NIST Special Publication 800-21

Guideline for Implementing Cryptography in the Federal Government

Annabelle Lee

Security Technology Group
Computer Security Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

November, 1999

U.S. Department of Commerce
William M. Daley, Secretary

Technology Administration
Dr. Cheryl L. Shavers, Under Secretary of Commerce for Technology

National Institute of Standards and Technology
Raymond G. Kammer, Director

Form SF298 Citation Data

Report Date *("DD MON YYYY")* 01111999	**Report Type** N/A	**Dates Covered (from... to)** *("DD MON YYYY")*

Title and Subtitle Guideline for Implementing Cryptography in the Federal Government	**Contract or Grant Number**
	Program Element Number
Authors	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) IATAC Information Assurance Technology Analysis Center 3190 Fairview Park Drive Falls Church VA 22042	**Performing Organization Number(s)**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Monitoring Agency Acronym**
	Monitoring Agency Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract	
Subject Terms	
Document Classification unclassified	**Classification of SF298** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** unlimited
Number of Pages 138	

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	11/1/99	Report

4. TITLE AND SUBTITLE
Guideline for Implementing Cryptography in the Federal Government

5. FUNDING NUMBERS

6. AUTHOR(S)
Not provided

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Information Assurance
Technology Analysis Center
(IATAC)
3190 Fairview Park Drive
Falls Church, VA 22042

8. PERFORMING ORGANIZATION REPORT NUMBER

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

Defense Technical
Information Center
DTIC-AI
8725 John J. Kingman Road,
Suite 944

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE
A

13. ABSTRACT ***(Maximum 200 Words)***

The purpose of this document is to provide guidance to Federal agencies on how to select cryptographic controls for protecting Sensitive Unclassified 1 information. This document focuses on Federal standards documented in Federal Information Processing Standards Publications (FIPS PUBs) and the cryptographic modules and algorithms that are validated against these standards. However, to provide additional information, other standards organizations, (e.g., American National Standards Institute (ANSI) and International Organization for Standardization (ISO)) are briefly discussed.

14. SUBJECT TERMS
Cryptography

15. NUMBER OF PAGES

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	Unlimited

GUIDELINE FOR IMPLEMENTING CRYPTOGRAPHY IN THE FEDERAL GOVERNMENT

CHAPTER 1

1. INTRODUCTION

1.1. Purpose

In today's world, both private and public sectors depend upon information technology systems to perform essential and mission-critical functions. In the current environment of increasingly open and interconnected systems and networks, network and data security are essential for the optimum use of this information technology. For example, systems that carry out electronic financial transactions and electronic commerce must protect against unauthorized access to confidential records and unauthorized modification of data.

Cryptography should be considered for data that is sensitive, has a high value, or represents a high value if it is vulnerable to unauthorized disclosure or undetected modification during transmission or while in storage. Cryptographic methods provide important functionality to protect against intentional and accidental compromise and alteration of data. These methods support communications security by encrypting the communication prior to transmission and decrypting it at receipt. These methods also provide file/data security by encrypting the data prior to placement on a storage medium and decrypting it after retrieval from the storage medium.

The purpose of this document is to provide guidance to Federal agencies on how to select cryptographic controls for protecting Sensitive Unclassified[1] information. This document focuses on Federal standards documented in Federal Information Processing Standards Publications (FIPS PUBs) and the cryptographic modules and algorithms that are validated against these standards. However, to provide additional information, other standards organizations, (e.g., American National Standards Institute (ANSI) and International Organization for Standardization (ISO)) are briefly discussed.

1.2. Audience

This document is intended for Federal employees, who are responsible for designing systems, and procuring, installing, and operating security products to meet identified security requirements. This document may be used by:

[1] Hereafter referred to as sensitive information. In the Computer Security Act of 1987, Congress assigned responsibility to the National Institute of Standards and Technology (NIST) for the preparation of standards and guidelines for the security of sensitive *Federal* systems. Excluded are classified and sensitive national security-related systems.

- A manager responsible for evaluating an existing system and determining whether cryptographic methods are necessary,

- A technical specialist requested to select one or more cryptographic methods/techniques to meet a specified requirement, or

- A procurement specialist developing a solicitation for a system or network that will require cryptographic methods to perform security functionality.

The goal is to provide these individuals with sufficient information to allow them to make informed decisions about the cryptographic methods that will meet their specific needs to protect the confidentiality, authentication, and integrity of data that is transmitted and/or stored in a system or network.

This document is **not** intended to provide information on the Federal procurement process or provide a technical discussion on the mathematics of cryptography and cryptographic algorithms.

1.3. Scope

This document limits its discussion of cryptographic methods to those that meet Federal standards. (The majority of the information in this guideline may be useful to both Federal and commercial personnel and applicable to all computer networks and environments.) Both the Federal government and industry use products that meet Federal standards and standards bodies such as ANSI have also adopted Federal standards.

This guideline provides information on selecting cryptographic services and methods and implementing the methods in new or existing systems. Specifically, the guideline includes discussions of the following:

- The cryptographic products selection process. This may include one or more of the following:

    1. Performing a *risk assessment* (or other process) to identify the:
        - assets that must be protected,
        - vulnerabilities of the system, and
        - threats that might exploit the vulnerabilities.

    2. Identifying the *security regulations and policies* that are applicable to the system.

    3. Specifying the *cryptographic security requirements*.

    4. Specifying the *security services* that will address the needs identified in items 1 through 3 above.

- Implementation issues, including:

    - implementation approach,
    - life cycle management of cryptographic components,
    - training for users, operators, and system engineers,
    - key management,
    - authentication techniques, and
    - testing – certification, independent verification and validation (IV&V).

1.4. Content

The guideline is divided into three parts. Part one provides an overview of selecting cryptographic services and products:

- Chapter 1 includes background information (purpose, audience, and scope) and advantages of using cryptography.

- Chapter 2 defines the role and use of standards, describes standards organizations that are outside the Federal government, and discusses the new international security standard, the Common Criteria.

- Chapter 3 describes some implementation issues (e.g., key management, authentication, and recommendations).

Part two focuses on specific methods:

- Chapter 4 describes the methods that are available for symmetric and asymmetric key cryptography.

- Chapter 5 discusses the Public Key Infrastructure (PKI).

- Chapter 6 discusses testing, including the Cryptographic Module Validation Program (CMVP).

Part three ties all of the information together:

- Chapter 7 describes the process of choosing types of cryptography and selecting a cryptographic method or methods to fulfill a specific requirement.

- Chapter 8 includes some examples of Federal projects that use cryptography.

- Chapter 9 describes future activities.

There are three appendixes to the guideline:

- Appendix A includes an acronym list.

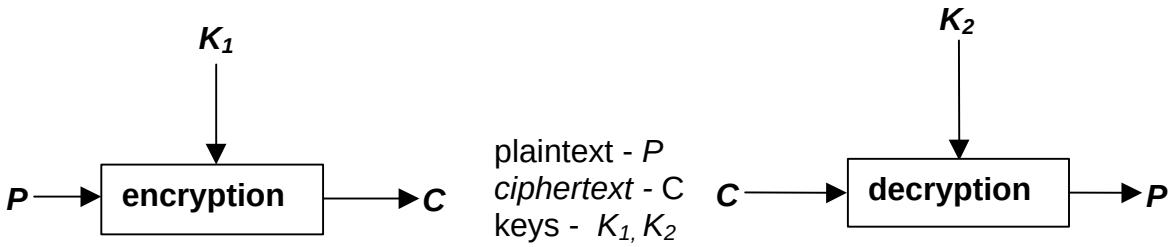
- Appendix B includes terms and definitions.

- Appendix C includes a bibliography of cryptographic standards and guidelines and cryptography texts.

A number of examples are included throughout this guideline. Each example is displayed in a shaded box for ease of viewing.

1.5. Uses of Cryptography

Cryptography is a branch of mathematics based on the transformation of data. Cryptography deals with the transformation of ordinary text (plaintext) into coded form (ciphertext) by encryption and the transformation of ciphertext into plaintext by decryption. Cryptography relies upon two basic components: an *algorithm* (or cryptographic methodology) and a *key*. The algorithm is the mathematical function used for encryption or decryption, and the key is the parameter used in the transformation. These transformations are illustrated in Figure 1.



(Note: K_1 and K_2 may be the same key or different keys)

Figure 1. Data Transformation

There are two basic types of cryptography: *secret key* systems (also called symmetric systems) and *public key* systems (also called asymmetric systems). In secret key systems, the same key is used for both encryption and decryption. That is, all parties participating in the communication share a single key. In public key systems, there are two keys: a *public* key and a *private* key. The public key used for encryption is different from the private key used for

decryption. The two keys are mathematically related, but the private key cannot be determined from the public key.

In general, cryptography is used to meet the following security objectives:

- *Confidentiality* services restrict access to the content of sensitive data to only those individuals who are authorized to view the data. Confidentiality measures prevent the *unauthorized* disclosure of information to unauthorized individuals or processes.

- *Data integrity* services address the unauthorized or accidental modification of data. This includes data insertion, deletion, and modification. To ensure data integrity, a system must be able to detect *unauthorized* data modification. The goal is for the receiver of the data to verify that the data has not been altered.

- *Authentication* services establish the validity of a transmission, message, or an originator. (Authentication services also verify an individual's authorization to receive specific categories of information. These services are not specific to cryptography.) Therefore, this service applies to both individuals and the information itself. The goal is for the receiver of the data to determine its origin.

- *Non-repudiation* services prevent an individual from denying that previous actions had been performed. The goal is to ensure that the recipient of the data is assured of the sender's identity.

CHAPTER 2

2. STANDARDS AND CRITERIA

Under the Information Technology Management Reform Act of 1996 and the Computer Security Act (CSA) of 1987 (Public Law 100-235), the National Institute of Standards and Technology (NIST) is responsible for developing technical standards and guidelines for Federal information resources. In addition, Appendix III to Office of Management and Budget (OMB) Circular No. A-130 - *Security of Federal Automated Information*, in part, establishes a minimum set of controls to be included in Federal automated information security programs and assigns Federal agency responsibilities for the security of automated information. The Appendix incorporates requirements of the Computer Security Act of 1987.

Some of the standards and guidelines used to protect sensitive information are issued by NIST as FIPS PUBs. Federal agencies must comply with all mandatory standards and they are expected to:

- Support the development of such standards,

- Avoid the creation of different standards for government and the private sector, and

- Use voluntary standards whenever possible,

Technically, NIST has authority to establish standards only for the Federal government. However, FIPS PUBs have a profound effect on commerce and industry. Since FIPS PUBs are established through a public process, the public is aware of their existence, and industry often uses conformance to applicable NIST standards as an evaluation factor when purchasing products. Also, NIST has a long history of participation in industry standards groups, including ANSI, ISO, Institute of Electrical and Electronics Engineers (IEEE), Internet Engineering Task Force (IETF), and others. In some cases, the Federal government adopts industry standards (ANSI X9.17 *Key Management* was adopted with restrictions as FIPS PUB 171), and industry has adopted FIPS PUBs (e.g., Data Encryption Standard (DES) and DES Modes were adopted by ANSI).

Standards contain consistent technical specifications or other criteria to be used as rules or guidelines to ensure that products, processes and services are appropriate for their stated purpose.

2.1. Benefits of Standards

Standards are important because they define common practices, methods, and measures/metrics. Therefore, standards increase the reliability and effectiveness of products and ensure that the products are produced with a degree of quality. Standards provide solutions that have been accepted by a wide community and evaluated by experts in relevant areas. By using standards, organizations can reduce costs and protect their investments in technology.

Standards provide for Information Technology (IT) interoperability, security, and integrity:

- **Interoperability.** Products developed to a specific standard may be used to provide interoperability with other products that conform to the same standard. By using the same cryptographic algorithm, data that was encrypted using vendor A's product may be decrypted using vendor B's product. The use of a common standards-based cryptographic algorithm is necessary, but may not be sufficient to ensure product interoperability. Other common standards, such as communications protocol standards, may also be necessary.

  By ensuring interoperability among different vendors' equipment, standards permit an organization to select from various available products to find the most cost-effective solution.

- **Security.** Standards may be used to establish a common approved level of security. Most agency managers are not cryptographic security experts, and, by using a FIPS approved cryptographic algorithm, a manager knows that a standard has been developed and the algorithm has been tested against this standard and the results validated by NIST. NIST validation means the algorithm has been found to be adequate for the protection of sensitive government data. In addition, most FIPS approved algorithms have gone through a significant period of public analysis and comment.

- **Integrity.** Standards may be used to assure the integrity of a product. Standards may:

  - Specify how a feature is to be implemented, e.g., the feature must be implemented in hardware.

  - Require a test or alarm to detect a malfunction.

  - Require specific documentation to assure proper implementation and product change management.

Many FIPS PUBs contain associated conformance tests and specify the conformance requirements. The conformance tests may be administered by NIST accredited laboratories and provide validation that the standard was correctly implemented in the product.

- **Common Form of Reference.** A standard may become a common form of reference to be used in evaluating vendors' products. FIPS PUB 140-1, *Security Requirements for Cryptographic Modules*, contains security and integrity requirements for *any* cryptographic module implementing cryptographic operations. FIPS PUB 140-1 establishes a common form of reference by defining four levels of security for each of eleven security attributes.

- **Cost Savings.** A standard can save a great deal of money by providing a single commonly accepted specification. Without standards, users may be required to become *experts* in every IT product that is being considered for purchase. Also, without standards, products may not interoperate with products purchased by other users. This will result in a significant waste of money or in the delay of implementing IT.

2.2. Standards Organizations

NIST develops standards that are used by vendors who are developing security products, components, and modules. These products may be purchased and used by Federal government agencies. In addition, there are other groups that develop and promulgate standards. The following organizations are briefly described below: ANSI, IEEE, IETF, and ISO.

2.2.1. American National Standards Institute (ANSI) [2]

The American National Standards Institute (ANSI) is the administrator and coordinator of the United States (U. S.) private sector voluntary standardization system. ANSI is a private, nonprofit membership organization supported by a diverse constituency of private and public sector organizations. ANSI does not itself develop American National Standards; rather it facilitates development by establishing consensus among qualified groups.

The primary goal of ANSI is the enhancement and global competitiveness of U.S. business. ANSI promotes the use of U.S. standards internationally, advocates U.S. policy and technical positions in international and regional standards

[2] The information in this section was taken from the ANSI web site: www.ansi.org.

organizations, and encourages the adoption of international standards as national standards where these meet the needs of the user community.

2.2.1.1. ANSI X9

X9 is an inter-industry user and developer of technical standards and is organized into sub-committees and working groups, as illustrated in Figure 2.

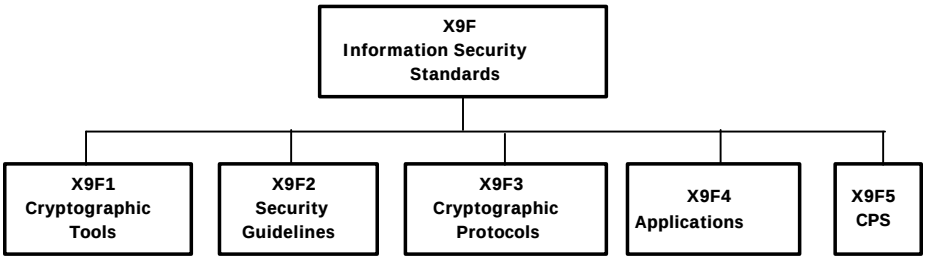


Figure 2. ANSI X9F Organization

The Accredited Standards Committee – X9 (banking) and F (security) *Financial Services* manages the development of information security and other standards for the financial services industry. The following ANSI standards are designed to support financial information infrastructures:

- Hash and signature algorithms
- Certificate management standards
- Key management and key agreement standards
- Other cryptographic methods

Table 1 lists FIPS PUBs and the corresponding ANSI standards. Some of the proposed ANSI standards may be considered for reference in existing FIPS PUBs after they have been adopted by ANSI.

Table 1. FIPS PUBs and Corresponding ANSI Standards

FIPS PUB	ANSI STANDARD
Symmetric Encryption	
DES - FIPS PUB 46-3, government tests	ANSI X3.92 - *Data Encryption Algorithm*
DES - FIPS PUB 46-3 and ANSI tests	ANSI X9.52 - *Triple Data Encryption Algorithm*, ANSI TG-19 tests (also published as NIST Special Publication (SP) 800-20)
Advanced Encryption Standard (AES) (TBD FIPS PUB and TBD government tests)	(eventual proposal to ANSI)

Table 1. FIPS PUBs and Corresponding ANSI Standards (Concluded)

FIPS PUB	ANSI STANDARD
Digital Signatures	
Digital Signature Standard (DSS) - FIPS PUB 186-2, government tests	ANSI X9.30 - Part 1: *The Digital Signature Algorithm (DSA)*
Digital Signature Standard (DSS) - FIPS PUB 186-2	ANSI X9.31 - *rDSA Signature Algorithm*, draft tests
Digital Signature Standard (DSS) - FIPS PUB 186-2	ANSI X9.62 - *Elliptic Curve Digital Signature Algorithm (ECDSA),* draft tests
Data Authentication	
Data Authentication Code (DAC) - FIPS PUB 113	ANSI X9.9 - *American National Standard for Financial Institution Message Authentication*[3]
Key Transport/Management	
Key Management Using ANSI X9.17 - FIPS PUB 171	ANSI X9.17 - *Financial Institution Key Management*[4]
(Propose adoption for government use after adopted as approved ANSI standard)	draft ANSI X9.42 - *Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, TBD tests
(Propose adoption for government use after adopted as approved ANSI standard)	draft ANSI X9.44 – *The Transport of Symmetric Algorithm Keys Using Reversible Public Key Cryptography*, TBD tests
(Propose adoption for government use after adopted as approved ANSI standard)	draft ANSI X9.63 - *Key Agreement and Key Transport Using Elliptic Curve-based Cryptography*, TBD tests
Hash Function	
Secure Hash Standard (SHS) - FIPS PUB 180-1	ANSI X9.30 - 1993 Part 2: *The Secure Hash Algorithm (SHA-1)*
Cryptographic Module Validation Program	
FIPS PUB 140-1, government tests	draft ANSI X9.66 – Cryptography Device Security

[3] This standard was withdrawn by ANSI in 1999.
[4] This standard was withdrawn by ANSI in 1999.

2.3.2. Institute of Electrical and Electronics Engineers (IEEE)[5]

The technical objectives of the IEEE focus on advancing the theory and practice of electrical, electronics and computer engineering, and computer science. The goals of IEEE activities are to: (1) enhance the quality of life for all peoples through improved public awareness of the influence and applications of its technologies and (2) advance the standing of the engineering profession and its members.

IEEE develops and disseminates voluntary, consensus-based industry standards involving leading-edge electro-technology. IEEE supports international standardization and encourages the development of globally acceptable standards.

2.2.2. Internet Engineering Task Force (IETF)[6]

The IETF is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. The actual technical work of the IETF is done in working groups, which are organized by topic into several areas (e.g., routing, transport, security, etc.). The primary role of the Security Area Directorate and the Security Area Advisory Group is to provide help to IETF working groups on how to provide for security in the protocols they design.

