cross certification will not provide full interoperability with other domains.

As is the case with the Class 3 PKI implementation, full interoperability is technically possible using the assimilation model. Since FORTEZZA is a GOTS product, assimilation would require ECAs to implement FORTEZZA, as opposed to adapting an existing COTS architecture to Class 4 policy. Additionally, Class 4 assimilation might meet with the same feasibility challenges discussed above for Class 3 assimilation. Although FORTEZZA ECAs have been considered by NSA, current policy calls for a limited deployment of FORTEZZA for use with DMS, and the development of a follow-

on Class 4 PKI with an architecture better suited for interoperability.

E. DOD PKI ROADMAP ARCHITECTURE

The DoD's general PKI strategy is articulated in *Public Key Infrastructure Roadmap for the Department of Defense* as follows:

> Recognizing that PKI technology is still immature in the marketplace and changing rapidly, DoD's strategy is to pursue early adoption of technology and services, actively participate with industry to obtain the detailed technical understanding needed to fully specify requirements, resolve standards issues, and accelerate industry wide convergence to a purely standards-based, interoperable capability which is not dependant on vendor-specific capabilities or technologies. (PKI Roadmap for the DoD, 1999, p. 1)

As stated previously, the Deputy Secretary of Defense has directed a transition to Class 4 by January of 2002. Since FORTEZZA is ill suited for interoperability and does not conform to this vision, a new Class 4 PKI is needed.

1. Security Services

The new Class 4 PKI, known as the target DoD PKI, will need to provide the same security services as FORTEZZA: confidentiality, integrity, non-repudiation, authentication, access control, and key recovery/escrow.

2. Topology

The DoD is disposed toward the assurance and centralized control that is facilitated by hierarchical PKI topologies. Figure 5 depicts the target PKI architecture. Note that its certificate management functions are controlled centrally to provide high-assurance and that the registration process is decentralized to scale well and reduce costs.

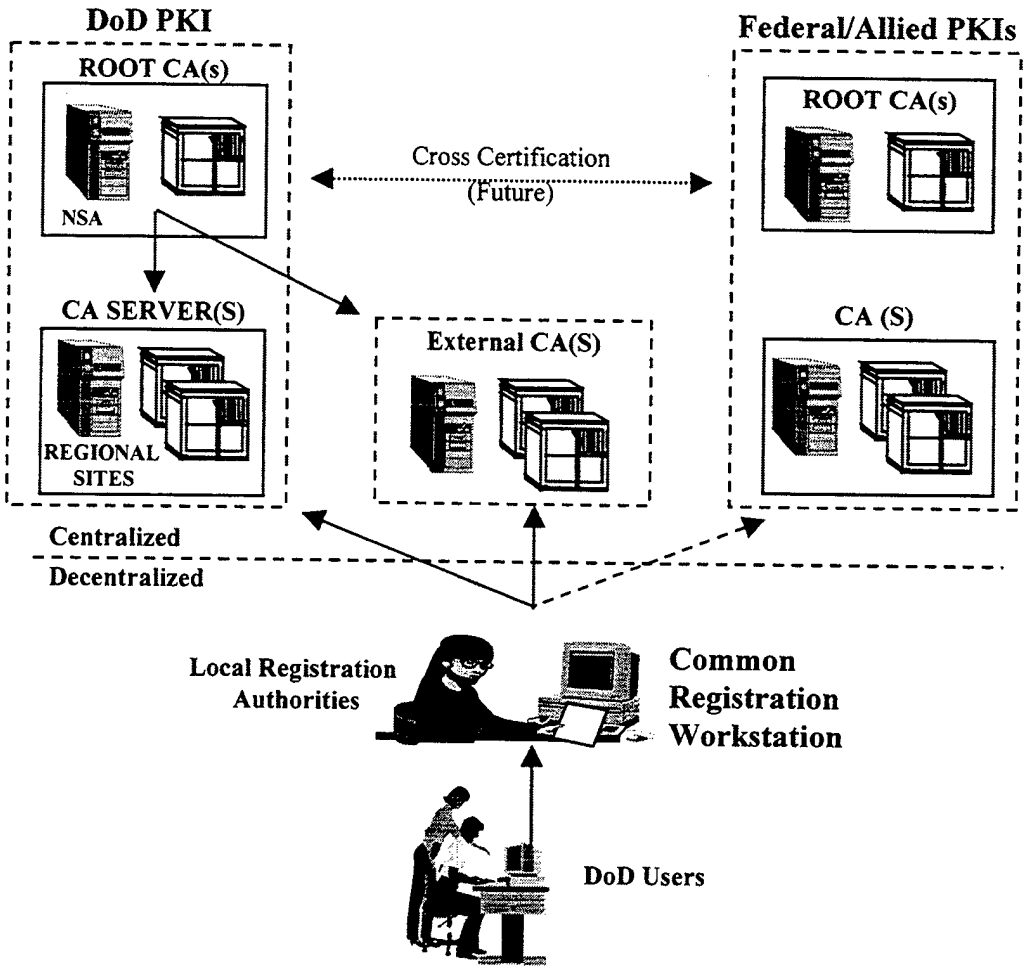

Figure 5. Target DoD PKI Architecture

Source: (PKI Roadmap for the DoD, 1999, p. 7)

3. Data Structures

The target DoD PKI's developers are committed to open standards. Given the focus of PKI standards bodies today and the DoD's experience with PKIs, it is very likely that the target DoD PKI will adopt X.509 v3 certificates and X.509 v2 CRLs. If OCSP is adopted widely by the marketplace, it is likely that the DoD PKI will use CRLs internally while supporting OCSP externally to achieve technical interoperability.

4. Cryptographic Algorithms

FIPS compliance is a stated target DoD PKI objective. (PKI Roadmap for the DoD, 1999, p. 5) As a result, the target DoD PKI is likely to adopt only FIPS approved cryptographic algorithms and FIPS validated cryptographic modules. While this may limit technical interoperability, it will probably not cripple it because NIST's desire to maintain the relevance of the FIPS system will ensure that a balance is maintained between strong cryptography and the commercial acceptance of its algorithms.