2.2.2.1. IETF Public-Key Infrastructure (X.509) (pkix) Working Group

Many Internet protocols and applications which use the Internet employ public-key technology for security purposes and require a public-key infrastructure (PKI) to securely manage public keys for widely-distributed users or systems. The X.509 standard constitutes a widely-accepted basis for such an infrastructure, defining data formats and procedures related to distribution of public keys via certificates digitally signed by certification authorities (CAs).

The task of the pkix working group will be to develop Internet standards needed to support an X.509-based PKI. The goal of this PKI will be to facilitate the use of X.509 certificates in multiple applications that make use of the Internet and to promote interoperability between different implementations choosing to make use of X.509 certificates. The resulting PKI is intended to provide a framework that will support a range of trust/hierarchy environments and a range of usage environments. The group will focus on tailoring and profiling the features

[5] The information in this section was taken from the IEEE web site: www.ieee.org.
[6] The information in this section was taken from the IETF web site: ietf.org.

available in the v3 X.509 certificate to best match the requirements and characteristics of the Internet environment.

2.2.3. International Organization for Standardization (ISO)[7]

ISO is a worldwide federation of national standards bodies from 100 countries. ISO is a non-governmental organization. Its mission is to promote the development of standardization and related activities in the world with a view to facilitating the international exchange of goods and services, and to developing cooperation in the spheres of intellectual, scientific, technological and economic activity. ISO's work results in international agreements that are published as International Standards.

The technical work of ISO is carried out in technical committees, subcommittees and working groups. In these committees, qualified representatives of industry, research institutes, government authorities, consumer bodies, and international organizations from all over the world come together in the resolution of global standardization problems.

2.3. Common Criteria

The *Common Criteria* (CC) is referenced throughout this guidance document. The CC represents the outcome of efforts to develop criteria for evaluation of IT security. These criteria will be used throughout the international community. The CC defines a set of IT requirements of known validity that can be used in establishing security requirements for prospective products and systems. The CC also defines the Protection Profile (PP) construct that allows prospective consumers or developers to create standardized sets of security requirements that will meet their needs. The CC presents requirements for the IT security of a product under the distinct categories of functional requirements and assurance requirements.[8]

The CC is a *voluntary* standard used to describe the security properties (functional and assurance) of IT products (or classes of products) and systems. In essence, the CC is a standard security specification "language." Products whose security properties have been specified using the CC may then be validated (tested) for conformance to their CC specifications. Such a validation, when performed by an accredited testing laboratory, confirms that the product meets its security specification(s).

In general, the FIPS PUBs referenced in this Guideline are *mandatory* standards that must be met. For example, FIPS PUB 46-3, *Data Encryption Standard*, is a

[7] The information in this section was taken from the ISO web site: www.iso.ch.
[8] This information was extracted from documents located at: csrc.nist.gov/cc/info/cc-summ.

specific set of technical security requirements for the Data Encryption Standard algorithm.

When developing a specification or criteria for selection a cryptographic module/product, both the CC and FIPS PUBs may be used. The CC may be used to specify the *functions* the algorithm will perform. The FIPS PUBs designate the specific type of algorithm (DES, DSA) and the level of independent testing required (FIPS PUB 140-1).

2.4. FIPS Waiver Procedure

Under certain exceptional circumstances, the heads of Federal agencies may approve waivers to FIPS. Waivers should be granted only when:

a. Compliance with a standard would adversely affect the accomplishment of the mission of an operator of a Federal computer system, or

b. Cause a major adverse financial impact on the operator that is not offset by Government-wide savings.

Agency heads may approve waivers only by a written decision that explains the basis on which the agency head made the required finding(s).

CHAPTER 3

3. SOME IMPLEMENTATION ISSUES

There are many issues that are applicable to the implementation of security methods/products. These are extensively discussed in other documents such as *The NIST Handbook* (SP 800-12), *Generally Accepted Principles and Practices for Security Information Technology Systems* (SP 800-14) and OMB Circular A-130, *Security of Federal Automated Information Resources*, Appendix III. Of particular relevance are the sections on training, contingency planning, assignment of roles and responsibilities, and security violation reporting and response. This chapter focuses on implementation issues that are specific to cryptography.

3.1. Interfaces/Use of CAPIs[9]

As application developers become aware of the need for cryptographic protection, they are adding "hooks" to access the cryptographic functionality developed by others. These "hooks" are known as the CAPI, or *cryptographic application programming interface*. A CAPI is an interface to a library of functions that software developers can call upon for security and cryptography services. Applications that utilize a standard CAPI can access multiple cryptographic implementations through a single interface. For example, a CAPI for confidentiality could interface with different products and algorithms without affecting the basic application. The goal of a CAPI is to make it easy for developers to integrate cryptography into applications. CAPIs can be targeted at different levels of abstraction, ranging from cryptographic module interfaces to authentication service interfaces. The goal is for general-purpose applications (e.g., spreadsheets, document processors, e-mail) to be *cryptographically* unaware, utilizing only a minimum number of high-level security calls without having to know about the underlying cryptography and security support (e.g., certificate management, key management, data isolation). Ideally, these calls would require no knowledge of specific cryptographic algorithms or modules.

3.2. Hardware vs. Software Solutions

The trade-offs among security, cost, simplicity, efficiency, and ease of implementation need to be evaluated. Cryptography can be implemented in hardware, software and/or firmware - each has its related costs and benefits.

Historically, software has been less expensive and slower than hardware, although for large applications, hardware may be less expensive. In addition, software is easier to modify or bypass than equivalent hardware products. The

[9] The information in this section was extracted from the NSA Report, *Security Service API: Cryptographic API Recommendation Second Edition.*

advantages of software solutions are in flexibility and portability, ease of use, and ease of upgrade.

In many cases, cryptography is implemented in a hardware device but is controlled by software and, therefore, a hybrid solution is provided. Again, the user must evaluate the solutions against requirements to determine the best solution.

3.2.1. Public vs. Secret Key Cryptography

The primary advantage of public-key cryptography is increased security and convenience: private keys never need to transmitted or revealed to anyone. In a secret-key system, the secret keys must be transmitted (either manually or through a communication channel). There may be a chance that an unauthorized individual can access the secret keys during their transmission.

The primary advantage of secret key cryptography is speed. There are popular secret-key encryption methods that are significantly faster than any currently available public-key encryption method. Alternatively, public-key cryptography can be used with secret-key cryptography to get the best of both worlds: the security advantages of public-key systems and the speed advantages of secret-key systems. The public-key system can be used to encrypt a secret key that is used to encrypt the bulk of a file or message.

In some situations, public-key cryptography is not necessary and secret-key cryptography alone is sufficient. This includes environments where secure secret-key agreement can take place; environments where a single authority knows and manages all the keys; and a single-user environment. In general, public-key cryptography is best suited for an open multi-user environment.

3.3. Key Management

The proper management of cryptographic keys is essential to the effective use of cryptography for security. Ultimately, the security of information protected by cryptography directly depends on the protection afforded the keys. All keys need to be protected against modification, and secret and private keys need to be protected against unauthorized disclosure. Listed below are recommendations for effective key management.

- *Make sure that users are aware of their liabilities and responsibilities, and that they understand the importance of keeping their keys secure.*

The security of cryptographic keys in an electronic or digital signature system is the foundation of a secure system; therefore, users must maintain control of their keys! Users must be provided with a list of responsibilities and liabilities, and each user should sign a statement acknowledging these concerns before

receiving a key (if it is a long-term, user-controlled key). If different user roles (e.g., security officer, regular user) are implemented in a system, users should be aware of their unique responsibilities, especially regarding the significance of a key compromise or loss.

- *Prepare for the possibility of compromise*

It is imperative to have a plan for handling the compromise or suspected compromise of central/root keys or key components at a central site; this should be established before the system goes "live." The contingency plan should address what actions should be taken with system software and hardware, central/root keys, user keys, previously generated signatures, encrypted data, etc.

If someone's private key is lost or compromised, others must be made aware of this, so that they will no longer encrypt messages using the invalid public key nor accept messages signed with the invalid private key. Users must be able to store their private keys securely, so that no intruder can find them, yet the keys must be readily accessible for legitimate use. Keys need to be valid only until a specified expiration date.

- *Sign and verify the code that implements the cryptographic functions.*

Software at the central key management site should be electronically signed and periodically verified to check the integrity of the code. This provides a means of detecting the unauthorized modification of system software. Within a cryptomodule, this feature of generating and verifying a cryptographic checksum is required by FIPS PUB 140-1.

- *A system implemented for a Federal government agency should have its centrally stored keys and system software controlled by Federal employees.*

Proper control of central/root keys and key management software and hardware is critical to the security of the system. In the situation where a Federal agency operates a system that was developed by a contractor, Federal employees should be in control of this material. This also applies to configuring the key management hardware and software. Once the system goes live, unlimited access to central data, code, and cryptomodules should not be given to non-Federal employees, including those who were contracted to develop and/or maintain the system.

- *Secure Key Management*

Key management provides the foundation for the secure generation, storage, distribution, and translation of keys. One of the fundamental principles for

protecting keys is the practice of split knowledge[10] and dual control[11]. Split knowledge and dual control may be used to protect the centrally stored user secret keys and root private keys, secure the distribution of user tokens, and initialize all cryptomodules in the system to *authorize* their use in performing cryptographic functions within a system. Another role of key management is key maintenance, specifically, the update/replacement of keys at the completion of a cryptoperiod. The cryptoperiod is determined based on the sensitivity of the information and the risk of key compromise.

Central sites play an important role in key management. In public-key systems, central sites typically include a CA, which is an entity that issues and revokes public key certificates and may even generate key pairs. The CA private key should be protected with split knowledge and dual control. Whether in a secret- or public-key system, the security of the central site is critical to the overall cryptographic security of the system.

3.3.1. Key Generation

The generation of keys is the most sensitive of all cryptographic functions. Any inadequacies in the implementation of the key generation function or in the physical security safeguards of that function will seriously undermine the integrity of other cryptographic mechanisms. The physical security measures are necessary to prevent unauthorized disclosure, insertion, and deletion of the system or keys produced by the system. Specifically, all automated resources which generate keys and initialization vectors (IVs) should be physically protected to prevent the:

- disclosure, modification, and replacement of the keys,
- modification or replacement of the IVs,
- modification or replacement of the generation algorithm, or device.

Depending on the desired management structure, there are some applications where the generation of keys is desirable and other applications where the distribution of keys from another source, such as a central authority, may be more desirable.

- *Maintaining control of central or root keys from the time of generation is critical.*

[10] A condition under which two or more parties separately possess key components, which, individually, convey no knowledge of the resultant cryptographic key. The resultant key exists only within secure equipment.
[11] A process of utilizing two or more separate entities (usually persons) operating in concert, to protect sensitive functions or information.

Central or root keys are most likely to be used in sensitive applications such as encrypting user keys, signing a central key database for integrity, binding a key pair to a user, or generating user keys. If these keys are compromised, a complete system compromise (involving the compromise of user keys, encrypted data, and/or signed data) becomes a very real threat. It is essential to maintain the security of these central keys from the very beginning - the generation process. No one but the proper owner(s) of a key or key component should ever be able to use that key or key component. If split knowledge and dual control are a requirement for central or root keys, then a failure to maintain split knowledge and dual control of those keys at any time in their lifecycle could present both a security problem and a potential system compromise.

- *If a key is stored on a token, and a PIN is used to access the token, then only that token's owner should ever have possession of both the token and its corresponding PIN.*

This applies to root security officers who may generate a token and its Personal Identification Number (PIN), as well as any intermediaries. To prevent a courier from having sole control of both items, security officers should distribute the token and PIN in separate mailings (in separate packages mailed on different days). Also, different roles should be used to generate and mail PINs. Receipt of each item should always be confirmed to the original sender. A failure to maintain control of a token and its corresponding PIN could lead to a key compromise and the misuse of cryptographic functions within the system.

3.3.2. Key Use

- *Cryptographic keys may need special physical protection.*

If keys or key components are stored on a token (e.g., floppy disk, personal computer (PC) Card, smartcard, etc.), this token may have to be stored in a special manner to prevent unauthorized individuals from accessing the key or key component. For example, if key components for starting a CA or Key Management Facility are stored on tokens which are secured in a safe, multiple people might have access to this token. Therefore, additional protection is needed for each token, possibly by using a tamper-evident envelope, to enable the token's owner to determine if another person used a token.

- *Authentication timeout features are important for protecting keys from compromise or misuse.*

An authentication timeout feature for a cryptographic module or token is important to minimize the possibility of an unauthorized individual accessing an "active" cryptomodule and using its cryptographic keys. This could happen if a cryptomodule is left unattended by a user who has authenticated to it and loaded his/her cryptographic keys. One alternative is to force a user to periodically

reauthenticate oneself to a cryptomodule, rather than allow him/her to stay logged in for an indefinite amount of time. For sensitive applications, it may be necessary to restrict the hours during which this can take place.

- *Sign all centrally stored data and encrypt sensitive data, such as secret keys that are used to provide confidentiality.*

All centrally stored data that is related to user keys should be signed for integrity, and possibly encrypted for confidentiality (all user secret keys and CA private keys should be encrypted). Individual key records in a database - as well as the entire database - should be signed. To enable tamper detection, each individual key record should be signed, so that its integrity can be checked before allowing that key to be used in a cryptographic function. When signing the entire database, at least the important fields that do not change regularly should be signed (this allows for faster verification).

- *Provide for key recovery capabilities.*

IT systems must protect the confidentiality of information. There must be safeguards to ensure that sensitive records are neither irretrievably lost by the rightful owners nor accessed by unauthorized individuals. Key recovery capabilities provide these controls. All key components should be available to an organization regardless of whether the associated user is currently working in the organization. Employees leave organizations voluntarily and some are removed and in either situation, the organization may need to access the key components to recover encrypted data. Key recovery capabilities allow organizations to restore key components.

It is very important to have backup copies of central/root keys, since the compromise or loss of those components could prevent access to keys in the central database, and possibly deny system users the ability to decrypt data or perform signature verifications.

3.3.3. Key Archiving

- *Archive user keys for a sufficiently long cryptoperiod.*

A cryptoperiod is the time during which a key can be used for signature verification or decryption; it should extend well beyond the lifetime of a key (where the lifetime is the time during which a key can be used to generate a signature and/or perform encryption). Keys should be archived for a lengthy cryptoperiod (on the order of decades), so that they can be used to verify signatures and decrypt ciphertext during the cryptoperiod.

3.3.4. Key Destruction

- *Determine reasonable lifetimes for keys associated with different types of users.*

Users with different roles in the system should have keys with lifetimes that take into account the users' roles and responsibilities, the applications for which the keys are used, and the security services which are provided by the keys (user/data authentication, confidentiality, data integrity, etc.). Reissuing keys should not be done so often that it becomes burdensome; however, it should be performed often enough to minimize the loss caused by a possible key compromise.

- *Handle the deactivation/revocation of keys so that data signed prior to a compromise date (or date of loss) can be verified.*

It should be possible to designate a signing key as “lost” or “compromised,” so signatures generated prior to a specified date can be verified. Otherwise, all data previously signed with a lost/compromised key would have to be reviewed and re-signed.

3.4. Authentication[12]

One of the primary security controls to ensuring individual accountability (determining the identity of the user) is to authenticate each user. Traditional authentication techniques include passwords and PINs. Additional methods for authenticating users are provided by cryptographic methods. The following discussion compares traditional and cryptographic techniques. The discussion makes the assumption that both the claimant’s and verifier's local environments are trusted. The protections described are aimed at the communications path between a *claimant (user)* and a *verifier*.

3.4.1. Traditional (Weak) Authentication

Weak authentication only provides protection against attacks in which an impostor **cannot** view, insert or alter the information passed between the user who is trying to prove identity (claimant) and the system checking on the claimant’s identity (verifier) during an authentication exchange and subsequent sessions. In this scenario, an impostor attempts to assume a claimant's identity by initiating an access control session as a valid user and attempting to guess a legitimate user's authentication data.

[12] Information in this section was based on an unpublished paper developed by J. Dray, NIST.

Traditional password schemes provide weak authentication because an impostor may be able to view and later use the password to assume the user's identity. The strength of this authentication process is highly dependent on the difficulty of guessing password values and how well these values are protected.

3.4.2. Authentication Using Dynamic Authentication Data

This type of authentication mechanism relies on *dynamic* authentication data that changes with each authenticated session between a claimant and verifier. An impostor who can view information passed between a claimant and verifier may attempt to record this information, initiate a separate access control session with the verifier, and replay the recorded authentication data in an attempt to assume the claimant's identity. This authentication mechanism protects against such attacks, because authentication data recorded during a previous session will not be valid for any subsequent sessions.

However, this type of authentication does not provide protection against active attacks in which the impostor is able to **alter** the content or flow of information between the claimant and verifier after a legitimate session has been established. If the verifier binds the claimant's identity to the logical communications channel for the duration of the session, the verifier believes that the claimant is the source of all data received through this channel.

One-time passwords and Digital Signature Authentication (as described in FIPS PUB 196) provide this level of protection.

3.4.2.1. Entity Authentication Using Public Key Cryptography

Authentication based on public key cryptography has an advantage over many other authentication schemes because no secret information has to be shared by the entities (parties A and B) involved in the exchange. Party A (claimant) uses a private key to digitally sign a random number "challenge" issued by Party B (verifier). If Party B can successfully verify the signed response using Party A's public key, then Party A has been successfully authenticated.

FIPS PUB 196 specifies two challenge-response protocols by which entities in a computer system may authenticate their identities to one another. In the *unilateral authentication protocol*, one entity is the claimant and the other is the verifier. In the *mutual authentication protocol*, each entity acts as both a claimant and a verifier. These protocols may be used during session initiation, and at any other time that entity authentication is necessary. Depending on which protocol is implemented, either one or both entities involved may be authenticated. The authentication protocols in this standard may be used in conjunction with other public key-based systems (e.g., a public key infrastructure that uses public key certificates) to enhance the security of a computer system.

To acceptably implement this standard, an implementation must meet the following criteria:

1) Each entity in an authentication exchange must use a FIPS approved digital signature algorithm to generate and/or verify digital signatures;

2) Each entity must generate (pseudo)random numbers using a FIPS approved (pseudo)random number generator;

3) Each entity acting as a claimant must be bound to a public/private key pair; the private key should remain in the sole control of the claimant who uses that key to sign a random challenge. The key binding requires a unique authentication identifier for each claimant, so that a verifier can distinguish between multiple claimants; and

4) One or both of the authentication protocols in FIPS PUB 196 must be implemented. For each protocol, steps and token fields marked as [OPTIONAL] do not need to be implemented, except where indicated otherwise. However, all other steps and token fields must be implemented.

3.4.3. Authentication Against Active Attacks

This type of authentication provides protections against impostors who can view, alter, and insert information passed between a claimant and verifier even after the claimant/verifier authentication is complete. These are typically referred to as active attacks, since they assume that the impostor can actively influence the connection between claimant and verifier. One way to provide this type of authentication is to implement a digital signature algorithm to every bit of data that is sent from the claimant to the verifier. There are other combinations of cryptography that can provide this form of authentication, however, some type of cryptography must be provided to every bit of data that is sent, otherwise any unprotected bit will be suspect. Authentication against active attacks could include encryption and digital signatures.

CHAPTER 4

4. CRYPTOGRAPHY METHODS

The objective in this chapter is to provide a brief overview of the various cryptographic methods that are available. The information is extracted from FIPS PUBs and ANSI Standards. For more detailed information, reference the complete standard or publication.

4.1. Symmetric/Secret Key Cryptography

In symmetric key cryptography, the sender and receiver of a message use a shared secret key.

4.1.1. Symmetric/Secret Encryption

In symmetric/secret encryption, the sender uses a secret key to encrypt the message and the receiver uses the same secret key to decrypt the message.

4.1.1.1. Data Encryption Standard (DES)[13]

The Data Encryption Standard (DES), initially issued in 1977, provides an encryption algorithm for protecting Federal sensitive information from unauthorized disclosure or undetected modification during transmission or while in storage. DES was developed to protect sensitive computer data in Federal computer systems against a number of passive and active attacks in communications and computer systems. Based on secret key cryptography, the standard was initially issued for government use.

DES is a publicly known cryptographic algorithm that converts plaintext to ciphertext using a key that consists of 64 binary digits ("0"s or "1"s) of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits, which are not used by the algorithm, are used for error detection. The DES consists of 16 "rounds" of operations that mix the data and key together in a prescribed manner using the fundamental operations of permutation and substitution. The same algorithm is used with the same key to convert ciphertext back to plaintext. Authorized users of encrypted computer data must have the key that was used to encipher the data in order to decrypt it.

The unique key chosen for use in a particular application makes the results of encrypting data using the algorithm unique. Selection of a different key causes the cipher that is produced for any given set of inputs to be different. The cryptographic security of the data depends on the security provided for the key used to encipher and decipher the data.

[13] The information in this section was extracted from FIPS PUB 46-3 (DES).

Data can be recovered from cipher only by using exactly the same key used to encipher it. Unauthorized recipients of the cipher who know the algorithm but do not have the correct key cannot derive the original data algorithmically. However, anyone who does have the key and the algorithm can easily decipher the cipher and obtain the original data.

Early versions of the DES required that the encryption algorithm be implemented in electronic hardware and firmware. The DES standard allows for implementation of the cryptographic algorithm in software, firmware, hardware, or any combination thereof to enable more flexible, cost-effective implementations.

FIPS PUB 81, DES Modes of Operation, describes four different modes for using the algorithm described in this standard. These four modes are called the:

- Electronic Codebook (ECB) mode. ECB is a direct application of the DES algorithm to encrypt and decrypt data.

- Cipher Block Chaining (CBC) mode. CBC is an enhanced mode of ECB which chains together blocks of cipher text;

- Cipher Feedback (CFB) mode. CFB uses previously generated cipher text as input to the DES to generate pseudorandom outputs which are combined with the plaintext to produce cipher, thereby chaining together the resulting cipher; and

- Output Feedback (OFB) mode. OFB is identical to CFB except that the previous output of the DES is used as input in OFB while the previous cipher is used as input in CFB. OFB does not chain the cipher.

The DES standard became effective July 1977. It was reaffirmed in 1983, 1988, 1993, and 1999.

Note: It is anticipated that triple DES and the Advanced Encryption Standard (AES) will coexist as FIPS approved algorithms allowing for a gradual transition to AES.

4.1.1.2. Triple DES (3DES)[14]

A more secure method for using the DES algorithm in three operations, called Triple DES, has been developed by the private sector. The DES standard was revised in 1999 to include Triple DES:

[14] The information in this section was extracted from ANSI X9.52.

1. Triple DES (i.e., TDEA) as specified in ANSI X9.52 will be recognized as a FIPS approved algorithm.

2. Triple DES will be the FIPS approved symmetric encryption algorithm of choice.

3. Single DES (i.e., DES) will be permitted for legacy systems only. New procurements to support legacy systems should, where feasible, use Triple DES products running in the single DES configuration.

4. Government organizations with legacy DES systems are encouraged to transition to Triple DES based on a prudent strategy that matches the strength of the protective measures against the associated risk.

The Triple Data Encryption Algorithm (TDEA) modes of operation are used for both enciphering and deciphering operations. These modes are based on three-fold compound operations of encryption and decryption using the Data Encryption Algorithm (DEA). If two or three independent keys are used for three DEA operations, it may extend the effective key space of DEA. Certain modes also provide increased protection against more sophisticated attacks.

TDEA supports direct extension of the four DEA modes of operation, so that backward compatibility with single DEA may be maintained. A TDEA mode of operation is backward compatible with its single DEA counterpart if, with a proper keying option for TDEA operation,

1. An encrypted plaintext with single DEA mode of operation can be decrypted correctly by the corresponding TDEA mode of operation; and

2. An encrypted plaintext with TDEA mode of operation can be decrypted correctly by the corresponding single DEA mode of operation.

For throughput performance improvement in multiple processor systems, interleaved and pipelined versions of these modes are specified. The modes of operation are:

- TDEA Electronic Codebook Mode (TECB)
- TDEA Cipher Block Chaining Mode (TCBC)
- TDEA Cipher Block Chaining Mode - Interleaved (TCBC-I)
- TDEA Cipher Feedback Mode (TCFB)
- TDEA Cipher Feedback Mode - Pipelined (TCFB-P)
- TDEA Output Feedback Mode (TOFB)

- TDEA Output Feedback Mode - Interleaved (TOFB-I)

The TECB, TCBC, TCFB and TOFB modes are based on the ECB, CBC, CFB and OFB modes obtained by substituting DEA encryption/decryption operations with TDEA encryption/decryption operations.

For applications in which high TDEA encryption/decryption throughput is important or in which propagation delay must be minimized, the new interleaved (for TCBC and TOFB) and pipelined (for TCFB) modes are provided. In an interleaved mode, the plaintext sequence is split into three subsequences of plaintext. The encryption can be done simultaneously. In a pipelined mode, the encryption is initiated with three IVs at three clock cycles so that after initiation, the three DEA functional blocks can process the data simultaneously.

For all TDEA modes of operation, the three cryptographic keys (K_1, K_2, K_3) define a TDEA key bundle. The bundle and the individual keys must:

a. Be secret;
b. Have integrity;
c. Be used in the appropriate order as specified by the particular mode;
d. Be considered a fixed quantity in which an individual key cannot be manipulated while leaving the other two keys unchanged; and
e. Cannot be unbundled for any purpose.

4.1.1.3. SKIPJACK[15]

SKIPJACK is a symmetric encryption/decryption algorithm. SKIPJACK is a 64-bit codebook using an 80-bit cryptovariable (session key). The session key is used to encrypt plaintext information and to decrypt resulting ciphertext to obtain the data. There are 32 rounds of processing per single encrypt/decrypt operation. SKIPJACK can be used in any one of the four operating modes defined in FIPS PUB 81 for use with DES:

Output Feedback (OFB),
Cipher Feedback Modes (CFB),
Electronic Codebook (ECB), and
Cipher-Block Chaining (CBC).

The SKIPJACK encryption/decryption algorithm has been approved for government applications requiring encryption of sensitive but unclassified data

[15] The information in this section was extracted from *Skipjack and KEA Algorithm Specifications, Version 2.0*.

telecommunications. Data for purposes of this standard includes voice, facsimile and computer information communicated in a telephone system.

4.1.1.4. Advanced Encryption Standard (AES)

In 1993, the following statement was included in the DES standard:

> "At the next review (1998), the algorithm specified in this standard will be over twenty years old. NIST will consider alternatives that offer a higher level of security. One of these alternatives may be proposed as a replacement standard at the 1998 review."

NIST foresees that a multi-year transition period to the Advanced Encryption Standard (AES) will be necessary to move toward any new encryption standard and that DES will continue to be of sufficient strength for many applications. (AES is discussed further in section 9.1.)

4.1.2. Message Authentication Code[16]

A data authentication algorithm (DAA) may be used to detect unauthorized intentional and accidental data modifications. DES is the basis for the DAA. By applying the DES algorithm, a Message Authentication Code (MAC) is calculated on and appended to information. The MAC provides for integrity using a cryptographic checksum value. To verify that the information has not been modified at some later time, the MAC is recalculated on the information. The new MAC is compared with the MAC that was previously generated and if they are equal then the information has not been altered.

The MAC as specified in ANSI X9.9 is computed in the same manner as the data authentication code (DAC) specified in FIPS PUB 113. Similarly, the Data Identifier (DID) specified in FIPS PUB 113 is sometimes referred to as a Message Identifier (MID) in standards related to message communications.

4.1.2.1. THE DAA Authentication Process

Applying the DAA to data generates a DAC. The DAC, which is a mathematical function of both the data and a cryptographic key, may then be stored, or transmitted, with the data. When the integrity of the data is to be verified, the DAC is generated on the current data and compared with the previously generated DAC. If the two values are equal, the integrity (i.e., authenticity) of the data is verified.

The DAA detects data modifications that occur between the initial generation of the DAC and the validation of the received DAC. It does not detect errors that occur before the DAC is originally generated.

[16] The information in this section was extracted from FIPS PUB 113.

The integrity provided by the DAA is based on the fact that it is infeasible to generate a DAC without knowing the cryptographic key. An adversary without knowledge of the key will not be able to modify data and then generate an authentic DAC on the modified data. It is therefore crucial that keys be protected so that their secrecy is preserved.

4.2. Hash Functions

A hash function compresses the bits of a message to a fixed-size hash value in a way that distributes the possible messages evenly among the possible hash values. A cryptographic hash function does this in a way that makes it extremely difficult to come up with a message that would hash to a previously computed hash value.

4.2.1. SHA and SHA-1[17]

The Secure Hash Algorithm (SHA-1) can be used to generate a condensed representation of a message called a message digest. When a message of any length $< 2^{64}$ bits is input, the SHA-1 produces a 160-bit message digest. The message digest can then be input to the Digital Signature Algorithm (DSA) which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message. The same hash algorithm must be used by the verifier of a digital signature as was used by the creator of the digital signature.

The SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify. SHA-1 is a technical revision of SHA[18]. The SHA-1 is based on principles similar to those used by Professor Ronald L. Rivest of MIT when designing the MD4 message digest algorithm ("The MD4 Message Digest Algorithm," *Advances in Cryptology - CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 303-311), and is closely modeled after that algorithm.

SHA-1 is required for use with the DSA as specified in the Digital Signature Standard (DSS) and whenever a secure hash algorithm is required for Federal applications.

---

[17] The information is this section was extracted from FIPS PUB 180-1.

[18] A circular left shift operation has been added to the specifications in section 7, line b, page 9 of FIPS PUB 180 and its equivalent in section 8, line c, page 10 of FIPS PUB180. This revision improves the security provided by this standard.

The SHA-1 may be used with the DSA in electronic mail, electronic funds transfer, software distribution, data storage, and other applications that require data integrity assurance and data origin authentication. The SHA-1 may also be used whenever it is necessary to generate a condensed version of a message.

4.3. Asymmetric Key Cryptography

The main problem with symmetric key cryptography is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations they must trust a courier, or a phone system, or some other transmission medium to prevent the disclosure of the secret key being communicated.

The concept of public-key cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman [DH76] in order to solve the key management problem. In their approach, each person gets a pair of keys, one called the public key and the other called the private key. Each person's public key is published while the private key is kept secret. All communications involve only public keys, and no private key is ever transmitted or shared. The only requirement is that public keys are associated with their users in a trusted (authenticated) manner. Anyone can send a confidential message by using only the public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient.

4.3.1. Digital Signatures

A digital signature is an electronic analogue of a written signature in that the digital signature can be used in proving to the recipient or a third party that the message was, in fact, signed by the originator. Digital signatures may also be generated for stored data and programs so that the integrity of the data and programs may be verified at any later time.

Digital signatures authenticate the integrity of the signed data and the identity of the signatory. Digital signatures may also be used in proving to a third party that data was actually signed by the generator of the signature. Digital signatures are intended for use in electronic mail, electronic funds transfer, electronic data interchange, software distribution, data storage, and other applications that require data integrity assurance and data origin authentication.

A digital signature is represented in a computer as a string of binary digits and is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. An algorithm provides the capability to generate and verify signatures. Signature generation makes use of a private key to generate a digital signature. Signature verification makes use of a public key which corresponds to, but is not the same as, the private key. Each user possesses a private and public key pair. Anyone can verify the signature of

a user by employing that user's public key. Signature generation can be performed only by the possessor of the user's private key. The security of a digital signature system is dependent on maintaining the secrecy of users' private keys. Users must, therefore, guard against the unauthorized acquisition of their private keys.

A hash function is used in the signature generation process to obtain a condensed version of data, called a message digest (see Figure 3). The message digest is then input to the digital signature (ds) algorithm to generate the digital signature. The digital signature is sent to the intended verifier along with the signed data (often called the message). The verifier of the message and signature verifies the signature by using the sender's public key. The same hash function must also be used in the verification process. Similar procedures may be used to generate and verify signatures for stored as well as transmitted data.

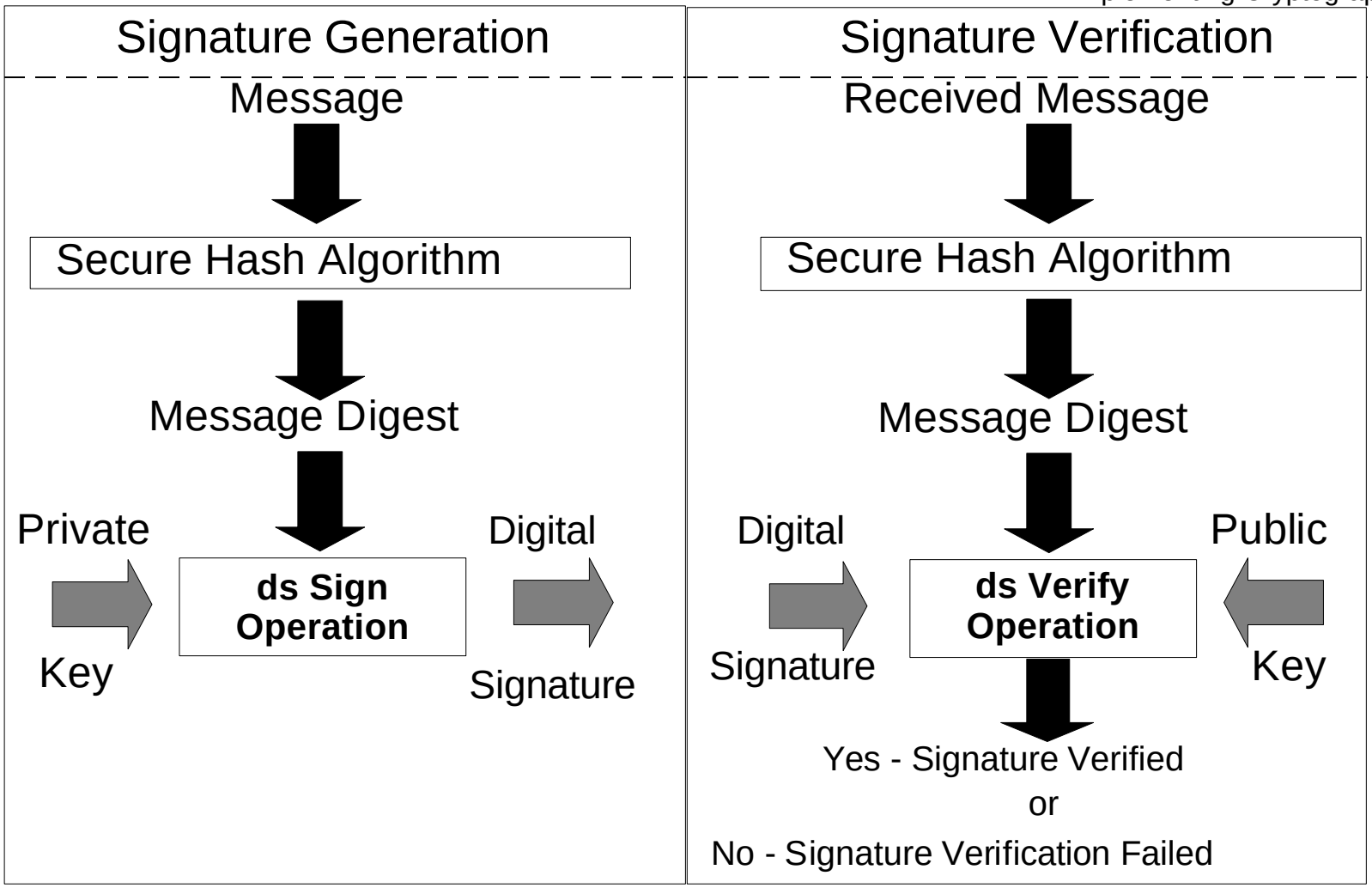


Figure 4. Digital Signatures

A digital signature can also be used to verify that information has not been altered after it was signed; this provides message integrity.

The non-repudiation property of a digital signature relies on the mathematical assumption that it is computationally infeasible to derive the private key from the public key and/or a set of messages and signatures prepared using the private key. The non-repudiation property of a digital signature also relies on the practical assumption that the private key is, or can be, associated with a single entity (the signer), that only the signer has knowledge of or use of the private key, and that the private key can and will be kept secret.

Digital signatures offer protection not available by alternative signature techniques. One such alternative is a digitized signature. A digitized signature is generated by converting a visual form of a handwritten signature to an electronic image. Although a digitized signature resembles its handwritten counterpart, it does not provide the same protection as a digital signature. Digitized signatures can be forged. They can also be duplicated and appended to other electronic data. Digitized signatures cannot be used to determine if information has been altered after it is signed.

4.3.1.1. Digital Signature Standard (DSS)[19]

FIPS PUB 186-2, *Digital Signature Standard (DSS)*, is based on public key cryptography which makes use of two keys: a public key and a private key. The DSS specifies a digital signature for use in computing and verifying digital signatures. DSS includes three digital signature algorithms: DSA, RSA and Elliptic Curve Digital Signature Algorithm (ECDSA). The DSS is used in conjunction with FIPS PUB 180-1, *Secure Hash Algorithm.*

FIPS PUB 186-2 allows for the use of DSA, ANSI X9.31 (*Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)),* and ANSI X9.62 (*Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm*). The ANSI X9.31 standard describes the Rivest-Shamir-Adleman (RSA) digital signature technique.

FIPS PUB 186-2 reflects the availability of conformity testing for DSA implementations. (ANSI's conformity testing programs for ANSI X9.31 and ANSI X9.62 implementations are not yet in place.)

Separate keys should be used for signature and confidentiality purposes when using the ANSI X9.31 standard. This is because the RSA algorithm can be used for both data encryption and digital signature purposes. To minimize any potential for spoofing digital signatures, keys used for signature purposes should

[19] The information in this section was extracted from FIPS PUB 186-2.

not be recoverable. Using separate keys will allow agencies to recover confidentiality keys but not signature keys.