5. Certificate Management and Operational Protocols

The marketplace will determine which, if any, of the emerging IETF certificate management and operational protocols will be adopted by the target DoD PKI. The degree of uncertainty associated with the commercial acceptance of these protocols precludes an educated assessment of which protocol suite is most likely to prevail.

6. Repository

Although "the target PKI calls for a 'common' DoD-wide directory to support all DoD public key enabled applications," the specific directory services technology is uncertain. (PKI Roadmap for the DoD, 1999, p. 8) The DoD's investment in legacy X.500 directory services and the ability of some X.500 directories to support strong authentication makes X.500 a contender for the target DoD PKI. The lack of widespread commercial X.500 use, however, may push DoD to one of the newer directory service products. If the decision were made today, a directory accessed via LDAP would likely prevail.

7. Interoperability Model

Initially, the DoD target PKI will adopt the assimilation model for interoperability and designate ECAs as it has done with the Class 3 PKI. This ensures that the Class 4 assurance is maintained while PKI technology matures.

At some undefined point, the DoD recognizes it will need to adopt another interoperability model.

> The DoD target PKI will eventually achieve secure interoperability with non-DoD entities through a process called "direct cross certification," which establishes a policy and process for recognizing third party CAs, or through an evolving concept like the Federal Public Key Infrastructure (FPKI) "Bridge Certification Authority." ... Achieving this objective is dependent upon maturation of existing commercial applications and standards. (PKI Roadmap for the DoD, 1999, p. 8)

8. DoD/Federal PKI Interoperability Demonstration Project

NSA is conducting an advance technical interoperability demonstration project between a projected target DoD PKI architecture with a cross certification capability and a projected FPKI architecture. The success of this demonstration is likely to influence how soon the DoD will adopt direct cross certification.

Nine vendors, including Entrust, Raytheon, SPYRUS, Motorola, and Chromatix, are assisting with the demonstration. The demonstration uses existing COTS hardware for both domains, but commissions the development of a software library for certificate path development to prove the Federal bridge CA concept.

F. FEDERAL PKI ARCHITECTURE

Although DoD has arguably been the pioneer of large scale PKI technology within the U.S. Federal government, its ability to make and enforce centralized policy decisions makes it a relatively homogeneous environment when compared to the FII as a whole. While the DII may have the hegemony to adopt assimilation as an interim interoperability model, this is clearly not the case in the FII where peer relationships dominate.

The FII currently employs a diverse collection of PKI pilots that are best characterized as enterprise-specific or application-specific solutions. The FPKI, therefore, faces PKI interoperability challenges that are a microcosm of those faced by the greater NII and GII. The requirement to couple existing PKIs with diverse architectures has led the designers of the FPKI to the third party bridge model for interoperability.

1. Federal PKI Architecture Engineers

The FPKI has its organizational genesis with Vice President Gore's National Partnership for Reinventing Government, and his efforts through the Government Information Technology Services Board (GITSB) to promote the use of information technology and the Internet to improve Federal services while reducing costs. The GITSB was established in July of 1996 with President Clinton's Executive Order 13011, *Federal Information Technology* in response to the Paperwork Reduction Act of 1995 and the Information Technology Management Reform Act of 1996. The GITSB falls under the Office of Management and Budget within the Executive Office of the President of the United States.

a) Federal PKI Steering Committee

The Federal Public Key Infrastructure (FPKI) Steering Committee was established under the GITSB to oversee the development and implementation of the FPKI. The FPKI Steering Committee is chaired by the GITSB's Champion for Security and Privacy, and is comprised of representatives from all interested Federal agencies. Its chartered mission is as follows:

> ... The Federal PKI Steering Committee will provide guidance and assist in the development of an interoperable public key infrastructure that utilizes commercial-off-the-shelf, standards-based products and services for a myriad of applications with a goal toward ensuring standards-based approval. ... The Steering Committee will: identify Federal government PKI requirements, recommend policies, procedures and standards development activities that support a Federal PKI, provide oversight of PKI activities in Federal PKI pilot projects, provide oversight and guidance on the establishment of key recovery techniques, specify technologies needed for a Federal PKI, establish and maintain liaison with appropriate communities of interest, establish interoperability and security requirements of products and protocols related to the Federal PKI, and make recommendations regarding establishment, demonstration, and operation of a Federal PKI. (FPKI Steering Committee Charter, 1999)

The FPKI Steering Committee has three subordinate working groups, Technical, Legal and Policy, and Business, to assist in the accomplishment of this mission and a U.S./Canadian Liaison Group.

b) Federal PKI Technical Working Group (FPKI TWG)

As the title implies, the FPKI TWG is responsible for providing technical advice to the FPKI Steering Committee in the performance of its chartered mission. Issues of technical interoperability are principally

addressed by the TWG. The FPKI TWG chair is nominated by NIST, which is responsible for cryptographic standards for the protection of sensitive and unclassified data, and is approved by a vote of the Steering Committee. The TWG is the only working group where representatives of industry are invited to participate as voting members.

c) Federal PKI Business Working Group

The Business Working Group is responsible for the development of the FPKI's business model. It is comprised of representatives of Federal agencies involved in PKI development and addresses issues of functional interoperability.

d) Federal PKI Legal and Policy Working Group

The Legal and Policy Working Group is responsible for providing legal guidance regarding the FPKI and recommending FPKI policy to the Steering Committee. It too is comprised of representatives of Federal agencies involved in PKI development and addresses issues of functional interoperability.

2. Proposed Federal PKI Architecture

The FPKI is still in the concept and initial design phase of development. Given the extensive investments agencies within the FII have made in PKI technologies and the diversity of PKI implementations in the greater NII and GII, the FPKI's developers envision the FPKI as a third party bridge to facilitate FII interoperability. This voluntarily employed domain nexus would serve to connect PKIs within the FII, and provide interoperability between PKIs in the FII and those in the greater NII or GII. The proposed FPKI is comprised of two fundamental elements, a

Bridge CA (BCA) and the border directory server architecture.

a) Bridge Certificate Authority (BCA) Architecture

The third party bridge concept, introduced in Chapter V, provides a means of establishing a certificate path between domains through principal CAs within each domain. A "principal CA" is a CA that is cross certified with a bridge CA. In hierarchically structured domains, the trust root is usually the principal CA, but any CA could be a principal CA in a mesh infrastructure domain. In order to reduce complexity, the FPKI will only support one principal CA per domain.