Digital Signature Algorithm (DSA)

DSA is used by a *signatory* to generate a digital signature on data and by a *verifier* to verify the authenticity of the signature. Each signatory has a public and private key. The private key is used in the signature generation process and the public key is used in the signature verification process. The private key is randomly generated and is kept secret. Its owner should control its use and it should be protected against modification as well as disclosure. Using this key and a mathematical process defined in the standard, the public key is generated. The public key can be known by anyone; however, no one should be able to modify it.

DSA must be used in designing and implementing public-key based signature systems that Federal departments and agencies operate or which are operated for them.

Digital Signature Process

The DSA is used with SHA-1 to generate and verify digital signatures. To generate a signature on a message, the owner of the private key first applies the SHA-1 to the message. This action results in a message digest. The owner of the private key then applies the private key to the message digest using the mathematical techniques specified in the DSA to produce a digital signature. Any party with access to the public key, message, and signature can verify the signature using the DSA. If the signature verifies correctly, the receiver (or any other party) has confidence that the message was signed by the owner of the public key and the message has not been altered after it was signed.

In addition, the verifier can provide the message, digital signature, and signer's public key as evidence to a third party that the message was, in fact, signed by the claimed signer. Given the evidence, the third party can also verify the signature. This capability, an inherent benefit of public key cryptography, is called non-repudiation. The DSS does not provide confidentiality of information. If confidentiality is required, the signer could first apply the DES to the message and then sign it using the DSA.

A means of associating public and private key pairs to the corresponding users is required. That is, there must be a binding of a user's identity and the user's public key. This binding may be certified by a mutually trusted party. For example, a certifying authority could sign credentials containing a user's public key and identity to form a certificate.

Applications of Digital Signatures. Because the DSA authenticates both the identity of the signer and the integrity of the signed information, it can be used in a variety of applications. For example, the DSA could be utilized in an electronic mail system. After a party generated a message, that party could sign it using the party's private key. The signed message could then be sent to a second party. After verifying the received message, the second party would have confidence that the message was signed by the first party. The second party would also know that the message was not altered after the first party signed it.

The DSA could also be useful in the distribution of software. A digital signature could be applied to software after it has been validated and approved for distribution. Before installing the software on a computer, the signature could be verified to be sure no unauthorized changes (such as the addition of a virus) have been made. The digital signature could be verified periodically to ensure the integrity of the software.

Random Number Generation

To use the DSA, a party must be able to generate random numbers to produce the public/private key pair and to compute the signature. Random numbers can be generated either by a true noise hardware randomizer or by using a pseudorandom number generator. Approved random number generators are found in Appendix 3 of FIPS PUB 186 and Appendix C of ANSI X9.17, *Financial Institution Key Management (Wholesale).* Random numbers are used to derive a user's private key, *x*, and a user's per-message secret number, *k*. These values are used in the DSA. The randomly or pseudorandomly generated integers are selected to be between 0 and the 160-bit prime *q* (as specified in the standard).

rDSA[20]

Digital Signatures Using Reversible Public Key Cryptography For The Financial Services Industry (rDSA), is a technique for generating and validating digital signatures. When implemented with proper controls, the techniques will provide the ability to determine:

- data integrity, and
- non-repudiation of the message origin and contents.

Additionally, rDSA provides the ability to detect duplicate messages and prevent the acceptance of replayed messages when the signed message includes:

1. The identity of the intended recipient, and
2. A message identifier (MID).

[20] The information in this section was extracted from ANSI X9.31.

The MID should not repeat during the cryptoperiod of the underlying private/public key pair.

The standard, adapted from ISO/IEC 9796-2 [2] and ISO/IEC 14888-3 [16], defines a method for digital signature generation and verification for the protection of financial messages and data using reversible public key cryptography systems without message recovery. In addition, rDSA provides the criteria for the generation of public and private keys required by the algorithm and the procedural controls required for the secure use of the algorithm.

For both signature generation and verification, the data that is referred to in this standard as a message, M, is reduced by means of a hash algorithm. Also, there must be a reliable binding of a user's identity and the user's public key. This binding may be accomplished by a mutually trusted party in the formulation of a public key certificate using a CA.

rDSA includes:

- *Key generation*. The outputs from key generation are a public verification key and a private signature key.
- *Signature process*. The signature generation process consists of the following steps: message hashing, hash encapsulation, signature production, and signature validation (optional).
- *Verification process*. The signature verification process consists of the following steps: signature opening, encapsulated hash verification, hash recovery, and message hashing and comparison.

For rDSA, the integrity of signed data is dependent upon:

1. The prevention of unauthorized disclosure, use, modification, substitution, insertion and deletion of *d* (private signature exponent)*, p and q* (private prime factors)*,* or seeds.
2. The prevention of unauthorized modification, substitution, insertion and deletion of *e* (public exponent) and *n* (public modulus)*.*

The primes *p* and *q* (the factors of the modulus *n*) must be kept secret or destroyed. If the private signature exponent, *d*, or the seeds are disclosed, the integrity of any message signed using that *d* can no longer be assured. Also, key generation should be protected from unauthorized access to prevent disclosure of sensitive keying material. Using the same seeds will produce the same keying material that may have been compromised.

An Overview of Elliptic Curve Schemes

Many public-key cryptographic schemes are based on exponentiation operations in large finite mathematical groups. The cryptographic strength of these

schemes is derived from the believed computational intractability of computing logarithms in these groups. The algebraic system defined on the points of an elliptic curve provides an alternate means to implement cryptographic schemes based on the discrete logarithm problem. The primary advantage of elliptic curve schemes is their apparent high cryptographic strength relative to the key size.

Elliptic curve systems are public-key (asymmetric) cryptographic algorithms that are typically used to:

1. Create digital signatures (in conjunction with a hash algorithm), and
2. Establish secret keys securely for use in symmetric-key cryptosystems.

Elliptic Curve Digital Signature Algorithm (ECDSA)[21]

The Elliptic Curve Digital Signature Algorithm (*ECDSA*) defines a technique for generating and validating digital signatures. ECDSA is the elliptic curve analogue of DSA. The ECDSA must be used in conjunction with the hash function SHA-1.

When implemented with proper controls, ECDSA provides data integrity, data origin authentication, and non-repudiation of the message origin and the message contents. Additionally, when used in conjunction with a MID, ECDSA provides the capability of detecting duplicate transactions.

The ECDSA is used by a *signatory* to generate a digital signature on data and by a *verifier* to verify the authenticity of the signature. Each signatory has a public and private key. The private key is used in the signature generation process, and the public key is used in the signature verification process. For both signature generation and verification, the message, M, is compressed by means of the SHA-1 prior to the signature generation and verification process.

Control of Keying Material: The signatory must provide and maintain the proper control of all keying material. In the ECDSA asymmetric cryptographic system, the integrity of signed data is dependent upon:

1. the prevention of unauthorized disclosure, use, modification, substitution, insertion, and deletion of the private key, *d*, the per-message value, *k,* and (optional) seeds input to their generation, and

2. the prevention of unauthorized modification, substitution, insertion, and deletion of elliptic curve parameters for the ECDSA computation procedures (see Section 5.1 of ANSI X9.62).

[21] The information in this section was extracted from ANSI X9.62.

If *d* is disclosed, the integrity of any message signed using that *d* can no longer be assured. Similarly, the values for the elliptic curve parameters must be protected.

(Note: Key generation should be performed on physically isolated equipment so that in the event of a hardware or software failure, no partial information is retained. For example, if a system crash causes a core dump, some of the keying material data may be captured.)

ECDSA includes:

- Key generation,
- Key validation,
- Signature generation, and
- Signature verification.

4.3.2. Key Transport/Agreement

Key management is extremely important because the security of any cryptographic system is dependent on the security provided to the cryptographic keys. For a cryptographic system to work effectively, keys must be generated, distributed, used, and destroyed securely. Key management is an issue in both secret key systems and public key systems.

Symmetric schemes such as the DEA provide services such as data integrity and data confidentiality. However, the major drawback with the implementation of such schemes is that any two communicating entities must establish in advance a shared secret key. As the size of a system or the number of entities using a system increases this can lead to a key management problem.

An attractive solution to this key management problem is for a system to employ asymmetric techniques that allow any pair of entities to establish a shared secret key suitable for use by a symmetric scheme despite the fact that the two entities may never have previously engaged in a secure communications together. Such asymmetric techniques are known as asymmetric key establishment schemes.

4.3.2.1. RSA[22]

Employing public key cryptography for the management of symmetric keys requires sound public key pair generation, key transport and key agreement.

[22] The information in this section was extracted from draft ANSI X9.44. Because this standard is in draft form, the information presented in this section is subject to revision.

ANSI X9.44, *The Transport of Symmetric Algorithm Keys Using Reversible Public Key Cryptography*, is a draft standard for secret key transport based on the RSA algorithm. The RSA and Rabin-Williams asymmetric algorithms are well understood and widely implemented public key techniques that facilitate cost-effective key management across modern networks, such as the Internet. RSA gets its security from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large (100 to 200 digits or even larger) prime numbers.

The standard defines mechanisms for managing symmetric cryptographic keys using reversible public key cryptography. The standard also addresses the security requirements and additional considerations when implementing key management in combination with digital signatures in a PKI. The techniques specified in the standard are designed to facilitate the secure establishment or secure transportation of symmetric keys.

Key Generation: the outputs from key generation are:

1. a public verification key.

2. a private signature key.

   Although each of the private signature key outputs are optional, enough information must be retained to regenerate *d*, the private signature exponent, for signature generation.

3. (Optional) audit information.

Key Transport is a mechanism whereby one party (the sender) generates a random symmetric key and transports the symmetric key encrypted using the public key of another party (the receiver). Key Transport using reversible public key cryptography consists of the following steps:

1. Symmetric key generation,
2. Symmetric key encryption, and
3. Symmetric key recovery.

Key Agreement is a mechanism whereby two parties actively share in the establishment of a random symmetric key without either party actually exchanging the symmetric key. Key Agreement using reversible public key cryptography consists of the following steps:

1. Symmetric key component generation,
2. Symmetric key component encryption,
3. Symmetric key component recovery, and
4. Symmetric key derivation.

4.3.2.2. Elliptic Curve Key Agreement and Transport Protocols[23]

ANSI X9.63 defines key establishment schemes that employ asymmetric techniques. Both key agreement and key transport schemes are specified. The arithmetic operations involved in the operation of the schemes take place in the algebraic structure of an elliptic curve over a finite field. The schemes may be used to compute shared keying data which may then be used by symmetric schemes to provide cryptographic services like data confidentiality and data integrity. The fundamental goal of any key establishment scheme is to distribute keying data. Ideally, the keying data should have precisely the same attributes as keying data established face-to-face. It should be randomly distributed, and no unauthorized entity should know anything about the keying data.

The asymmetric key establishment schemes in ANSI X9.63 are used by an entity *U* who wishes to establish a symmetric key with another entity *V*. Each entity has an elliptic curve (EC) key pair. If *U* and *V* simultaneously execute a scheme with corresponding keying material as input, then at the end of the execution of the scheme, *U* and *V* will share keying data that can be used by symmetric algorithms.

ANSI X9.63 specifies a variety of asymmetric key establishment schemes. Each of the mechanisms, when implemented securely and embedded within a cryptographic system in an appropriate manner, is capable of providing two entities with a shared secret key suitable for use in symmetric algorithms like the DEA. A variety of schemes is specified because of the wide variety of services that it may or may not be desirable for a key establishment scheme to provide depending on the environment in which the scheme is going to be used. The schemes in this Standard employ other cryptographic transformations in their operation. The transformations used are: DEA, DEA-based MAC, SHA-1, and the ECDSA.

Implementing the Schemes Securely

Two common prerequisites for the implementation of schemes in ANSI X9.63 are that all entities involved in the use of the schemes are provided with an authentic copy of the elliptic curve parameters being used and that every entity is provided with a genuine copy of every other entity's static public key. The latter binding between an entity and its static public key may be accomplished by using a Certification Authority.

[23] The information in this section was extracted from draft ANSI X9.63. Because this standard is in draft form, the information presented in this section is subject to revision.

However, satisfying the stated prerequisites is not enough to ensure the security of an implementation. The secure implementation of the schemes in ANSI X9.63 is also dependent upon:

1. The prevention of unauthorized disclosure, use, modification, substitution, insertion, and deletion of an entity's static private key d_s;
2. The prevention of unauthorized modification, substitution, insertion, and deletion of the elliptic curve parameters being used;
3. The secure implementation of the transformations involved in an execution of a scheme so that the integrity and confidentiality of the computations involved is maintained.

Note that this includes the secure destruction of any ephemeral values involved in the operation of a scheme. Any implementation must also provide explicit key authentication of any session key established using one of the key establishment schemes. Finally, secure implementation of the schemes does not guarantee the security of the operation of the implementation. It is the responsibility of the operator to put an overall process in place with the necessary controls to ensure the secure operation.

Key Agreement Schemes

The key agreement scheme is used by an entity *U* who wishes to agree keying data with an entity *V*. In some cases, the protocols specified are "symmetric," and so it suffices to describe just one transformation. In other cases, the protocols are "asymmetric," and so it is necessary to describe two transformations, one of which is undertaken by *U* if *U* is the initiator, and one of which is undertaken by *U* if *U* is the responder. In the specification of each transformation, equivalent computations that result in identical output are allowed.

Key Transport Schemes

The key transport scheme is used by an entity *U* who wishes to establish keying data with an entity *V*. Both protocols specified are 'asymmetric', so it is necessary to describe two transformations, one of which is undertaken by *U* if *U* is the initiator, and one of which is undertaken by *U* if *U* is the responder. In the specification of each transformation, equivalent computations that result in identical output are allowed.

4.3.2.3. Agreement of Symmetric Keys Using Discrete Logarithm Cryptography[24]

The Diffie-Hellman and MQV key agreement protocols (also called exponential key agreement) allows two users to exchange a secret key over an insecure medium without any prior secrets. This secret key can then be used to encrypt further communications between the parties. The protocols depend on the discrete logarithm problem for their security. The basic versions provide protection in the form of secrecy of the resulting key from passive adversaries (eavesdroppers), but not from active adversaries capable of intercepting, modifying, or injecting messages.

The basic algorithms used to calculate a shared secret number are the Diffie-Hellman algorithm and the MQV algorithm. A cryptographic key will be derived from the shared secret number by using a key derivation function. The key derivation function must be a one-way hash function. The default hash function is SHA-1.

Key Agreement Using the Diffie-Hellman Algorithm

dhStatic: Each party has only static data available. Individual static private/public key pairs are generated using the same set of static key domain parameters.

dhEphem: Each party has only ephemeral data available. Individual ephemeral private/public key pairs are generated using the same set of ephemeral key domain parameters.

dhOneFlow: One party has only static data and the other party has only ephemeral data. Two private/public key pairs are generated using the same domain parameters.

dhHybrid1: Each party has two pairs of private/public keys: one key pair is static and another is ephemeral. Two private/public pairs are generated using the same domain parameters.

dhHybrid2: Each party has two pairs of private/public keys: one pair is static and the other is ephemeral. The static key pair is generated using a set of static key domain parameters. The ephemeral key pair is generated using a set of ephemeral key domain parameters.

[24] The information in this section was extracted from draft ANSI X9.42. Because this standard is in draft form, the information presented in this section is subject to revision.

dhHybridOneFlow: One party has both a static and an ephemeral private/public key pair and the other party has one static private/public key pair. All of the key pairs are generated using the same domain parameters.

Key Agreement Using the MQV Algorithm

MQV2: Each party has two pairs of private/public keys: one pair is static and the other is ephemeral. Individual static and ephemeral private/public key pairs are generated using the same domain parameters.

MQV1: Two parties contribute different amounts of information and use different algorithms to obtain the common shared secret value. Party A has two pairs of private/public keys. One key pair is static and the other key pair is ephemeral. Party B has one private/public key pair that is static.

4.3.2.4. Key Exchange Algorithm (KEA)

KEA is a key exchange algorithm. All calculations for KEA require a 1024-bit prime modulus. This modulus and related values are to be generated as per the DSS specification. The KEA is based upon a Diffie-Hellman protocol utilizing SKIPJACK to reduce final values to an 80-bit key. The KEA provides security commensurate with that provided by SKIPJACK, on the order of 2^{80} operations. KEA requires that each user be able to validate the public values received from others, but does not specify how that is to be done.

4.4. Key Management[25]

FIPS PUB 171, along with ANSI X9.17, provides a key management system for:

- a Point-to-Point environment in which each party to a key exchange shares a key encrypting key which is used to distribute other keys between the parties,

- a Key Distribution Center environment in which each party shares a key encrypting key with a center who generates keys for distribution and use between pairs of parties, and

- a Key Translation Center environment in which each party shares a key encrypting key with a center who translates keys generated by one party which will be distributed to another party, the ultimate recipient.

[25] The information in this section was extracted from FIPS PUB 171 and ANSI X9.17.

ANSI X9.17-1985, *Financial Institution Key Management (Wholesale),* is a voluntary industry standard that defines procedures for the manual and automated management of the data (e.g., keys and initialization vectors) necessary to establish and maintain cryptographic keying relationships. It defines the protocols to be used by financial institutions, such as banks, to transfer encryption keys. This protocol is aimed at the distribution of secret keys using symmetric (secret-key) techniques. This data is known as keying material. ANSI X9.17 specifies the minimum requirements for:

- Control of the keying material during its lifetime to prevent unauthorized disclosure, modification or substitution;

- Distribution of the keying material in order to permit interoperability between cryptographic equipment or facilities;

- Ensuring the integrity of keying material during all phases of its life, including its generation, distribution, storage, entry, use and destruction; and

- Recovery in the event of a failure of the key management process or when the integrity of the keying material is questioned.

ANSI X9.17 utilizes DES to provide key management solutions for a variety of operational environments. As such, ANSI X9.17 contains a number of options. Systems which are built to conform to all options of ANSI X9.17 are likely to be complex and expensive. FIPS PUB 171 adopts ANSI X9.17-1985 and specifies a particular selection of options for the automated distribution of keying material by the Federal government using the protocols of ANSI X9.17. In FIPS PUB 171 each option is numbered and listed, its use in ANSI X9.17 is described, the selection for Federal government use is specified along with any other additional requirements, and a brief justification for the selection is provided. The options selected were chosen with regard to the degree of cryptographic protection that can be provided for the data with which the keys will be used, as well as a decision to reduce the complexity and cost of ANSI X9.17 implementations by limiting the number of options which are implemented and tested.

CHAPTER 5

5. PUBLIC KEY INFRASTRUCTURE (PKI)[26]

Public key cryptography can play an important role in providing needed security services including confidentiality, authentication, digital signatures, and integrity. This chapter includes an overview of a Public Key Infrastructure (PKI), a discussion of alternative PKI architectures, interoperability and policy issues, a description of the Minimum Interoperability Specification for PKI Components (MISPC), and the Federal government PKI.

5.1. Public Key Infrastructure (PKI) Overview

A PKI provides the means to bind public keys to their owners and helps in the distribution of reliable public keys in large heterogeneous networks. Public keys are bound to their owners by public key certificates. These certificates contain information such as the owner's name and the associated public key and are issued by a reliable CA.

The basic components of a PKI are:

Public Key Certificate - An electronic record that binds a public key to the identity of the owner of a public-private key pair and is signed by a trusted entity. Public key certificates are the mechanism for describing trust relationships in a PKI. Certificates may be issued to CAs or other end entities. Certificates issued to CAs indicate the certificate holder is trusted to issue additional certificates. Certificates issued to other end entities are appropriate for provisioning other security services, but are not trusted for issuing additional certificates. Certificates include an expiration date. However, if the CA ceases to trust the certificate holder before certificate expiration, the CA must revoke the certificate.

Certificate Revocation List (CRL) - A list of certificates that have been revoked. The list is usually signed by the same entity that issued the certificates. Certificates can be revoked for several reasons. For example, a certificate can be revoked if the owner's private key has been lost; the owner leaves the company/agency; or the owner's name changes. CRLs also provide an important mechanism for documenting the historical revocation status of certificates. That is, a dated signature may be presumed to be valid if the signature date was within the validity period of the certificate, and the current CRL of the issuing CA at that date did not show the certificate to be revoked.

[26] The information in this section was extracted from unpublished papers developed by W. E. Burr, D. F. Dodson, N. A. Nazario, and W. T. Polk of NIST.

CA - A trusted entity that issues and revokes public key certificates and certificate revocation lists.

Registration Authority (RA) - An entity that is trusted by the CA to register or vouch for the identity of users to a CA.

Certificate Repository - An electronic site that holds certificates and CRLs. CAs post certificates and CRLs to repositories.

Certificate User - An entity that uses certificates to know, with certainty, the public key of another entity.

Certificate Holder - An entity that is issued a certificate and can sign digital documents.

Clients - Entities that validate digital signatures and their certification paths from a known public key of a trusted CA.

5.2. PKI Architectures

A PKI is often composed of many CAs linked by trust paths. The CAs may be linked in several ways. They may be arranged hierarchically under a "root CA" that issues certificates to subordinate CAs. The CAs can also be arranged independently in a mesh[27]. Recipients of a signed message with no relationship with the CA that issued the certificate for the sender of the message can still validate the sender's certificate by finding a path between their CA and the one that issued the sender's certificate. Figures 4 and 5 illustrate the two basic PKI architectures.

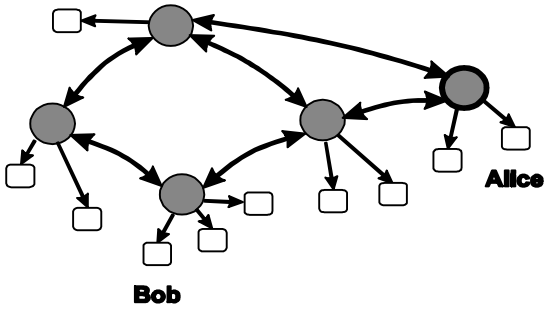


Figure 4. Mesh Architecture

[27] A mesh PKI model is sometimes referred to as a *network* PKI model.

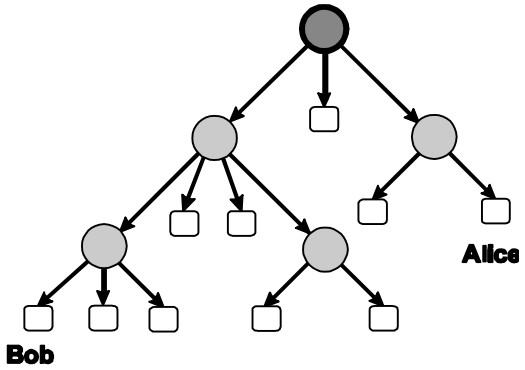


Figure 5. Hierarchical Architecture

In hierarchical models, trust is delegated by a CA when it certifies a subordinate CA. Trust delegation starts at a root CA that is trusted by every node in the infrastructure. In mesh models, trust is established between any two CAs in peer relationships (cross-certification), thus allowing the possibility of multiple trust paths between any two CAs.

5.3. Security Policies of Other CAs and the Network

It is important to consider the integrity and security of the PKI components. The confidence that can be placed in the binding between a public key and its owner depends much on the confidence that can be placed on the system that issued the certificate that binds them. The rules expressed by certificate policies are reflected in certification practice statements (CPSs) that detail the operational rules and system features of CAs and other PKI components. By examining a CA's CPS, users can determine whether to obtain certificates from it, based on their security requirements. Other CAs can also use the CPS to determine if they want to cross-certify with that CA. The essential issue with cross-certificates is how to allow CAs to cross-certify with other CAs to meet the particular needs of their own users, without compromising the security of users of other CAs. For example, a particular agency might have a close working relationship with a local government office, a particular contractor or law firm that has its own CA. That relationship, however, would not necessarily justify extension of trust to other government agencies or commercial organizations.

5.4. Interoperability

To be useful in a global sense, PKI components need to interoperate regardless of the source of the equipment and the software involved. PKI technology promises to deliver security services across user communities, even where business partners have not met face to face. However, the current PKI products and services fall somewhat short of this promise, and interoperability is one major factor. For example, incompatible transaction protocols and certificate

formats prevent implementation of heterogeneous PKIs. PKI components from different vendors may be unable to communicate. PKI users may find they can communicate, but cannot process each other's certificates.

Although there have been several proposed formats for public key certificates, most certificates available today are based on an international standard (ITU-T X.509 version 3). This standard defines a certificate structure that includes several optional extensions. The use of X.509v3 certificates is important because it provides interoperability between PKI components. Also, the standard's defined extensions offer flexibility to support specific business needs. If a community identifies additional information that is not covered by the standard extensions, they can define it and include it in their certificates or CRLs without violating the current format.

Impediments to interoperability remain, however. The number of standard extensions is relatively large, and vendors are struggling to prioritize their implementation. When a certificate is issued, extensions may be marked as "critical" or "non-critical." If an extension is critical, a product must recognize and process the extension or reject the certificate. If users have certificates with critical extensions that are not broadly supported in products, they will not be able to provision services with other users. In addition, many extensions are broadly defined. The content and semantics of the extensions are unclear, so two PKI products may interpret them differently.

5.5. Minimum Interoperability Specification for PKI Components (MISPC)

To enhance interoperability of Commercial-Off-The-Shelf (COTS) products, NIST recently completed the development of a *Minimum Interoperability Specification for PKI Components, version 1 (MISPC)*. The MISPC was produced in cooperation with ten industry partners through Cooperative Research and Development Agreements (CRADAs) and provides a basis for interoperable PKI components from different vendors.[28] The MISPC focuses primarily on the aspects of PKI interoperation most apparent to end users, that is, how to request and be issued a certificate, how to sign documents, how to retrieve the certificates of others, and how to validate signatures.

The goal of the MISPC is to further interoperability among heterogeneous public key certificate management systems, thus enabling large user communities to take advantage of digital signature technology. Transactions for issuing, revoking and managing public key certificates are defined in the specification. These transactions support the use of digital signatures as a replacement for

[28] The participating companies were: AT&T; Bolt Beranek Newman (BBN); Certicom; Cylink; DynCorp; Entrust Technologies (Northern Telecom); Information Resources Engineering (IRE); Motorola; Spyrus, Inc.; and VeriSign, Inc.

handwritten signatures and as a reliable authentication mechanism. This is sufficient to meet two basic goals of interoperability. First, conforming products and services can be mixed to build a heterogeneous PKI. Second, users with conforming certificates can establish the trust relationships required to use signatures and perform authentication.

The MISPC is based on the use of X.509v3 certificates and v2 CRLs. The specification includes a certificate profile that enumerates support and use of the standardized certificate extensions. It provides specifications for the five PKI components: Certificate Authorities, Registration Authorities, Repositories, Certificate Holders (which hold certificates and can sign documents) and Clients (which validate signatures). The MISPC includes protocols for issuing and revoking certificates and retrieving them from repositories and supports the use of digital signature certificates and recognizes three digital signature algorithms including: DSA, ECDSA and RSA with SHA-1 message digests.

The MISPC supports both mesh and hierarchical and trust models, as shown in Figures 5 and 6 and assumes that X.509v3 extensions will be included in certificates to explicitly manage trust relationships. The MISPC assumes that certificates and CRLs are available in a repository for retrieval without authentication. MISPC clients can perform path validation by obtaining the necessary certificates and CRLs from the appropriate repositories.

5.6. Federal PKI Architecture

Currently, there are many efforts in Federal agencies to set up independent CAs to support individual applications. In general, an application that supports the agency mission, such as purchasing, grants, or travel pays for operating the CA. For these applications, the use of public key technology must be justified in terms of its direct benefit to a specific agency application. Alternatively, the Federal government may use commercial CA service providers to issue certificates and facilitate delivering services. The various agency projects that rely upon these certificates will pay the commercial CA service provider.

Broader government-wide PKI needs and associated systems do not generally facilitate interagency operation, or the creation of a broader national PKI. The main issue for the Federal PKI is how to create certification paths between Federal agencies that will provide for reliable and broad propagation of trust. A *Bridge CA (BCA)* provides systematic certification paths between CAs in agencies, and outside the government. Federal CAs that meet certain standards and requirements will be eligible to cross-certify with the BCA, thereby gaining the certification paths needed for broad trust interoperation in the Federal and national PKI. While the certification path processing limitations of some less functional clients may confound interoperability at times, the existence of these certification paths is a necessary precondition for broad trust interoperation.

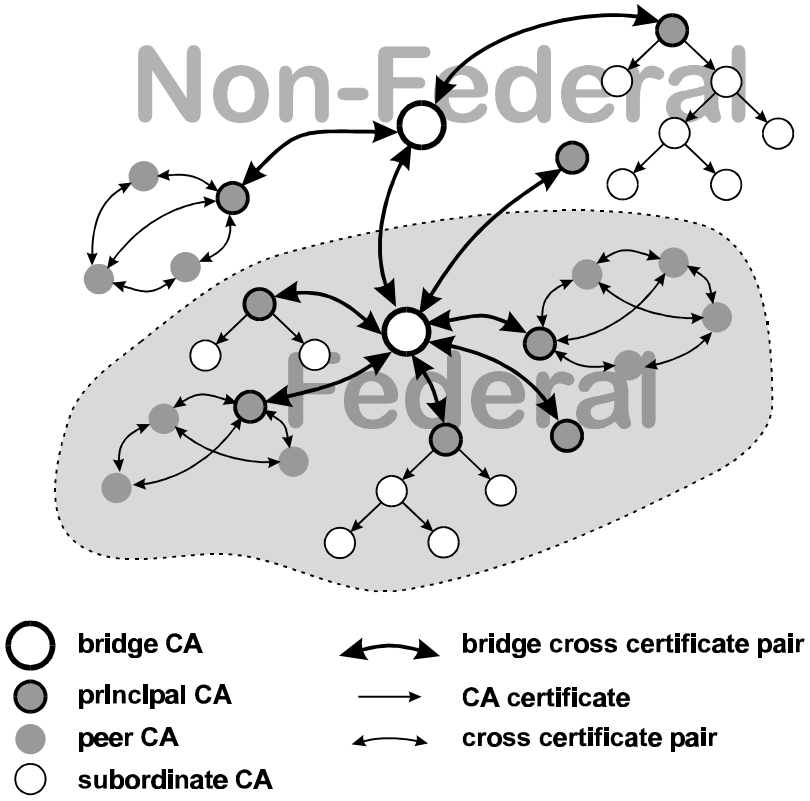


Figure 6. Proposed Federal PKI Certification Path Architecture

5.6.1. Architecture Components

The certification path elements of the proposed architecture are illustrated in Figure 6. The complete architecture is composed of the following components:

- ***Federal Policy Management Authority (FPMA)***: this management authority sets the overall policies of the Federal PKI, and approves the policies and procedures of trust domains within the Federal PKI. It operates a Federal Bridge CA, and repository.

- ***Trust Domains***: In the Federal context, a trust domain is a portion of the Federal PKI that operates under the management of a single *policy management authority*. One or more CAs exist within each trust domain. Each trust domain has a single *principal CA*, but may have many other CAs. Each trust domain has a domain repository.

- ***Domain Policy Management Authory (DPMA)***: a policy management authority approves the certification practice statements of the CAs within a trust domain and monitors their operation. The DPMAs operate or supervise

a domain repository.

- **Certification Authorities (CA):**

  - ***Bridge CA (BCA)***: the Federal Bridge CA is operated by the FPMA. Its purpose is to be a *bridge of trust* that provide trust paths between the various trust domains of the Federal PKI, as well as between the Federal PKI and non-Federal trust domains. FPMA-approved trust domains designate a principal CA that is eligible to cross-certify with the Federal BCA. The BCA is not a *root CA* because it does not typically begin certification paths.

  - ***Principal CA***: A CA within a trust domain that cross-certifies with the Federal BCA. Each trust domain has one principal CA. In a domain with hierarchical certification paths, it will be the root CA of the domain. In a mesh organized domain, the principal CA may be any CA in the domain. However it will typically be one operated by or associated with the DPMA.

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

  - ***Root CA***: In a hierarchical trust domain, the root CA is the CA that initiates all trust paths. Certificate holders and relying parties are given the self-signed root CA certificate by some authenticated means and all trust paths are initiated from that point. For hierarchical trust domains, the root CA is also the principle CA for that domain.

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

- ***Repositories*:** Repositories are online facilities that provide certificates and certificate status information. Repositories in the Federal PKI will provide information via the LDAP protocol and may also provide information in other ways. The FPMA will maintain an open LDAP repository for CA certificates and revocations.

- ***BCA Repository***: The BCA repository will be open to Internet access by anyone via LDAP, and will provide the following:
  - All certificates issued by the BCA,

  - All certificates held by the BCA,

  - All cross certificate pairs containing certificates held or issued by the BCA,

  - The current CRL for all certificates issued by the BCA,

  - Many or all CA certificates issued by CAs within the overall Federal PKI as an aid to finding certification paths,

  - Many or all cross certificate pairs between CAs in the Federal PKI, and

  - Other certificates and CRLs as determined by the FPMA.

- ***Certificate Status Responders (CSR)***: CSRs will use the emerging Internet Online Certificate Status protocol to provide relying parties with an online, real time response to the question, "Has this end entity certificate been revoked or suspended?" CSRs will only be used for end-entity certificates, which will be the vast number of certificates in the Federal PKI. End-entity certificates will be changed and revoked much more frequently than CA certificates.

5.6.2. Operational Concept

The *Federal BCA* will be the unifying element to link otherwise unconnected agency CAs into a systematic overall Federal PKI. The BCA is not a root CA. It does not begin certification paths, it simply connects *trust domains* through cross certificate pairs to designated *principal CAs*. It is a bridge of trust. A FPMA will supervise BCA operation and establish the requirements for cross certifying with the BCA. These trust domains may be within the government or outside the government.

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

To provide maximum flexibility to Federal agencies and not intrude upon their prerogatives:

- Agencies will not be required to adopt the BCA's policies. Rather, agencies will retain the option to use other policies defined by their own internal PMAs, or by commercial certificate service providers.

- Agencies will not be required to use the BCA to interoperate with other Federal agencies or organizations outside the Federal government.

Alternatively, Federal agencies may communicate directly with an agency/organization to establish requirements for interoperating.

5.6.3. Federal PKI (FPKI) Steering Committee[29]

The mission of the Federal PKI (FPKI) Steering Committee is to provide clear, strong leadership within the U.S. Federal government during the development and implementation phases of the Federal PKI. This committee consists of over 50 members from two dozen Federal agencies and will:

- Provide guidance and assist in the development of an interoperable public key infrastructure that utilizes COTS standards-based products,
- 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.

[29] Information in this section is extracted from www.gits-sec.treas.gov.

CHAPTER 6

6. TESTING

Cryptographic services are provided using cryptographic modules (cryptomodules), which may include capabilities such as signature generation and verification (possibly involving key notarization), encryption and decryption, key generation, key distribution, etc.

If a large number of cryptographic modules are needed to provide security services in a system, then an undetected error in a cryptographic module's design could potentially affect the performance of a cryptographic function for every user in the system. For example, the verification of a chain of public key certificates might not function correctly, or key notarization (for secret keys) might be done improperly by a cryptomodule. Verifying a chain of public key certificates helps a signature verifier determine if a signature was generated with a particular key. Likewise, key notarization helps ensure that no party other than the signer of the data can use the data key to sign or encrypt information. If either of these functions were implemented incorrectly in a cryptomodule, the potential for the dissemination of weak cryptography could be introduced into the system, possibly allowing for signature forgery or the verification of invalid signatures. Therefore, it is important to have all cryptographic modules tested before distributing them throughout a system.

Figure 7 illustrates a general security testing model, including testing of cryptographic modules, and the various levels of testing that are required. This model, and the applicable testing organizations, is described in this chapter.

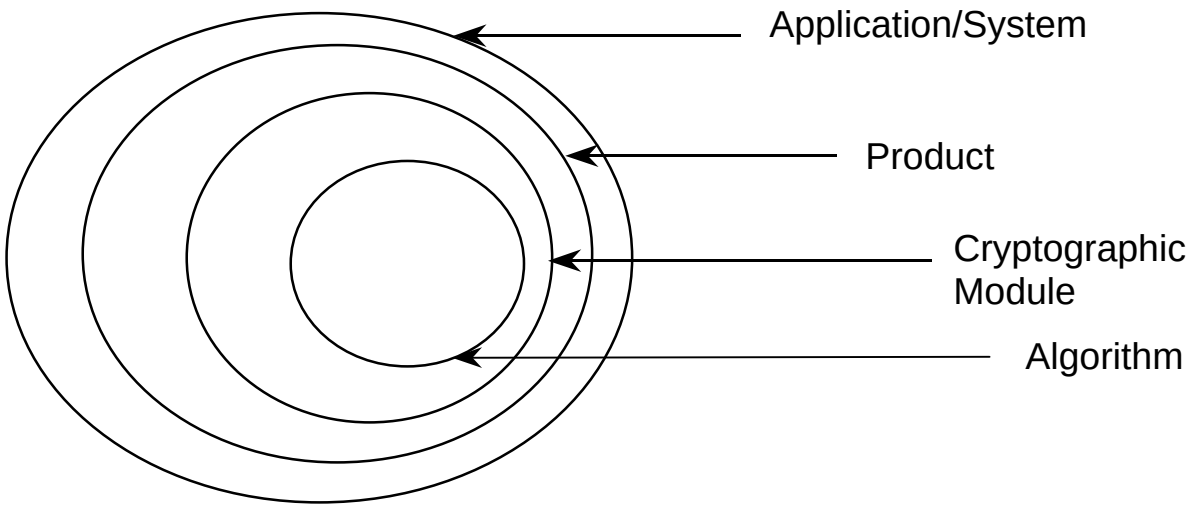


Figure 7. Testing Model

The following table illustrates the range of cryptographic methods that are tested, from individual algorithms (lowest level) to complete systems (highest level).