The resulting vision for the certificate path architecture of the FPKI BCA is depicted in Figure 6.

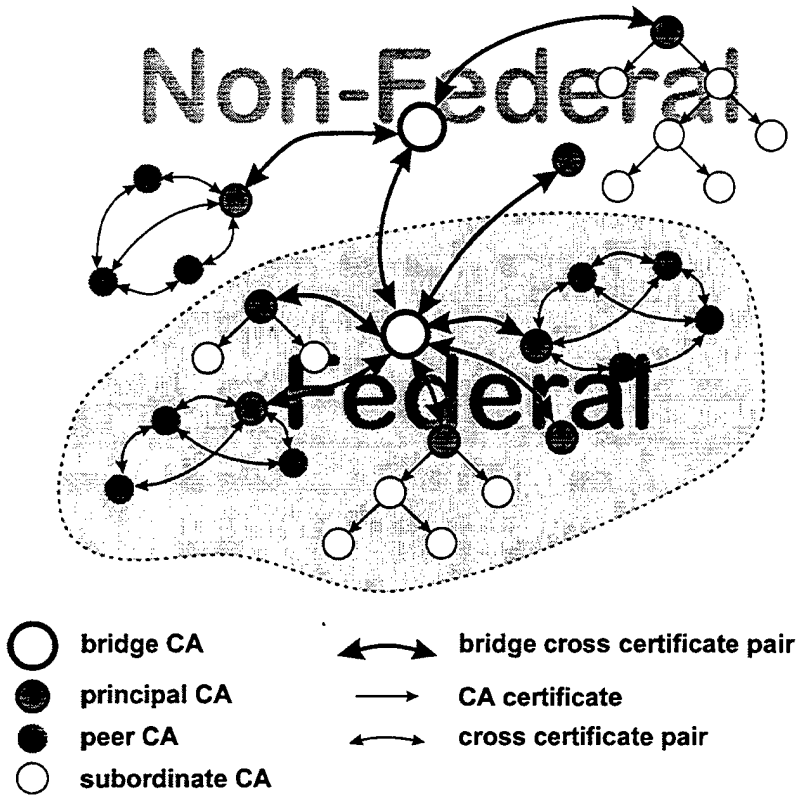

Figure 6. Proposed Federal PKI Certificate Path Architecture

Source: (Burr, W.E., September 1998, p. 21)

Note that the bridge CA does not have a traditional domain; its domain is limited to itself. Instead, the BCA's subscribers are all principal CAs within their own domains. It is envisioned that the Federal Bridge CA will only issue cross-certificates, and possibly its own self-certificate.

b) Border Directory Architecture

Although a certificate path is necessary, it is not sufficient to perform certificate path validation. Path processing also requires that relying parties have access to certificate-status information from each domain in the path. Since different domains can employ different operational protocols, the FPKI must provide access to this information

without requiring the path processing agents to handle multiple directory access protocols. Additionally the BCA will require at least one repository to post certificates issued to or by the BCA, FPKI policy statements, BCA certificate practice statements, and the BCA's CRL. The proposed FPKI border directory architecture address these requirements with an FII certificate-management repository comprised of a BCA directory server and a set of border directory servers.

Figure 7 depicts this architecture using three domains, each with a different internal directory structure, interconnecting with the BCA directory server. The proposed border directory architecture requires domains using alternative directory services to provide a border directory server that will translate their internal operational protocol to either X.500 or LDAP so that they can communicate with the BCA Directory Server.

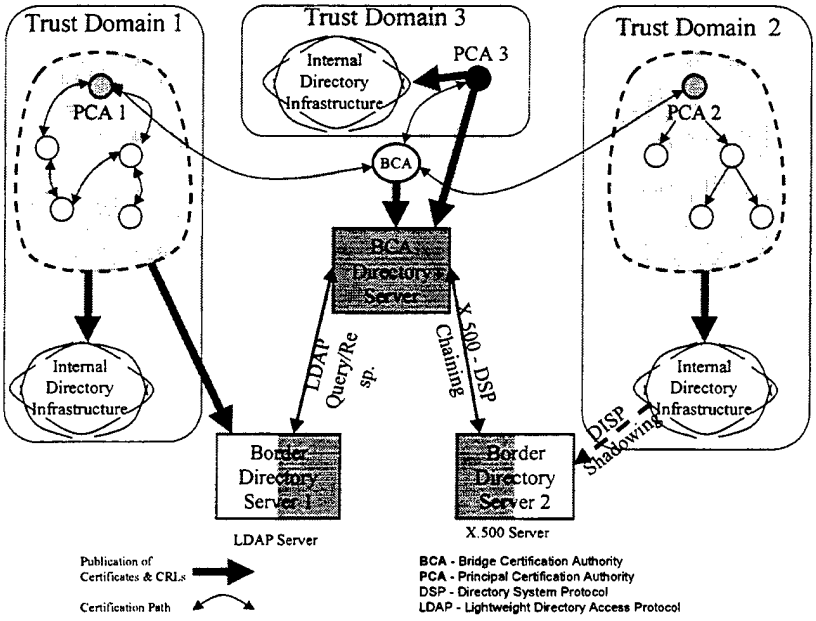

Figure 7. Proposed Federal PKI Border Directory Architecture

Source: (FPKI Border Directory CONOPS, 1999, p. 5.)

The current draft of the *Federal PKI Directory Concept of Operations* explains the architecture as follows:

> Trust Domain 1 publishes certificate information to border directory server 1 through any protocol it chooses. The border directory server is provided by the trust domain and supports the Lightweight Directory Access Protocol (LDAP) for queries and responses from other trust domains.
>
> Trust Domain 2 publishes information from its internal directory to its border directory server using the X.500 Directory Information Shadowing Protocol (DISP). The border directory server supports the X.500 Directory System Protocol (DSP), using chaining to support queries and responses from other trust domains.
>
> Trust Domain 3 does not have a separate border directory server. Instead, the PCA is responsible for posting certificate information directly to the BCA directory server. (FPKI Border Directory CONOPS, 1999, pp. 5-6.)