Level	**Example**	**Specification**
Application/System	Air Traffic Control	Common Criteria, Certification Guidance
Product	Security Module	Common Criteria
Security Module	Crypto Module	FIPS PUB 140-1
Algorithm	DES	FIPS PUB 46-3

At the lowest level are the cryptographic algorithms and cryptographic modules. These must be tested prior to integration into an existing or new system. The cryptographic modules are tested by the developer and then submitted to the Cryptographic Module Validation Program (CMVP) for testing against FIPS PUB 140-1, *Security Requirements for Cryptographic Modules*.

For all Federal agencies, including defense agencies, the use of encryption products that conform to FIPS PUB 140-1 is mandatory for the protection of sensitive unclassified information when the agency determines that cryptographic protection is required[30]. Agencies are required to use the standard in designing, acquiring, and implementing cryptographic-based security systems within computer and telecommunications systems (including voice systems).

6.1. Cryptographic Module Validation Program (CMVP)[31]

NIST and the Communications Security Establishment (CSE) of the government of Canada established the CMVP. The goal of the CMVP is to provide Federal agencies with a security metric to use in procuring equipment containing cryptographic modules. The results of the independent testing, by accredited laboratories, provide this metric. Cryptographic module validation testing is performed using the Derived Test Requirements (DTRs) for FIPS PUB 140-1. The DTRs list all of the vendor and tester requirements for validating a cryptographic module and are the basis of testing done by the Cryptographic Module Testing (CMT) accredited laboratories. This section includes five subsections: background information on the use of cryptographic modules, FIPS PUB 140-1 requirements, validated modules list, implementation guidelines, and testing requirements.

6.1.1. Background

A cryptographic module is a set of hardware, firmware or software, or some combination that implements cryptographic logic or processes. Examples include a standalone device such as a link encryptor, an add-on encryption board embedded in a computer system, and a software application running on a microprocessor such as a digital signature application. If the cryptographic logic is implemented in software, then the processor, which executes the software, is also part of the cryptographic module.

There are many advantages to using validated modules:

- Assurance that modules incorporate necessary features.

[30] National security-related information is excluded from this requirement.
[31] The information in this section is extracted from FIPS PUB 140-1.

- Protection of technical assets and staff time of government personnel by assuring that purchased products comply with a standard and have been tested.

- Provide users with a set of available and relevant security features.

- Increase flexibility to choose security requirements that meet application-specific and environment-specific requirements.

Figure 8 illustrates the CMV process. The process begins with the submission of the cryptographic module to one of the accredited laboratories. During the testing process, there are typically many interactions between the laboratory and the vendor and between the laboratory and NIST/CSE. NIST/CSE respond to questions about a specific validation and issue general implementation guidance that is applicable to all validations. The implementation guidance is not static, and is augmented as needed to respond to questions. The laboratory then writes the test report and submits it to NIST/CSE for validation. NIST/CSE review the test report and request clarification from the laboratory, as required. Finally, NIST/CSE issue the validation certificate and update the CMVP web site[32].

[32] FIPS 140-1, DTRs, implementation guidance, and validated modules list data are located at the web site: csrc.nist.gov/cryptval.

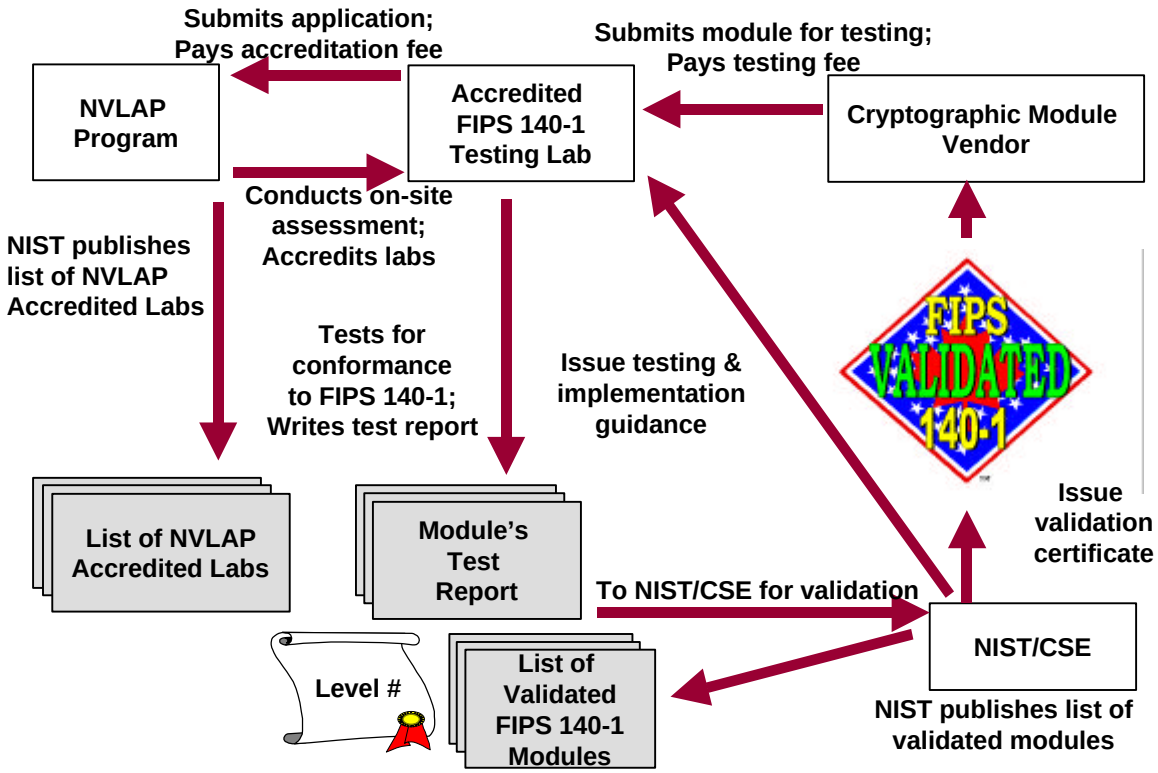


Figure 8. CMVP Process

In general, NIST/CSE responsibilities include:

- Reviewing reports and issuing validation certificates.
- Issuing CMVP policies.
- Issuing guidance and clarifications of FIPS PUB 140-1 and other cryptography standards (to labs, vendors, government organizations, and others).
- Assisting the National Voluntary Laboratory Accreditation Program (NVLAP) in laboratory assessments.

6.1.2. FIPS PUB 140-1 Requirements

The security requirements in FIPS PUB 140-1 cover 11 areas related to the design and implementation of a cryptographic module. Within most areas, a cryptographic module receives a security level rating of 1-4, from lowest to highest, depending on what requirements are met. For other areas that do not provide for different levels of security, a cryptomodule receives a rating that reflects fulfillment of all of the requirements for that area.

An overall rating is issued for the cryptomodule, which indicates:

(1) the minimum of the independent ratings received in the areas with levels, and

(2) fulfillment of all the requirements in the other areas.

On a vendor's validation certificate, individual ratings are listed as well as the overall rating. It is important for vendors and users of cryptographic modules to realize that the overall rating of a cryptographic module is not necessarily the most important rating. The rating of an individual area may be more important than the overall rating, depending on the environment in which the cryptographic module will be implemented (this includes understanding what risks the cryptographic module is intended to address). Modules may meet different levels in different security requirement areas; a module may implement identity-based authentication (level 3 or 4) and display tamper evidence (level 2).

Table 2 lists the security requirements for cryptomodules. To illustrate the various levels as described above, at Security Level 2, there are overall requirements for cryptographic modules, cryptographic algorithms, and self-tests; and specific requirements for roles and services and operating system security.

Table 2. Summary of Security Requirements

	Security Level 1	*Security Level 2*	*Security Level 3*	*Security Level 4*
Crypto Module	Specification of cryptographic module and cryptographic boundary. Description of cryptographic module including all hardware, software, and firmware components. Statement of module security policy.			
Module Interfaces	Required and optional interfaces. Specification of all interfaces and of all internal data paths.		Data ports for critical security parameters physically separated from other data ports.	
Roles & Services	Logical separation of required and optional roles and services.	Role-based operator authentication.	Identity-based operator authentication.	
Finite State Machine	Specification of finite state machine model. Required states and optional states. State transition diagram and specification of state transitions.			
Physical Security	Production grade equipment.	Locks or tamper evidence.	Tamper detection and response for covers and doors.	Tamper detection and response envelope.
EFP/EFT[33]	No requirements.			Temperature and voltage.
Software Security	Specification of software design. Relate software to finite state machine model.		High-level language implementation.	Formal model. Pre- and post-conditions.
Operating System Security	Executable code. Authenticated. Single user, single process.	Controlled access protection (C2 or equivalent).	Labeled protection (B1 or equivalent). Trusted communications path.	Structure protection (B2 or equivalent).
Key Management	FIPS approved generation/distribution techniques.		Entry/exit of keys in encrypted form or direct entry/exit with split knowledge procedures.	
Crypto Algorithms	FIPS approved cryptographic algorithms for protecting unclassified information.			
EMI/EMC	FCC Part 15, Subpart B, Class A (Business use). Applicable FCC requirements (for voice).		FCC Part 15, Subpart B, Class B (Home use).	
Self-Tests	Power-up tests and conditional tests.			

[33] Environmental Failure Protection/Environmental Failure Testing

6.1.3. Validated Modules List

The Validated Modules List includes the following information for each cryptographic module:

- Vendor Name and Point-of-Contact (POC)
- Module Name and Version Number
- Module Type (software, hardware, firmware)
- Date of Validation
- Level(s) of Validation
- Description of Module (or products which incorporate this module)

A module on the list may be a product used in multiple products from that vendor or used in another vendor's product(s).

6.1.4. Effective Use of FIPS PUB 140-1

When implementing cryptography in a system:

- Examine FIPS PUB 140-1. Consider the requirements in each area. Determine those requirements that specify the features that are desired. Determine those requirements (if any) specified in FIPS PUB 140-1 that were not originally considered. Specify the appropriate level in each area of the standard based on the acceptable level of risk.

- Obtain or develop cryptographic modules that meet or exceed the selected levels.

6.2. National Voluntary Laboratory Accreditation Program (NVLAP)

The NIST National Voluntary Laboratory Accreditation Program (NVLAP) accredits testing organizations based on technical accreditation requirements and quality system requirements. NVLAP assesses the testing organization against the NVLAP accreditation requirements to determine if the organization is competent to perform specific tests and calibrations. Competence is defined as the ability of a laboratory to meet the NVLAP conditions and to conform to the criteria in NVLAP publications for specific calibration and test methods.

6.3. Industry and Standards Organizations

The next higher level of testing, above algorithm and module testing, is at the product level. Products are tested by the vendor, standards organizations, and by independent verification and validation (IV&V) organizations. Vendors test their products to ensure that they function properly and in a secure manner. Cryptographic modules and components may be integrated or embedded into

these products. For government applications, the embedded cryptographic modules must meet the requirements of FIPS PUB 140-1. At this level of testing, it is important to ensure that the product does not compromise or circumvent the cryptographic features, resulting in a non-secure device. Currently, products may be tested to the *CC* or to the *Trusted Computer System Evaluation Criteria*, DOD-5200.28-STD.

6.3.1. National Information Assurance Partnership (NIAP)[34]

The National Information Assurance Partnership (NIAP)[SM] is a U.S. Government Initiative designed to meet the security testing needs of both information technology producers and users. The program is intended to foster the availability of objective measures and test methods for evaluating the quality of IT security products. In addition, it is designed to foster the development of commercial testing laboratories that can provide the types of testing and evaluation services which will meet the demands of both producers and users.

NIAP is a collaboration of NIST and the National Security Agency (NSA). NIAP will develop tools, test methods and tests for specification-based IT security products. This means that the security functionality and assurance requirements fo a product or system must be formally described or specified. These specifications then form the basis for the development and conduct of tests for the product or for a class of products (e.g., for a firewall, an access control device, or even a network router).

6.4. Certification and Management Authorization

The highest level of testing is at the application or system level. At a Federal agency, this level of testing is certification testing. Certification is the comprehensive analysis of both the technical and non-technical security controls and other safeguards of a system. Certification testing establishes the extent to which a particular system meets the security requirements for its mission and operational needs. Certification is performed in support of management ‘s authorization to operate a system. Certification examines the system in the operational environment and examines external systems that are networked to the system under test (SUT). One of the major tasks of certification testing is to verify that external systems should not be able to compromise or circumvent the security features (including cryptographic features) of the SUT. Certification requires examining not only the technical controls but also all the other security controls, for example, physical controls, administrative procedures, and personnel controls. For Federal agencies, it is recommended that certification testing be performed by a department or organization that is *not* the developing organization. This is to ensure independence and objectivity in the testing.

[34] This information was extracted from niap.nist.gov.

Certification testing may be performed against several standards, including the CC and agency-specific requirements.

At all levels of testing, it is important to be able to trace the implemented cryptographic controls and other security features through the requirements back to a standard.

CHAPTER 7

7. SELECTING CRYPTOGRAPHY - THE PROCESS

The process used to select cryptographic methods is similar to the process used to select any IT method. This selection process is documented in the system development life cycle model that contains four phases: planning, definition, acquisition, and operations (including disposal). The system development life cycle model may be embedded in any of the three major system developmental approaches:

- *waterfall* - the phases are executed sequentially;

- *spiral* - the phases are executed sequentially with feedback loops to previous phases; or

- *evolutionary* - there is replanning at each phase in the life cycle based on feedback. Each phase is divided into multiple project cycles with deliverable measurable results at the completion of each cycle.

The goal of the selection process is to specify and implement cryptographic methods that address specific agency/organization needs.

Prior to selecting a cryptographic method, an agency should consider the operational environment, requirements of the application, the types of services that can be provided by each type of cryptography, and the cryptographic objectives that must be met when selecting applicable products. Based on the requirements, several cryptographic methods may be required. Also, both secret key and public key cryptography may be needed in one system: each performing different functions.

The following questions should be addressed in determining the appropriate degree of security, including cryptography, which will be required for a system:

- How critical is the system in meeting the organization’s mission?

- What are the security/cryptographic objectives required by the system, e.g., integrity, confidentiality?

- What regulations and policies are applicable in determining what is to be protected?

- What are the threats that are applicable in the environment where the system will be operational?

- Who selects the protection mechanisms that are to be implemented in the system?

- Are the users knowledgeable about cryptography and how much training will they receive?

The answers to these questions can be used to formulate a strong developmental approach to integrating cryptographic methods into existing or new systems. A sound approach in integrating cryptographic methods is to develop requirements that are derived from the protection goals and policies for the system. The goals and policies are derived from a risk assessment. The following areas relate specifically to cryptography and should be included when developing requirements:

- security of the cryptographic module
- hardware versus software implementation
- applying cryptography in a networked environment
- implementing FIPS-validated algorithms
- secret key versus public key cryptography
- key management

It is important to be able to demonstrate *traceability* from the requirements back to the policies and goals and associated risk assessment.

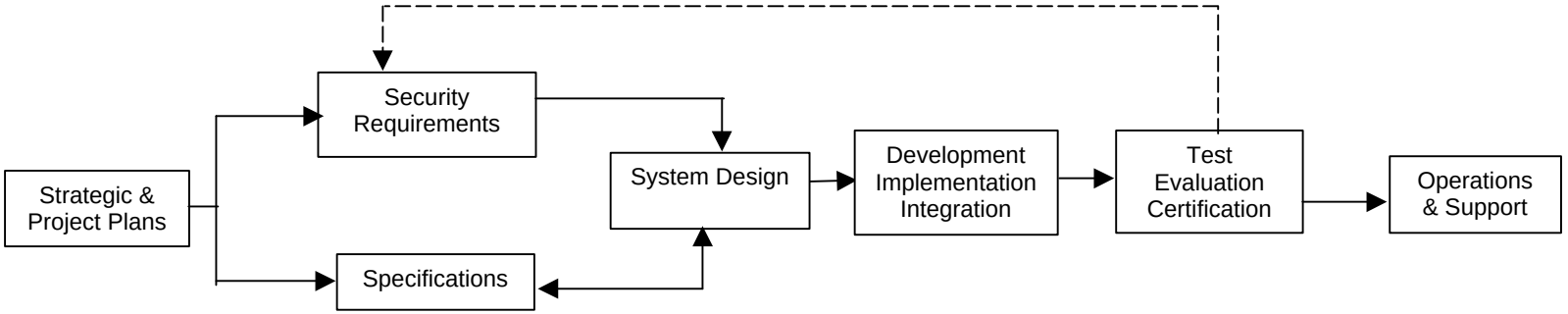
There are other issues to be addressed in achieving overall security. Cryptography is best used when it is designed as an integrated part of the system, rather than as an add-on feature. When this cannot be done, cryptographic functions should be carefully added so that the security that they are intended to provide is not compromised. The least effective approach to implementing cryptography is to immediately begin implementing technical approaches. (Note: implementing technical solutions without determining the requirements is **never** recommended.) Also, cryptographic methods are intended to address specific security risks and threats. Therefore, implementing only cryptographic methods, and no other security mechanisms in a system, will not necessarily provide adequate security. The example described in section 8.3 (Treasury Electronic Certification System) provides an illustration of selecting and implementing cryptographic methods. As illustrated by the example, a cryptographic solution may be initially implemented as a pilot project to ensure the solution is effective.

By consistently replacing traditional methods, the security and efficiency of a system improves. Benefits from replacing handwritten signatures with electronic or digital signature techniques include reducing the possibility of forgery, reducing administrative processing time, and decreasing the burden of maintaining "traditional" paperwork. A system implementing cryptography will naturally generate new types of documentation, and the cryptographic

technology should be applied in handling that documentation. Security officers, for example, may have to generate and sign requests for keys or cryptographic modules. Instead of using paper forms, electronic forms could be generated, signed, and sent to the appropriate parties, who can verify the signatures and act on the request in a very timely manner.

Figure 9 illustrates the system development life cycle model phases and the tasks that are to be performed within each phase when specifying, selecting, and implementing cryptographic methods. These tasks should be performed when acquiring and implementing new systems requiring cryptographic products or when acquiring cryptographic products for existing systems. The tasks listed in Figure 9 and discussed in this guideline are tailored to cryptographic methods. Also, because of the role cryptographic methods play in protecting sensitive information, greater emphasis should be placed on developing applicable documentation, (e.g., user procedures and crypto-officer manuals) and implementing ongoing operational controls, (e.g., key management).

In general, the tasks are listed in sequence; (e.g., identifying potential countermeasures) is performed prior to executing certification testing. Realistically, some of the tasks may be executed concurrently, for example, performing a risk assessment and developing objectives; and selecting and implementing controls and developing applicable documentation. To ensure that a cryptographic product is correctly implemented to provide appropriate security functionality, **all** tasks should be performed, particularly documentation development, training, and ongoing operations tasks. The phases are described in more detail in the following sections, with a focus on cryptography.