G. U.S. GOVERNMENT'S INTEROPERABILITY LESSONS

The U.S. Federal government's PKI experience to date offers several interoperability lessons to other early adopters of PKI technology.

1. Carefully Specify Assurance Levels

Although most enterprises define the security and privacy services they need from their PKI, they do not specify the level of assurance they need. Although the DoD's valuation of information may not map well to other organizations, the metric is not as important as the process. Establishing the value of the information to be protected by a PKI is critical to making a prudent selection of the level of assurance needed. Subsequent selection of the assurance level needed is necessary before functional interoperability can be meaningfully established with other domains.

2. Understand Risks of Custom PKI Implementations

When an enterprise decides to implement a custom PKI, it is very likely that this decision will complicate interoperability. The FORTEZZA experience is a good example. Since interoperability between computer networks is often a utility multiplier for the system, the potential sacrifice of interoperability should be carefully evaluated prior to implementing a custom PKI.

3. Match Topology to Organizational Structure

Whenever possible, the selection of PKI topology should reflect the organizational structure of the community it is intended to support. The DoD's culture and hierarchical organizational structure makes its use of a hierarchical PKI topology appropriate. A mismatch between PKI topology and

the organizational structure of the PKI's users may impede functional interoperability.

4. Carefully Select a PKI Interoperability Model

Interoperability should be a critical distinguishing characteristic when purchasing PKI products. The DoD experience has demonstrated the limitations of the assimilation model for interoperability. Similarly, if the DoD BCA is successful, it may prove to be an excellent model for providing interoperability among legacy PKIs.

VIII. CONCLUSIONS AND RECOMMENDATIONS

In this thesis interoperability is defined as the capacity of a PKI to provide both security and privacy services at an established level of assurance across domain boundaries. Although previous work had largely developed the elements of technical interoperability, the elements of functional interoperability for a PKI are new. Finally, the thesis addresses PKI interoperability within a unified framework with anecdotal references from the DII and FII to technical and functional interoperability.

A. FUNDAMENTAL ASSUMPTIONS

Since PKI interoperability is defined in terms of a PKI's ability to support trust between domains, the basic assumptions that underpin the technology of public key cryptography must be recognized. These include the intractability of the cryptographic algorithm's "hard problem," the protection of private keys, the physical security provided to the infrastructure, and operating system security. Invalidating any of these assumptions has the potential to compromise PKI security.

Absolute security does not exist because security resources are inherently limited, and it is impossible to completely control the environment in which a system resides. As a result, PKI engineering decisions, including those that impact on interoperability, are made in a risk management environment. PKIs are not a silver bullet; but they can support security and privacy in the context of an overall information assurance strategy. The DoD's use of PKI technology as an element of its "defense in depth" information assurance strategy illustrates both the limits and utility of PKIs.

B. INTEROPERABILITY RECOMMENDATIONS

Governments and PKI vendors have been working for years to build PKIs that provide interoperability. The following recommendations suggest a redirection of effort to areas that have not received adequate attention to date:

1. Cross Certification

As discussed in Chapter VII, the assimilation model of interoperability has limited application. Similarly, the trust list topology provides little assurance and limits interoperability for the reasons outlined in Chapter V. It follows, therefore, that a PKI that is going to support general interoperability must employ cross-certificates, either directly between CAs or via a bridge CA.

The conspicuous absence of certificate path processing capability in popular commercial software products that can handle cross-certificates is a principal impediment to interoperability. Web browsers, email clients, and other software that would benefit from PKI services need to be "PKI enabled" before the infrastructure will be commonly used. Moreover, cross-certificate processing clients in desktop software will likely stimulate consumer demand for PKI interoperability.

These PKI-enabled applications are unlikely to be robust enough to support multiple algorithms, certificate management protocols, and operational protocols. As a result, software engineers are going to be compelled to make decisions that will limit technical interoperability. If the marketplace has not selected a common suite of open standards, this requirement for "thin" clients is likely to drive a market for commercial bridge CA services to facilitate interoperability.

2. Standardized Policies

Although certificates and CRLs are profiled and PKIX has created a standard format for the topics certificate policies should address, open standards do not exist for standardized policies or levels of assurance. In order to enable automated policy processing, either a standard matrix or profile of certificate policies and assurance levels should be created. These profiles should be analogous to what the platform for privacy preferences (P3P) promises to be for Web browser privacy. (Bridis, T., 1999)

3. Independent Validation

This thesis has emphasized the importance of a systems engineering approach to providing security. Security must be made a system requirement across the development continuum from design, through implementation, to actual use. Independent validation at each stage of development is critical to provide high assurance. This continuum is often broken at the later stages of the system's life-cycle. Mechanisms must be developed to audit the practices of CAs and RAs, whether it is done as a condition of licensing or as a matter of sound business practice as is done with financial audits.

4. User Interface

As discussed in Chapter VI, PKI enabled applications must have a user interface that is operationally apparent to support functional interoperability. Although a "plug-and-play" approach is attractive, it is critical that the software is correctly configured for its PKI. The creation of standardized policies and levels of assurance, as discussed above, will facilitate standardization of configurations. Irrespective of how the PKI is realized, users must be impressed with the importance of protecting

private keys and the need to have their trust anchors securely delivered via an out-of-band mechanism.

C. FUTURE WORK

The infancy of PKI technology and its rapid pace of change will continue to make PKI interoperability a fertile field for research. Possible topics for future work include the following:

1. Cost Recovery

Today, commercial PKI service providers, like VeriSign, Inc., sell certificates directly to subscribers. As commercial PKI implementations become more prevalent, it is possible that a usage fee will be assessed on a per certificate validation basis to relying parties for repository access. This will require a technical means to support billing that is likely to complicate technical interoperability. Similarly, the ability of a repository to generate income will have a significant impact on the functional interoperability of the PKI. As PKIs develop, the relative merits of various PKI cost recovery schemes and their impact on interoperability will need to be analyzed.

2. Certificate Lifetime

The strength of cryptographic keys decreases over time and with use. The two principal factors that affect this process are the pace of cost reduction for computer processing power and the state of the art of mathematics. As the lifetime of keys and their certificates shrink, the costs of PKI operation will increase. An analysis of the relationship between the projected intractability of an algorithm's cryptographic work function and the costs of operating a PKI as certificate lifetime declines is necessary to design for functional interoperability.