Planning	*Definition*	*Acquisition*	*Operations*
Specify policies & laws - Federal - organizational - system - application Document security environment Perform risk assessment - threats & threat agents - assets - vulnerabilities Develop objectives - confidentiality - integrity - authentication - non-repudiation - IT and non-IT security objectives	Develop cryptographic security requirements/specifications (including cryptographic services, e.g., crypto algorithms, key management) - functional - assurance - environmental Specify cryptographic requirements in an RFP Develop product selection criteria, e.g., - FIPS PUB 140-1 validated - algorithms conformance tested Identify potential cryptographic countermeasures	Vendor submits product/module for FIPS PUB 140-1 validation (if not already validated) Configure crypto module Select and implement technical and assurance controls, e.g., - configuration management - encryption method - encryption algorithm - key generation Develop applicable documentation, e.g., user and crypto officer guides Perform IV&V/certification testing using - test criteria - methodology - test scenarios	Operational practices, e.g., - training - physical security - admin/personnel controls Operational controls, e.g., - crypto key management - self tests, e.g., power-up tests, conditional tests - product/module maintenance Vendor submits product for FIPS PUB 140-1 revalidation (if required)

Figure 9. Cryptographic Method Selection Process

7.1. Planning Phase

In the Planning Phase, the goal is to document the objectives that the potential cryptographic methods/techniques are intended to address. These objectives are partially based on applicable policies and regulations. Objectives are also derived from the existing (or proposed) security environment and a preliminary risk analysis with identified threats and vulnerabilities. Policy identification and development, risk assessment, and objective development tasks are described in the following sections.

7.1.1. Security Policies

According to NIST Special Pub. 800-12, *An Introduction to Computer Security: The NIST Handbook*, computer security policy is defined as the "documentation of computer security decisions." This includes management's directives to: (1) create a computer security program, establish its goals, and assign responsibilities; (2) specify security rules for particular systems, and (3) develop policies such as an organization's e-mail privacy policy or fax security policy. After policies are established, requirements (including security and cryptographic requirements) are specified and an overall system design is developed. The system design includes software and hardware implementations, procedures, environmental requirements, physical security considerations, etc.

Typically, policies are based on the need to protect IT resources, data and information. Security policies exist at all levels of an organization, from agency-level policy to application specific policy. There are also security policies that are applicable to all organizations or agencies - these are at the Federal and state level. Federal-level policies relative to IT security and, more importantly, to the use of cryptographic techniques give a consistent security direction for the protection of Federal IT resources. Policies at the Federal level and at the agency (or organizational) level can be used in making protection decisions at the application level.

1. **Federal** policies provide guidance on using cryptographic techniques, for example, guidelines for replacing handwritten signatures and agency responsibilities for protecting information. Federal policies are developed by Congress and are applicable to all relevant agencies and systems, (e.g., The *Privacy Act*).

2. **Agency** policies include the mission statements regarding the security (confidentiality, availability, integrity, non-repudiation) of the application systems that support the mission. Agency policies set the organizational strategic directions for security and assign resources for their implementation. Agency policies may include *issue-specific* policies that focus on areas of current relevance or concern to the organization.

Agency policies are typically developed by upper-level management because they affect the entire agency/organization.

Agency policies directly impact program policy by providing guidance on what should be included in a program policy. For example, agency policy on digital signatures may require a specific system to implement a specific cryptographic method. Also, agency policy on protecting the confidentiality of certain data records may result in the development of a confidentiality policy enforced through the implementation of secret key encryption.

Compliance to an agency policy is critical to the effective selection and use of security methods. Program policies impact lower level policies and should be used as a starting point for writing system, program, and application specific policies. Similar to security requirements, it is important to have *traceability* from the lowest level policy to the highest level policy - to show how the low-level policies were derived and to ensure that their specificity accurately reflects the needs of the organization.

Agency policies may provide the justification for system policy to require the use of cryptography to replace or supplement conventional security controls or procedures.

4. **Application** policies are very specific in setting usage and configuration guidance for cryptographic controls. These policies consist of security objectives and operational security rules. These policies are system specific.

   The developers and users of a system typically produce application policies. Users understand how their systems operate and the goals and objectives each system addresses. Developers identify security features that meet a policy and the associated costs for that feature. Application policies may vary in length from one page to several sections depending on the sensitivity of the information that needs to be protected and the number of risks to be mitigated.

Example 1 illustrates a sample policy hierarchy. The Federal policy is the most general and the application policy is the most specific. This hierarchy illustrates the traceability of the lowest policy (application) to the highest (Federal).

Example 1. Sample Policy Hierarchy

POLICY TYPE	POLICY STATEMENT
Federal Policy	Protection of information must be commensurate with the risk of unauthorized disclosure.
Agency Policy	Information that is accessible via the network will be protected from disclosure using cryptographic techniques.
Application Policy	On-line access to employment candidates will be made available to those in the personnel office and those managers currently hiring new employees. Access to candidate information is available over the network using the application, cryptographic software implementing confidentiality controls, and proper user authentication.

Policies (and applicable laws and regulations) can be used effectively in the design, development and implementation of cryptography-based controls and procedures, if they are implemented in a practical (real-world) manner.

The next major task in the planning phase is to perform a risk assessment and, specifically, identify the unique requirements associated with each IT system. After the risk assessment has been performed, policies should be developed regarding the use of evaluated operating systems and validated cryptographic modules in a range of environments. Also, policies that have been previously written may need to be revised or tailored throughout the system life cycle.

The following are some topics that should be addressed when developing cryptography policies and requirements:

1. Policies regarding algorithm usage and algorithm parameter configuration,

2. Policies regarding the classes of users (e.g., crypto-officers, networked users, operators) that may use the cryptographic methods and assigning associated privileges,

3. Identification and authentication requirements when a user first accesses a system or cryptographic module,

4. Procedures employed when adding, modifying, or deleting users and user privileges associated with cryptographic methods/products,

5. Policies defining when confidentiality controls, integrity controls, and advanced authentication techniques are required,

6. Security measures relating to the physical environment of a cryptographic method/product,

7. Audit procedures,

8. Guidelines for requiring non-repudiation,

9. Guidelines for performing risk assessments to:

   - Ensure the unique risks of an IT system are considered,

   - Evaluate the potential risks and determine the level of control required to minimize the risks, commensurate with the cost or value of the data,

10. Key Management including key distribution, generation, use, destruction, and archiving.

11. Backward compatibility of software/hardware and architecture.

12. Forward compatibility with envisioned future developments such as new cryptographic techniques, digital signature systems, authentication mechanisms, FIPS PUBs, and

13. Interoperability between governments, commercial communities, law enforcement communities, etc.

Example 2 lists sample policy statements that are applicable to a Public Key Infrastructure (PKI) Policy Certification Authority (PCA) application. Because the policies are for a specific application and technology, they are at a low level of detail.

Example 2. Application Policy Statements

- *Identification and Authentication (I&A) Requirements:* When a user registers his or her certificate, the user must provide required I&A information to the certificate issuer to prove that he/she is indeed the claimed person.

- *Security Controls:* Each PCA must specify the security measures that it will employ for (1) the hardware and software that are used for certificate generation and signing and (2) maintaining certificate revocation lists (CRLs).

- *Audit Procedures:* Each PCA must specify the procedures for manual audits. The procedures may include a schedule of the manual audit and may also include provisions for impromptu audits.

7.1.2. Risk Assessment

Risk management consists of two components:

- *Assessing* risks using a risk-based approach to determine the impact of given losses and the probability that these losses will occur. The major losses addressed by cryptographic methods are the unauthorized disclosure and modification of data.

- *Selection and implementation* of countermeasures that either reduce the probability of threat occurrence or minimize the impact of loss. The goal is to reduce the risk to an acceptable level.

The purpose of an IT risk management process is to ensure that the impacts of threats are known and that cost-effective countermeasures are applied to determine *adequate security* for a system. *Adequate security* is defined in OMB Circular A-130, Appendix III, as "security commensurate with the risk and magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of information. This includes assuring that systems and applications used by an agency operate effectively and provide appropriate confidentiality, integrity, and availability through the use of cost-effective management, personnel, operational, and technical controls." This definition explicitly emphasizes the risk-based policy for cost-effective security established by the Computer Security Act.[35]

Risk assessment, the process of analyzing and interpreting risk, includes the following activities.

[35] The Appendix no longer requires the preparation of formal risk analyses. Rather than continue to try to precisely measure risk, security efforts are better served by generally assessing risks and taking actions to manage them.

- Identify assets
- Assess current security and protection mechanisms

- Identify and classify threats affecting:
  - Integrity
  - Confidentiality
  - Authentication
  - Non-repudiation
  - Availability

- Identify potential losses
  - Classify potential losses by criticality and sensitivity
  - Quantify cost of loss

- Develop risk scenarios
  - Develop risk measurement levels

- Identify potential countermeasures/safeguards
  - Evaluate potential countermeasures so that implementation decisions can be made

  - Perform cost/benefit analysis for proposed countermeasures. (The analysis should include both *monetary* and *non-monetary* perspectives.)

- *Risk mitigation* involves the selection and implementation of security controls to reduce risk to an acceptable level. In Federal agencies, the common method to select security controls (including cryptographic methods) is to develop a Request for Proposal (RFP) and select the proposal that provides the best solution.

The following two examples illustrate how cryptographic methods can address integrity and non-repudiation threats.

Example 3. Threat Mitigation

Security Control to Mitigate Threat to Integrity: Both secret key and public key cryptography can be used to ensure integrity. When secret key cryptography is used, a data authentication code (DAC)[36] is generated. Typically, a DAC is stored or transmitted with the data. When the integrity of the data is to be verified, the DAC is generated on the current data and compared with the previously generated DAC. If the two values are equal, the integrity (i.e., authenticity) of the data is verified.

Public key cryptography verifies integrity by using public key signatures and secure hashes. A secure hash algorithm is used to create a message digest (hash). The hash will change if the message is modified. The hash is then signed with a private key. The hash may be stored or transmitted with the data. When the integrity of the data is to be verified, the hash is recalculated and the corresponding public key is used to verify the integrity of the message.

Security Control to Mitigate Threat to Non-repudiation: Data is electronically signed by applying the originator's private key to the data. The resulting digital signature can be stored or transmitted with the data. Any party using the public key of the signer can verify the signature. If the signature is verified, then the verifier has confidence that the data was not modified after being signed and that the owner of the public key was the signer. A certificate *binds* the public key to the identity of the signer.

Typically, a risk assessment is performed for all new and existing systems, even if it is not called a formal risk assessment. The type of risk assessment that is performed is usually a qualitative analysis, rather than a formal quantitative analysis and the results are used in developing the system requirements and specifications. A team comprised of users, system developers, and security specialists typically conducts the risk assessment. The scope of this task varies depending on the sensitivity of the information and number and types of risks that need to be addressed. For systems with minimal security requirements, the risk assessment may be a few pages in length.

7.1.3. Security Objectives

The third major task in the planning phase, in addition to specifying policies and performing a risk assessment, is to develop security objectives. These objectives are at a high level and should address security, in general, and cryptography, in specific. Example 4 lists sample security objectives.

[36] A DAC is commonly referred to as a Message Authentication Code (MAC).

Example 4. Security Objectives

1. Security mechanisms should be able to evolve as technology evolves.

2. *Integrity*: The correctness of cryptographic keys and other critical security parameters must be preserved. Authentication, authorization and non-repudiation should be supported. The correctness of the security mechanisms/features should be ensured.

3. *Availability*: The security mechanisms/features should be continually available (at least 99.5% of the time). Availability periods must be tailored to particular systems or environments. Response time to suspected compromise, for example, disclosure or modification, should be minimized. Systems should be responsive and adaptable to changing security requirements and threats.

4. *Assurance*: An acceptable level of assurance should be maintained to ensure that the security mechanisms/features are operating correctly. There should be no increase in vulnerability to an individual system due to a connection to external systems or networks, e.g., the Internet. A risk assessment should be performed prior to linking a system to an external system to determine the level of risk.

5. *Authentication*: The security mechanisms/features must provide at least the same level of accountability as paper-based systems. Accountability is typically accomplished by identifying and authenticating users of the system and subsequently tracing their actions. User accountability should be limited to security-relevant conditions. Accounting data should be accessible to designated crypto-officers and operators. Accountability should be reflected in an audit trail.

6. *Digital signatures:* Digital signature techniques may be used to validate the:
   - identity of the signer of a message and
   - integrity of the information received from the signer of that information.

   Digital signatures may represent an individual or an entity (system).

7.2. Definition Phase

In the Definition Phase, the objective is to develop the requirements/specifications for the proposed cryptographic methods. After the requirements have been developed, general selection criteria based on these requirements are produced. Finally, categories of methods that meet the requirements are identified. The security requirements are based on user needs and estimates of an organization's resources to meet proposed requirements.

Requirements should be detailed - this aids in product selection, implementation, and testing.

7.2.1. Security Requirements/Specifications

The cryptographic requirements may be divided into three categories: functional, assurance, and environmental requirements. (The following definitions are taken from the CC.)

Functional requirements describe the expected security behavior of a product/system and are intended to meet the security objectives.

Assurance requirements ensure that an IT product or system meets its security objectives. Assurance that the security objectives are achieved by the selected security functions is based on the following two factors:

a) confidence in the correctness of the implementation of the security functions, i.e., the assessment whether they are correctly implemented; and

b) confidence in the effectiveness of the security functions, i.e., the assessment whether they actually satisfy the stated security objectives.

Environmental requirements are intended to counter threats and risks in the operating environment of the product/system and/or cover any identified organizational security policies and assumptions. Environmental requirements generally apply to the system, rather than to specific cryptographic methods. Environmental requirements address the physical and operational controls that are applicable to the system.

Environmental requirements are one component of the security environment. The *security environment* includes all the laws, organizational security policies, customs, expertise and knowledge that are determined to be relevant. The security environment also includes the threats to security that are present in the physical environment.

Table 3 identifies security objectives and requirements for cryptographic components that may be addressed by cryptographic methods and techniques. The purpose of the table is to provide individuals with a *roadmap* to identifying cryptographic functional and assurance requirements in the CC classes that will meet the needs of a system in an organization. After the functional and assurance classes are selected, the specific requirements are extracted from the CC. The specific requirements may need to be refined/tailored to meet security objectives. The tailoring operations are defined in the CC. Table 3 is not intended to list all the specific CC requirements; rather it serves as a reference guide to the CC classes. To effectively use the table, it is important to have

documented the objectives that must be addressed. These objectives were developed in tasks in the Planning Phase.

- *Column 1*, *Cryptographic Category,* covers the areas related to the design and implementation of a cryptographic product/module. Some examples are roles and services, physical security, and cryptographic key management.

- *Column 2*, *Security Objectives*, lists the security objectives applicable to a cryptographic category.

- *Columns 3 and 4*, *Functional and Assurance Requirements,* list the functional and assurance requirements that address the security objectives for a cryptographic category. The functional and assurance requirements are listed by *CC* class, for example, audit, cryptographic support, identification and authentication, configuration management, guidance documents, development.

- *Column 5* contains *Procurement Recommendations* to ensure the cryptographic requirements are adequately addressed.

Table 3. Cryptographic Technical and Assurance Requirements

Cryptographic Category	Security Objectives	Functional Requirements	Assurance Requirements	Procurement Recommendations
Cryptographic Module: cryptographic boundary; diagram configuration; security policies	IT security objectives[37]	*Rules of the System (OMB Circular A-130, Appendix III)*	*Development* (CC) - includes functional specifications; high and low level design; and internal structure	The functional and assurance requirements for the cryptographic module should be consistent[38] with the requirements for the other components of the system/product.
Module Interfaces: logical data paths	IT security objectives	*User Data Protection* (CC) - includes security function policies; forms of user data protection; offline storage; import and export; and inter-product communications	*Development* (CC); *Guidance Documents* (CC) - includes administrative and user guidance documentation	The functional and assurance requirements for the cryptographic module should be consistent with the requirements for the other components of the system/product.

[37] Examples: The module must be designed and implemented in a manner which ensures that the security policies are enforced. The module must provide all the functions and capabilities necessary to support the authorized administrators that are responsible for the security management.

[38] *Consistency* ensures that security requirements are not compromised by other system requirements and that system features are implemented to address the security requirements.

Table 3. Cryptographic Technical and Assurance Requirements (Continued)

Cryptographic Category	Security Objectives	Functional Requirements	Assurance Requirements	Procurement Recommendations
Roles and Services: roles and associated services; authorization and access control mechanisms	- Authentication[39]	*User Data Protection* (CC). *Identification and Authentication* (CC) - includes user identification; user attribute definition; user authentication; user subject binding; and authentication failures. *Security Management* (CC) - includes security management roles; management of security attributes, functions, and data; revocation; security attribute expiration. *Trusted Path/Channels* (CC) - includes inter-component trusted channels.	*Development* (CC); *Guidance Documents* (CC)	A system administrator may include the cryptographic officer role.

[39] Example: The module must ensure that only authorized users gain access to the module and its resources.

Table 3. Cryptographic Technical and Assurance Requirements (Continued)

Cryptographic Category	Security Objectives	Functional Requirements	Assurance Requirements	Procurement Recommendations
Physical Security: physical security configuration and mechanisms; specify features or testing procedures. (Includes *EFP/EFT*).	Environmental security objectives	*Protection of the TOE*[40] *Security Functions* (CC)	*Development* (CC)	Verify that the physical controls adequately protect the cryptographic module.
Software Security: describe the design of the software; correspondence between the design and the security policy.	IT security objectives	No trapdoors or trojan horses	*Development* (CC). *Configuration Management (CM)* (CC) - includes CM scope and capabilities and use of automated tools. *Life Cycle Support* (CC) - includes development security, flaw remediation, life cycle definition, and tools and techniques.	The use of CM tools and life cycle support products should be the same for the cryptographic module and the complete system/product.

[40] TOE - Target of Evaluation. An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

Table 3. Cryptographic Technical and Assurance Requirements (Continued)

Cryptographic Category	Security Objectives	Functional Requirements	Assurance Requirements	Procurement Recommendations
Operating System Security: access, authorization, audit controls; identify critical security parameters and cryptographic data	IT security objectives	TCSEC C2, B1, or B2 operating system. *User Data Protection* (CC). *Identification and Authentication* (CC). *Trusted Path/Channels* (CC). *Security Management* (CC). *Protection of the TOE Security Functions* (CC) - includes fail secure; trusted recovery; availability, confidentiality, and integrity of exported data; and physical protection. *Audit* (CC) - includes audit event selection, audit data generation, audit review, audit event storage, and audit analysis.	Applicable *Evaluation Assurance Level (EAL)* from the CC	Verify that the overall system/product physical controls are adequate. Include applicable cryptographic module events in the audit able events.