3. Key Sterilization

Asymmetric cryptographic algorithms often have keyspaces that are not uniformly secure. This means that "dirty keys" can be chosen that significantly weaken the algorithm's security. Key sterilization is a process of testing public keys prior to certificate issuance to detect insecure private keys. This is an area of advanced research that has the potential to affect functional interoperability.

4. Bridge CA Algorithm Translation

Some PKI domains use clients that can only process a single digital signature algorithm. Bridge CAs like the Federal BCA might support interoperability between domains that cannot accommodate multiple signature algorithms by issuing mixed algorithm certificates. A mixed algorithm certificate is signed with a different algorithm than the algorithm of the key being certified. The FPKI TWG has examined several possible approaches to the multiple algorithm BCA. (Burr, W.E., July 1998) Development of a multiple algorithm protocol for bridge CAs is another field for future study.

5. Consolidation Migration

As was the case with the railroad infrastructure, as PKI technology matures it is very likely that after the initial battle for marketshare there will be a consolidation of PKI vendors. This will have a profound affect on technical interoperability. Various scenario-based models have been developed to analyze technical risk for immature technologies such as PKI. Creating such a model and using it to predict technical interoperabilty trends is a subject for future research.

6. DoD International Interoperability

This decade's trend toward coalition warfare places significant interoperability demands on the DII. This challenge goes beyond the sovereignty issues discussed in Chapter VI. Yesterday's adversary may be today's coalition partner, and tomorrow they may be adversaries again. If a coalition partner is given access to DII systems, how does the DoD protect against the creation of trap doors that will allow them access later should they become adversarial? PKI interoperability in this environment is also a fertile field for future research.

APPENDIX. GLOSSARY

Arbitrated Protocol: A protocol that employs a trusted third party to participate in each transaction between sender and receiver to ensure that both sides act properly.

Attribute Certificate: A public key certificate whose subject is a non-entity.

Authentication: The process used to ascertain the identity of a subject.

Certificate (also Public Key Certificate): A certificate is a formatted and digitally signed data structure that binds a public key to a subject.

Certification Authority (CA): An entity trusted within a PKI by one or more users to issue and revoke certificates. Certificate authorities are responsible for the validity of the subject/key binding from issuance to revocation. Optionally, they may also create the keys for users.

Certificate Policy: A named set of rules that dictates its management and use within its domain.

Certification Practice Statement (CPS): A statement of the practices that a certification authority employs in issuing certificates.

Certificate Revocation List (CRL): A formatted, digital data structure generated by a certificate authority and designed to identify certificates that have been revoked or suspended prior to their expiration dates. CRLs are generally posted to a directory.

Ciphertext: The encrypted form a message whose meaning has been obscured.

Composite Number: Any integer (positive whole number) which is not prime.

Confidentiality: A security service that protects information from unauthorized disclosure.

Cryptanalysis: The study of encryption and ciphertext with the goal of "breaking" the encryption (discovering the plaintext message).

Cryptography: The practice of using encryption to conceal text.

Decipher: The inverse process to enciphering that returns ciphertext to plaintext.

Decode: The inverse process to encoding that returns ciphertext to plaintext.

Decryption: The inverse process to encryption that returns ciphertext to plaintext.

Digital Signatures: A transformation of a message using an asymmetric cryptographic system and a hash function such that a person having the initial message and the signer's public key can accurately determine if the transformation was created using the corresponding signer's private key. In addition, it can be determined if the initial message has been altered since the transformation was made.

Directory: The directory is a repository or database of certificates, CRLs, and other information available online to users.

Domain: The logical realm over which a PMA determines policy.

Encipher: The processes of individually translating the letters or symbols of a plaintext message so as to create an obscured ciphertext.

Encoding: The process of translating entire words or phrases of a plaintext message to other words or phrases so as to create an obscured ciphertext.

Encryption: The processes of either encoding or enciphering a plaintext message into ciphertext so as to obscure its meaning.

End Entity (EE): The subject of a public key certificate when the subject not a Certificate Authority or Registration Authority.

Identity Certificate: A public key certificate that binds the name of an entity (the subject) to a public key. The subject(s) of identity certificates are the entity's name and possibly other identity information.

Integrity (also Data Integrity): The security service that protects information from unauthorized modification.

Interoperability: The capacity of a public key infrastructure to support trust by retaining its security services accross domaims at an established level of assurance.

Key: The correspondence between the elements of the plaintext and ciphertext alphabets that is employed by the encryption algorithm to perform encryption or decryption.

Local Registration Authority (LRA): *See Registration Authority.*

Non-Repudiation: Strong and substantial evidence of the identity of the signer of a message and of message integrity, sufficient to prevent a party from successfully denying the origin, submission or delivery of the message and the integrity of its contents.

Organizational Registration Authority (ORA): *See Registration Authority.*

Permutation: The reordering of the elements of a set such as the characters or symbols contained in plaintext. Also known as transposition.

Plaintext (also cleartext): The original, easily discernable, form of a message.

Policy Management Authority (PMA): The entity (owner) that determines policy and accepts liability for a given PKI domain.

Prime Number: An integer (positive whole number) that is divisible (with remainder zero) by only 1 and itself.

Private Key: The key of a key pair to be safeguarded by the owner. A private key is used to generate a digital signature. Private keys are used to decrypt information, including key encryption keys during key exchange. It is computationally infeasible to determine a private key given the associated public key.

Protocol: An unambiguous, complete, pre-established and mutually subscribed sequence of steps taken by two or more parties to accomplish a task.

Public Key: The key of a key pair released to the public. The signer's public key is used to verify a digital signature. Public keys are used for encryption, including privacy keys during key exchange.

Public Key Certificate: *See also Certificate.*

Public Key Infrastructure (PKI): The underlying system designed to support public key cryptography services comprised of the applications, policies, standards and laws which govern the generation, storage, distribution and management of cryptographic keys and digital certificates. PKI's exist to propagate trust across computer networks by providing a defined set of security services with an established level of assurance. These security services can include confidentiality, integrity, authentication, non-repudiation, key-recovery and access control.

Registration Authority (RA): The person who is responsible to the CA for local (onsite) identification of end-entity's identity.

Relying Party: Any user of a public key infrastructure attempting to establish a trust path to the relying party's trust anchor.

Root Certification Authority: The topmost trusted entity within a hierarchial PKI domain which is responsible for establishing and managing the PKI domain by issuing CA certificates to entities it authorizes and trusts to perform CA functions. The Root CA is the ultimate trust anchor for its domain.

Subject: The entity (CA, RA or EE) named in a certificate. Subjects can be persons, organizations, computers (as represented by DNS names, an email address or IP addresses), software agents or some attribute such as an account authorization or computer resource access.

Subordinate Certificate Authority: A CA that is not the trust anchor for the relying party in question.

Substitution: The exchange of the meaning of one letter or symbol for another.

Token: A physical device (e.g. *floppy diskette, smart card, PC Card, etc.*) which is used to protect and transport the private keys of a user.

Trust Anchor (also Most-Trusted Certificate Authority): the CA whose self-signed certificate is directly trusted by a relying party. The secure transfer of the trust anchor's public key to a relying party must be accomplished by a secure, out-of-band protocol.

LIST OF REFERENCES

Abrams, M. and Podell, H., *Cryptography*, as printed in *Information Security: An Integrated Collection of Essays*, IEEE Computer Society Press, 1995, pp.350-384.

Adams, C., Cain, P., Pinkas, D., and Zuccherato, R., "Internet X.509 Public Key Infrastructure Time Stamp Protocols," Internet Draft, *The Internet Society*, [http://www.ietf.org/internet-drafts/draft-ietf-pkix-time-stamp-02.txt], June 1999.

Adams, C. and Meyers M., "Internet X.509 Public Key Infrastructure Certificate Management Message Formats," Internet Draft, *The Internet Society*, [http://www.ietf.org/internet-drafts/draft-ietf-pkix-cmmf-02.txt], July 1998.

Adams, C. and Farrell, S., "Internet X.509 Public Key Infrastructure Certificate Management Protocols," RFC 2510, *The Internet Society*, [http://www.ietf.org/rfc/rfc2510.txt], March 1999.

"Advanced Encryption Standard (AES) Development Effort," NIST, [http://csrc.nist.gov/encryption/aes/aes_home.htm] August 1999.

American Bar Association, Section of Science and Technology, Information Security Committee, "Digital Signature Guidelines Tutorial," [http://www.abanet.org/scitech/ec/isc/dsg-tutorial.html], August 1998.

"ANX Service Overview," [http://www.anxo.com/TP_SP/SP_Service_Overview.htm], August 1999.

Arsenault, A. and Turner, S., "Internet X.509 Public Key Infrastructure (PKIX) Roadmap," Internet Draft, *The Internet Society*, [http://www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-02.txt], June 23, 1999.

Atkins, D., Stallings, W., and Zimmermann, P., "PGP Message Exchange Formats," RFC 1991, *The Internet Society*, [http://www.ietf.org/rfc/rfc1991.txt], August 1996.

Beth, T., Frisch, M., and Simmons, G., *Public Key Cryptography: State of the Art and Future Directions*, Springer-Verlag, 1992.

Boeyen, S., Howes, T., and Richard, P., RFC 2559, "Internet X.509 Public Key Infrastructure Operational Protocols -- LDAPv2," *The Internet Society*, [http://www.ietf.org/rfc/ rfc2559.txt], April 1999.

Boeyen, S., Howes, T., and Richard, P., RFC 2587, "Internet X.509 Public Key Infrastructure LDAPv2 Schema," *The Internet Society*, [http://www.ietf.org/rfc/ rfc2587.txt], June 1999.

Branchaud, Marc, *A Survey of Public-Key Infrastructures*, Masters Thesis, Department of Computer Science, McGill University, Montreal, [http://www1.xcert.com/ ~marcnarc//PKI/thesis/], March 1997.

Bridis, T., "Microsoft Corp. will help protect privacy of Web users," The Monterey County Herald, p. A5.

Brinkley, D. and Schell, R., *Concepts and Terminology for Computer Security*, as printed in *Information Security: An Integrated Collection of Essays*, IEEE Computer Society Press, 1995, pp. 40-97.

Bruno, L. "Certificate Authorities: Who Do You Trust?" *CMPnet*, [http://www.data.com/roundups/certificate.html], 21 March 1998.

Bruno, L. "Internet Security: How Much is Enough?" *CMPnet*, [http://www.data.com/roundups/how_much_is_enough.html], April 1996.

Burr, W. E., "Multiple Algorithms and the Bridge CA Concept," Tecnical Working Group of the Federal Public Key Infrastructure Steering Committee, (TWG-44-98), [http://csrc.nist.gov/pki/twg/papers/twg-98-44.pdf], July 24, 1998.

Burr, W. E., "Public Key Infrastructure (PKI) Technical Specifications: Part A - Technical Concept of Operations" Tecnical Working Group of the Federal Public Key Infrastructure Steering Committee, (TWG-59-98), [http://csrc.nist.gov/pki/twg/baseline/PKICON20b.doc], September 4, 1998.

Burr, W., Dodson, D., Nazario, N., and Polk, W.T., "MISPC, Minimum interoperability specification for PKI Components, Version 1," NIST Special Publication 800-15, [http://csrc.nist.gov/pki/documents/welcome.html], September 3, 1997.

Callas, J., Donnerhacke, L., Finney, H., and Thayer, R., "OpenPGP Message Format," RFC 2440, *The Internet Society*, [http://www.ietf.org/rfc/rfc2440.txt], November 1998.

Cavanaugh, K., "Is the Digital Pen Mightier Than Cryptography," *CyberTimes*, [http://search.nytimes.com /books/search/bin/fastweb?getdoc+cyber-lib+cyber-lib+8603+24+wAAA+cryptography], April 16, 1997.

Certicom Corp., "An Introduction to Information Security," [http://www.certicom.ca/ecc/ wecc1.htm], March 1997.

Certicom Corp., "Current Public-Key Cryptographic Systems," [http://www.certicom.ca/ ecc/ wecc2.htm], April 1997.

Certicom Corp., "Remarks on the Security of the Elliptic Curve Cryptosystem," [http://www.certicom.ca/ecc/ wecc3.htm], September 1997.

Chokhani, S., "Public Key Infrastructure(PKI) Compatibilitiy," Version 0.5 (draft), CygaCom Solutions, Inc, November 10, 1997.

Chokhani, S. and Ford, W., "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," RFC 2527, *The Internet Society*, [http://www.ietf.org/rfc/rfc2527.txt], March 1999.

"CME's Cryptography Timeline," [http://www.clark.net/pub /cme/html/timeline.html], March 1998.

Diffie, W. and Hellman, M.E. "New Directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. 22, No. 6, November 1976.

Denning, D. E., "Descriptions of Key Escrow Systems," [http://www.cosc.georgetown.edu/~denning/crypto/ Appendix.html], February 26, 1997.

Denning, D. E. and Branstad, D. K., "A Taxonomy for Key Escrow Encryption Systems," *Communications of the ACM*, Vol. 39, No. 3, March 1996 or [http://www.cosc.georgetown.edu/ ~denning/crypto/Taxonomy.html].

Devore, J.L., *Probability and Statistics for Engineering and the Sciences*, Fouth Edition, Duxbury Press, 1995.

Dunn, A., "Of Keys, Decoders and Personal Privacy," *CyberTimes*, [http://www.nytimes.com/ library/cyber /surf/100197mind.html], October 1, 1997.

Ellison, C., Frantz, B., Lampson, B., Rivest, R. Thomas, B., and Ylonen, B., "SPKI Certificat Theory," Internet Draft, *The Internet Society*, [http://www.ietf.org/internet-drafts/draft-ietf-spki-cert-theory-05.txt], May 28, 1999.

"Federal Public Key Infrastructure (PKI) Directory Concept of Operations," Tecnical Working Group of the Federal Public Key Infrastructure Steering Committee, (TWG-99-29), [http:// csrc.nist.gov/pki/twg/papers/twg-99-29.pdf], April 20, 1999.

Federal Public Key Infrastructure Steering Committee, "Access With Trust," Government Information Technology Services Board, Office of Management and Budget, [http://gits-sec.treas.gov/gits-sec-home.htm], September 1998.

"Federal Public Key Infrastructure Steering Committee Charter," Government Information Technology Services Board, Office of Management and Budget [http://gits-sec.treas.gov/oofpkicharter.htm], June 16, 1999.

"Federal Public Key Infrastructure (PKI) Technical Specification: Part D - Interoperabilty Profiles," Tecnical Working Group of the Federal Public Key Infrastructure Steering Committee, December 15, 1995.

"Federal Public Key Infrastructure (PKI) X.509 Certificate and CRL Extensions Profile," Tecnical Working Group of the Federal Public Key Infrastructure Steering Committee, (TWG-99-01), [http:// csrc.nist.gov/pki/twg/papers/twg-99-01.pdf], January 4, 1999.

Fillingham, D., "Managing Interoperabilty and Security across the MISSI Security Management Infrastructure Boundry," National Security Agency (X33)., December 3, 1996.

Ford, W. and Baum, M., *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption*, Prentice Hall PTR, 1997.

Garfinkel S. and Spafford, G., *Practical UNIX and Internet Security*, 2nd Edition, O'Reilly & Associates, Inc., 1996.

"Guidelines for External Certification Authority Interoperability with the Department of Defense Public Key

Infrastructure (Draft)" Cygnacom Solutions, Version 0.5, January 26, 1999.

Ha, K.O., "Oceans apart on privacy," *San Jose Mercury News*, October 26, 1998, p. 1E and 8E.

Hamre, J.J., "Department of Defense (DoD) Public Key Infrastructure" Deputy Secretary of Defense Memorandum, [ftp://infosec.nosc.mil/pub/docs/navy/PKI/PKI_Policy.pdf], May 6, 1999.

Housley, R., Ford, W., Polk, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," RFC 2459, *The Internet Society*, [http://www.ietf.org/rfc/rfc2459.txt], January 1999.

Housley, R. and Hoffman, P., "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP," RFC 2585, *The Internet Society*, [http://www.ietf.org/rfc/rfc2585.txt], May 1999.

Jurisic, A. and Menezes, A., "Elliptic Curves and Cryptography," [http://www.certicom.ca/ecc/wecrypt.htm], March 1998.

Kent, S., *Internet Privacy Enhanced Mail*, as printed in *Information Security: An Integrated Collection of Essays*, IEEE Computer Society Press, 1995, pp. 405-422.

"Key Recovery Tutorial," [http://gits-sec.treas.gov/krdptut.htm], August 1999.

Kocher, P., "Timing attacks on Implementations of Diffe-Hellman, RSA, DSS and Other Systems," [http://www.cryptography.com/timingattack/paper.html], March 1998.

Kohnfelder, L. M., "Towards a Practical Public-key Cryptosystem," MIT S.B. Thesis, May 1978.

McBride, Baker, and Coles, "Provisions Relating to Liability of Certification Authorities in Enacted Legislation," [http://www.mbc.com/legis/table06.html], May 24, 1999.

McBride, Baker, and Coles, "Scope of Authorization to Use of Electronic Signatures in Enacted Legislation," [http://www.mbc.com/legis/table01.html], May 24, 1999.

Merkow, M., "Growing a Tree of Trust -- Part One," *The Electronic Commerce Guide, internet.com,* [http://ecommerce .internet.com/opinions/merkow/article/0,1281,5561_125791,00. html], December 31, 1998.

Merkow, M., "Growing a Tree of Trust -- Part Two," *The Electronic Commerce Guide, internet.com,* [http://ecommerce .internet.com/opinions/merkow/article/0,1281,5561_125821,00. html], January 14, 1999.

Merkow, M., "Growing a Tree of Trust -- Part Three," *The Electronic Commerce Guide, internet.com,* [http://ecommerce .internet.com/opinions/merkow/article/0,1281,5561_125771,00. html], January 28, 1999.

Merkow, M., "Growing a Tree of Trust -- Part Four," *The Electronic Commerce Guide, internet.com,* [http://ecommerce .internet.com/opinions/merkow/article/0,1281,5561_125761,00. html], February 18,1999.

Meyers, M., Adams, C., Solo, D., and Kemp, D., "Internet X.509 Certificate Request Message Format," RFC 2511, *The Internet Society,* [http://www.ietf.org/rfc/rfc2511.txt], March 1999.

Muftic, S., *Security Mechanism for Computer Networks,* Ellis Horwood Limited, 1989.

National Institute of Standards and Technology, U.S. Department of Commerce, *Guideline for the Use if Advanced Authentication Technology Alternatives,* FIPS PUB 190, 1994.

Pfitzmann, B., *Digital Signature Schemes: General Framework and Fail-Stop Signatures,* Lecture Notes in Computer Science vol. 1100, Springer-Verlag, 1996.

Pfleeger, C., *Security in Computing,* Prentice-Hall, Inc., 1989.

Pinsky, L., "Digital Signatures: A Sign Of The Times," [http://www.digsigtrust.com/resources /lawrence-pinsky.html], 1997.

Pompili, T., "Evolving Internet Security Methods," PC Magazine, [http://www.zdnet.com/pcmag/issues/1508 /pcmg0076.html], April 23, 1996.

"Public Key Infrastructure Roadmap for the Department of Defense" Version 2.0, Revision C, Department of Defense, May 6, 1999.

Ritter, T., "The Fenced DES Cipher -- Stronger Than DES But Made From DES," [http://www.io.com/~ritter/FENCED.HTM], November 10, 1996.

Schneier, B. and Banisar, D., *The Electronic Privacy Papers: Documents on the Battle for Privacy in the Age of Surveillance*, John Wiley & Sons, Inc., 1997.

Schneier, B., *Applied Cryptography, Second Edition*, John Wiley & Sons, Inc., 1996.

Schneier, B., "Why Cryptography Is Harder Than It Looks," [http://www.counterpane.com/whycrypto.html], 1996.

Schneier, B., "Factoring -- State of the Art and Predictions," [http:www.cs.hut.fi/crypto/rsa-key-length-recommendations], 12 February 1995.

Sloane, N.J.A. and Wyner, A., *Claude Elwood Shannon, Collected Papers*, IEEE Press, 1992.

"United States Department of Defense X.509 Certificate Policy", Version 2.0, Department of Defense, [http://csrc.nist.gov/pki/twg/dod.htm], March 1999.

"Utah Digital Signature Program," [http://www.commerce.state.ut.us/web/commerce/digsig/dsmain.htm], March 1998.

Wahal, M., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (V3)," RFC 2251, *The Internet Society*, [http://www.ietf.org/rfc/rfc2251.txt], December 1997.

Wayner, P., "A Patent falls, and the Internet Dances" *CyberTimes*, [http://www.nytimes.com/library/cyber/week/090697patent.html], September 6, 1997.

Wayner, P., "British Document Outlines Early Encryption Discovery" *CyberTimes*, [http://search.nytimes.com/books/search/bin/fastweb?getdoc+cyber-lib+cyber-lib+18574+0+wAAA+cryptography], December 24, 1997.

Wayner, P., "PGP Offers New Encryption Software for Corporations" *CyberTimes*, [http://www.nytimes.com/library/cyber/week/100397pgp.html], October 3, 1997.

Welsh, D., *Codes and Cryptography*, Oxford University Press, 1988.

Whittle, R., "Public Key Authentication Framework: Tutorial," [http://www.ozemail.com.au/~firstpr/crypto/pkaftute.htm], 2 June 1996.

Yeong, Y., Howes, T., and Kille, S., "Lightweight Directory Access Protocol," RFC 1777, *The Internet Society*, [http://www.ietf.org/rfc/rfc1777.txt], March 1995.

Ylonen, T., "International Cryptography Pages," [http://www.cs.hut.fi/crypto/], August 1998.

Zimmermann, P., *The Official PGP User's Guide*, MIT Press, 1995.

INITIAL DISTIBUTION LIST

1. Defense Technical Information Center..... 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library..... 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Dean Dan Boger 1
   CODE IW
   Naval Postgraduate School
   Monterey, California 93943-5118

4. Professor James Bret Michael..... 2
   CODE CS/Mj
   Naval Postgraduate School
   Monterey, California 93943-5118

5. Professor Timothy J. Shimeall..... 2
   CODE CS/St
   Software Engineering Institute
   4500 Fifth Ave
   Pittsburgh, Pennsylvania 15213

6. Mr. Don Heckman (V51)..... 1
   National Security Agency
   9800 Savage Road
   Fort George G. Meade, Maryland 20755-6000

7. Mr. Gary Dahlquist (V51)..... 1
   National Security Agency
   9800 Savage Road
   Fort George G. Meade, Maryland 20755-6000

8. Mr. Al Arsenault (V51) 1
   National Security Agency
   9800 Savage Road
   Fort George G. Meade, Maryland 20755-6000

9. Mrs. Gloria Serrao 1
   National Security Agency
   9800 Savage Road
   Fort George G. Meade, Maryland 20755-6000

10. Mr. William E. Burr 1
National Institute of Standards and Technology
100 Bureau Dr. Stop 8930
Gaithersburg, Maryland 20899-8930

11. LCDR Paul Friedrichs,USN 1
IAESO
5600 Columbia Pike
Falls Church, Virginia 22041

12. CNO Staff N643 1
Presidential Tower 1 Suite 5412
2511 South Jefferson Davis Hwy
Arlington, Virginia 22202

13. CAPT Dan Galik, USN 1
SPAWARSYSCOM PMW-161
4301 Pacific Highway
San Diego, California 92110-3127

14. CNSG N6 1
Naval Security Group Headquarters
9800 Savage Road
Fort George G. Meade, Maryland 20755-6000

15. CAPT J. T. Daly, USN 1
Commanding Officer
Naval Information Warfare Activity
9800 Savage Road
Fort George G. Meade, Maryland 20755-6000

16. Fleet Information Warfare Center 1
Attn: N5
2555 Amphibious Drive
Norfolk, Virginia 23521

17. LT Anthony Hansen, USN 2
P.O. Box 230551
Centreville, Virginia 20120-9997