Table 3. Cryptographic Technical and Assurance Requirements (Continued)

Cryptographic Category	Security Objectives	Functional Requirements	Assurance Requirements	Procurement Recommendations
Cryptographic Key Management: key generation, distribution, input, use, output, storage, destruction, and archiving	- Authentication	*User Data Protection* (CC). *Security Management* (CC). *Protection of the TOE Security Functions* (CC). *Trusted Path/Channels* (CC). *Cryptographic Support* (CC) - includes cryptographic key management.	*Development* (CC); *Guidance Documents* (CC)	Include the key management procedures in the administrative guidance and user responsibilities in the user guidance.
Cryptographic Algorithms: identify FIPS-approved algorithms and other cryptographic algorithms	- Confidentiality - Non-repudiation - Authentication - Data Integrity	*Cryptographic Support* (CC) - includes cryptographic operations. *Communication* (CC) - includes non-repudiation of origin and receipt.	*Tests* (CC) - includes coverage and functional tests	Verify the module or product[41] is on the FIPS 140-1 validated modules list. No unique requirements beyond specifying the required algorithms.

[41] A validated cryptographic module may be embedded in a product that is submitted for validation.

Table 3. Cryptographic Technical and Assurance Requirements (Concluded)

Cryptographic Category	Security Objectives	Functional Requirements	Assurance Requirements	Procurement Recommendations
Self-Tests: identify power-up and conditional tests	Detect errors in the operation of the cryptographic module; prevent compromise of critical security parameters.	*Protection of the TOE Security Functions* (CC)	*Guidance Documents* (CC)	No unique requirements beyond specifying the tests for the required algorithms.
EMI/EMC[42]*:* FCC conformance requirements		EMI/EMC FCC part 15, Subpart B, Class A (business use) or Class B (home use) requirements	No unique assurance requirements.	No unique requirements beyond specifying the FCC requirements.

[42] Electromagnetic Interference/Electromagnetic Compatibility

7.2.2. Cryptographic Method Example

The following example focuses on a specific cryptographic method, digital signatures, and illustrates how requirements may be derived from a high-level digital signature policy statement.

Example 5. Digital Signature Policy

Background: Historically, handwritten signatures were used to provide authenticity and liability for a document. The proposed successor to handwritten signatures is digital signatures.

*Policy Statement***.** Digital signatures will be accepted as valid only if the user who verifies a signature has an acceptable level of assurance of the integrity of the electronic document that was signed and the identity of the signer of that document. In addition, the verifier must be able to **trust** that the signer will be held legally responsible for the information content of the document.

One of the digital signature policies is to ensure the integrity of electronic documents and provide non-repudiation of document origin. The requirements resulting from this policy include all three types: functional, assurance and environmental.

Example 6. Digital Signature Requirements

Requirements:

1. *Document preservation*. Associated signatures and the certificates necessary to verify those signatures must accompany electronic documents. A record of certificate validity must also be kept along with an audit trail of document movement. Expert testimony about this entire procedure and the audit data collected will lay the foundation for the testimony if documents are required as evidence.

2. Digital signatures do not, by themselves, provide time-related information. A trusted time stamp is required to prove when a document was originated or received. This service must be provided by a trusted third party, which may be serving the purpose of a notary. The originator will generate a hash of the document and send a copy of the document and the hash to a private sector vendor serving as a notary. This trusted party could time and date stamp the hash of the document, store a copy of the hash and the document, keep an audit log of the action, and serve as an intermediary between the document's originator and receiver.

3. *Establishment of user and CA responsibility*. The document signer must be responsible for protecting the private key used to sign a document and obtaining time stamped document receipts, when required. A document verifier must ensure that all certificates used to verify a received document are valid at the time the document is received, the received document is time stamped, and the required information is archived in case of litigation. The CA is responsible for protecting the private key used to sign certificates, establishing the identity of its subscribers, and providing certificates and revocation information in a timely manner.

4. Each entity, whether an originating or sending entity or a CA, must maintain an audit log of digital signature related activity, including messages sent and received, activity by persons associated with the signature process and other security-relevant events.

5. Policies and procedures must be established to ensure that control is maintained on all processes involving the electronic authorization and authentication of electronic documents.

6. Policies and procedures must be established that will ensure that an approved process protects the distribution and communication of authorities.

7.2.3. Selecting Cryptographic Countermeasures

The final task in the definition phase is to identify categories of cryptographic methods/techniques that meet the requirements and mitigate the specific risks. (Note: there may be more than one method category that can mitigate each risk. For example, both DES and digital signatures can protect against the undetected modification of data.) For many of the methods, there are assurance features that increase the confidence that the method performs correctly.

Table 4 lists the technical and assurance features that meet the technical and assurance requirements documented in Table 3. The features in Table 4 map directly to the requirements listed in Table 3.

- *Column 1* lists the Cryptographic Category.

- *Column 2* identifies the risks and attacks that apply to a cryptographic category, for example, unauthorized access or unauthorized disclosure.

- *Columns 3 and 4* list the technical and assurance features that are applicable to a cryptographic category and mitigate the potential risks. Where applicable, the technical and assurance features are numbered and listed in ascending order of protection, to address increasing levels of risk.

The levels vary from 1 to 4, corresponding to the security levels in FIPS PUB 140-1.

- *Column 5* lists the FIPS PUBs that describe the technical features. The information included in the cryptographic category columns is the same as that presented in the requirements table (Table 3). This provides for traceability from the requirements to the methods and features.

Tables 3 and 4 illustrate traversing from a high level of abstraction in the requirements to a lower level of granularity in identifying specific features. It is important to understand that Table 4 does **not** specify the necessary conditions for the secure implementation of a product in a particular system/application. This task is left to those who implement the system.

Table 4. Risks and Cryptographic Technical and Assurance Features

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Cryptographic Module: specify cryptographic boundary; diagram configuration; specify security policy; describe operational and error states	- Incorrect implementation	*No unique cryptographic technical features required. Cryptographic requirements addressed in overall system/product requirements.*	Security policy (including security rules), configuration block diagram	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*
Module Interfaces: specify physical and logical data input and output paths	- Unintentional output of plaintext data - Design error	Physical/logical separation of data input /output ports,control input, status output, data input, data output,	Documentation of the interfaces and input and output data paths	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*
Roles and Services: identify roles and associated services; specify authorization and access control mechanisms	- Unauthorized access by authorized/ unauthorized individuals - Masquerade - Password compromise - Replay attacks	1. Role-based authentication mechanisms. 2. identity-based authentication mechanisms, maintenance-access interface	Documentation of the authorized roles, services, operations, and functions	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*

Table 4. Risks and Cryptographic Technical and Assurance Features (Continued)

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Roles and Services (continued)	(same as above)	- token based authentication - biometrics based authentication - cryptographic authentication protocols (secret key and public key cryptosystems)	*No unique cryptographic assurance features required. Cryptographic requirements addressed in overall system/product requirements.*	FIPS PUB 190: *Advanced Authentication*
Roles and Services (concluded)	(same as above)	- digital signature algorithm - digital signatures - random/pseudorandom number generator - unilateral authentication protocol - mutual authentication protocol	*No unique cryptographic assurance features required. Cryptographic requirements addressed in overall system/product requirements.*	FIPS PUB 196: *Entity Authentication Using Public Key Cryptography*

Table 4. Risks and Cryptographic Technical and Assurance Features (Continued)

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Physical Security: specify physical security configuration and mechanisms; specify features or testing procedures. (Includes *EFP/EFT*).	- Unauthorized physical access to the contents - Unauthorized use or modification, e.g., module substitution - Unusual environmental conditions or fluctuations that results in disclosures of critical security parameters - Unauthorized disclosure of plaintext critical security parameters	1. Production grade enclosures. 2. tamper evidence, or tamper resistance. 3, 4. Tamper response of shutdown of the module; zeroization of plaintext security keys and other unprotected critical security parameters (CSPs).	1, 2, 3. Specification of the physical embodiment, description of the applicable physical security mechanisms, 4. specification of the environmental failure protection features, documentation of the environmental failure tests performed and the results of those tests	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*
Software Security: describe the design of the software; explain the correspondence between the design and the security policy.	- Incorrect/invalid operation of the module	*No unique cryptographic technical features required. Cryptographic requirements addressed in overall system/product requirements.*	1, 2, 3. Finite state machine model including state transitions, rules of operation, source code listings. 4. formal model, informal proof	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*

Table 4. Risks and Cryptographic Technical and Assurance Features (Continued)

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Operating System Security: specify access, authorization, audit controls; identify CSPs and cryptographic data	- Unauthorized access by authorized/ unauthorized individuals - Undetected modification of cryptographic component - Unauthorized modification, substitution, insertion, and deletion of cryptographic keys and other CSPs	1. No protection. 2. restricted access (DAC) (TCSEC[43] C2) to plaintext CSPs. 3, 4. labeled protection (TCSEC B1 or B2) of cryptographic software and other critical security parameters; structured protection of cryptographic software and other CSPs.	*No unique cryptographic assurance features required. Cryptographic requirements addressed in overall system/product requirements.*	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*
Cryptographic Key Management: specify key generation, distribution, input, use, output, storage, archiving, destruction.	- Unauthorized disclosure, modification, and substitution of secret/private keys - Unauthorized substitution and modification of public keys	key entry/output: 1, 2. plaintext. 3,4. encrypted keys or split knowledge for manual-distribution. key destruction: zeroize all plaintext cryptographic keys and other unprotected CSPs	Specification of the FIPS-approved key generation algorithm, documentation of the key distribution techniques	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*

[43] Trusted Computer System Evaluation Criteria

Table 4. Risks and Cryptographic Technical and Assurance Features (Continued)

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Cryptographic Key Management (concluded)	(same as above)	- NIST-approved key generation algorithms - Use of error detection code (message authentication code) - Encrypted IVs - Key naming - Key encrypting key pairs - Notarization of keys - Random/pseudorandom number generation	*No unique cryptographic assurance features required. Cryptographic requirements addressed in overall system/product requirements.*	FIPS PUB 171: *Key Management*
Cryptographic Algorithms: identify FIPS-approved algorithms and other cryptographic algorithms	- Unauthorized disclosure of data or undetected modification of data (intentional and accidental) during transmission or while in storage - denial of service - session capture - man-in-the-middle attack	FIPS-approved DES algorithm	NIST conformance tests	FIPS PUB 46-3: *DES*

Table 4. Risks and Cryptographic Technical and Assurance Features (Continued)

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Cryptographic Algorithms (continued)	(same as above)	Data Authentication Algorithm (DAA), data authentication code (DAC) computed in DES CBC or CFB mode (ANSI X9.9 MAC is computed in the same manner)	NIST conformance tests	FIPS PUB 113: *Data Authentication*
Cryptographic Algorithms (continued)	(same as above)	FIPS-approved cryptographic algorithms	NIST/CSE conformance testing	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*
Cryptographic Algorithms (continued)	(same as above)	Secure Hash Algorithm (SHA-1), message digest	NIST conformance tests	FIPS PUB 180-1: *Secure Hash Standard*
Cryptographic Algorithms (continued)	(same as above)	Digital Signature Algorithm (DSA) and rDSA, digital signature generation/verification, message digest, random/pseudorandom number generation, SHA-1	Algorithms for generating primes p and q, private key generation, pseudorandom number generator, NIST conformance tests	FIPS PUB 186: *Digital Signature Standard*

Table 4. Risks and Cryptographic Technical and Assurance Features (Continued)

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Cryptographic Algorithms (concluded)	(same as above)	digital signature algorithm (DSA, rDSA, ECDSA), digital signature, public key cryptography, message authentication algorithms, SHA-1	*No unique cryptographic assurance features required. Cryptographic requirements addressed in overall system/product requirements.*	NIST Special Pub. 800-15: *MISPC*
EMI/EMC: identify FCC conformance requirements	- Emanations	conformance to FCC requirements	*No unique cryptographic assurance features required. Cryptographic requirements addressed in overall system/product requirements.*	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*

Table 4. Risks and Cryptographic Technical and Assurance Features (Concluded)

Cryptographic Category	Risks/ Attacks	Technical Features/Functions	Assurance Features	Cryptographic Toolkit Reference
Self-Tests: identify power-up and conditional tests	- Module malfunction - Unauthorized disclosure of sensitive data	*No unique cryptographic technical features required. Cryptographic requirements addressed in overall system/product requirements.*	Documentation on error conditions and actions to clear the errors; - cryptographic algorithm test - critical functions test -statistical/continuous random number generator tests (monobit test, poker test, runs test, long run test) - pair-wise consistency test (for public and private keys) -software/firmware load test - manual key entry test	FIPS PUB 140-1: *Security Requirements for Cryptographic Modules*

To clarify how all this information fits together, Example 7 walks through the two tables and illustrates the process of defining requirements, identifying risks, and then selecting cryptographic methods that meet those requirements and mitigates the risks. Additional explanatory information is included in brackets.

Example 7. Using Tables 3 and 4

- *Risk/Attack*: Unauthorized disclosure of data or undetected modification of data (intentional and accidental) during transmission or while in storage [the risk was identified as the result of a risk assessment]

- *Functional Requirements:* Implement FIPS-approved security methods for data integrity assurance [the requirement addresses the risk]

- *Assurance Requirements:* Tests [the cryptographic algorithm must be tested to ensure that it is compliant with the FIPS standard. Also, tests may be executed to ensure the algorithm was implemented correctly.]

- *Cryptographic Category/Services:* Cryptographic algorithms [these methods provide features that track any change, e.g., modification, insertion, deletion, to security-relevant data]

- *Technical Features:* FIPS-approved DES algorithm [implementations of the algorithm which have been tested and validated by NIST are compliant with the standard]

- *Assurance Features:* NIST conformance tests [the tests are used to validate compliance with the standard]

- *Cryptographic Toolkit Reference:* FIPS PUB 46-3: DES [specific DES modes can be used to calculate a data authentication code that provides for data integrity]

- *Procurement Recommendations*: Federal agencies that use cryptography to protect sensitive information must use systems that have been FIPS PUB 140-1 validated.

7.3. Acquisition Phase

During the Acquisition Phase, one product/module is selected that meets the documented requirements. The product is then configured, implemented, and tested in the system. There are several types of testing that may be required, such as validation against FIPS PUB 140-1, system testing, and certification testing in support of system management authorization. Extensive testing of cryptographic controls is particularly important because of their role in ensuring the security of the overall system.

A second major task in the Acquisition Phase is to develop documentation for users and cryptographic officers to inform them of their responsibilities in maintaining a secure system. The Department of Energy project described in section 8.1.1 identifies some acquisition issues.

The next two sections specify some tips for implementing cryptography.

7.3.1. Implementation Approach

The security provided by a cryptographic system depends on the mathematical soundness of the algorithm, the length of the cryptographic keys, key management, and mode of operation. A weakness in any one of these components may result in a weakness or compromise to the security of the cryptographic system. A weakness may be introduced at any phase of the system life cycle.

During product design and development, it is the responsibility of the manufacturer of a cryptographic product to build a module that meets specified security requirements and conforms to a FIPS. However, conformance to a standard does NOT guarantee that a particular product is secure. To provide a level of assurance that the cryptographic product is secure, the product should be validated in the CMVP. The level of security in a cryptographic product/module must also be considered in the product selection phase. During this phase:

- Identify information resources and determine the sensitivity to and potential impact of losses. Determine security requirements based on risk assessment and applicable organizational security policies. Look at data sensitivity and the environment in which the data is placed. Consider threats to the data or application as a whole, and what level of risk is acceptable.

- Determine the acceptable safeguards for the system. Determine which cryptographic services provide an acceptable safeguard. Define those security features that are desirable for use and determine the appropriate security level from FIPS 140-1.

Finally, it is the responsibility of the integrator to configure and maintain the cryptographic module to ensure its secure operation because the use of a cryptographic product that conforms to a standard in an overall system does not guarantee the security of the cryptographic module or of the overall system. To summarize, the proper functioning of cryptography requires the secure design, implementation, and use of a validated cryptographic module.

There are many interdependencies among cryptography and other security controls, for example:

- *Physical Access Control*. Physical protection of a cryptographic module is required to prevent, or detect, physical replacement or modification of the cryptographic system and the keys within the system.

- *Logical Access Control*. Cryptographic modules may be embedded within a host system. With an embedded module, the hardware, operating system, and cryptographic software may be included within the cryptographic module boundary. Logical access control may provide a means of isolating the cryptographic software.

- *User Authentication.* Cryptographic authentication techniques may be used to provide stronger authentication of users. (Advanced authentication techniques are discussed in a later section.)

- *Assurance*. Assurance that a cryptographic module is properly and securely implemented is essential. NIST CMVP provides assurance that a module meets stated standards.

- *Integrity Controls*. Cryptography may provide methods that protect security-relevant software, including audit trails, from undetected modification.

The major rule is: **BUYER BEWARE**!! Example 8 illustrates how important it is to correctly implement and manage all of the security and cryptography controls to ensure that keys are not compromised.

Example 8. Implementation Problems

- Cryptographic algorithm may be strong, but the random number generator (RNG) may be weak

- RNG may be strong but the Key Management weak

- Key Management may be strong but the user authentication weak

- Authentication may be strong but the physical security weak

The following three rules guide the implementation of cryptography.

- *Determine what information must be protected using a cryptographic function.*

The implementor should be aware of the information that is being signed and encrypted. Fields containing sensitive data should be identified, and then a determination should be made of what cryptographic functions should be applied to those fields: integrity, authenticity, and/or confidentiality.

- *Protect data prior to signature generation/verification and encryption/decryption. Be careful of how data is handled during these processes!*

Implementers should be very careful about how data is handled before it is signed/verified (encrypted/decrypted). If data is stored in a central database and transferred to the computer only at the time the cryptographic function is performed, the data should be very carefully protected during transmission. If data is not carefully protected, an intruder could potentially alter data before a signature is generated, without the signer's knowledge. The data should be signed on the *signer's* machine, not in the central database.

- *Provide the capability for users to locally view all data that is being signed/encrypted.*

Users should be able to see all the data that is being signed, and it should be clearly marked for the signer. Also, users should know what is encrypted. It is not essential that all data being signed/encrypted should appear on one screen, but the user should at least be able to view all of the data before performing the cryptographic function.

7.4. Operations Phase

In the Operations Phase, the goal is to ensure the continued secure operation of the cryptographic methods. Two critical areas are training and the management of cryptographic components.

7.4.1. Training and Documentation

It is particularly important that all users be aware of their responsibilities, the procedures they must follow in ordinary as well as unusual circumstances, and who they should contact for assistance. These procedures should be standard among all sites in the system. Of special importance are the central sites, where security officers are responsible for equipment that might generate and manage keys for system users. If no documented set of procedures is followed, weaknesses may be introduced into the system such as transmitting data in clear text.

7.4.2. Life Cycle Management of Cryptographic Components

Maintenance of cryptographic components is critical to ensure the secure operation and availability of the module/product. For example, cryptographic keys that are never changed, even when disgruntled employees leave, are not secure. The following are maintenance areas that need to be considered during the cryptographic product life cycle:

1. *Hardware/firmware* (e.g., new capabilities, expansion of the system to accommodate more users, replacement of non-functional equipment, change of platforms, hardware component upgrades, etc.)

2. *Software maintenance/update* (e.g., new capabilities, fixing errors, improved performance, key replacement, etc.)

3. *Application maintenance* (e.g., changes in roles and responsibilities, remote updates, updating passwords, deleting users from access lists, etc.)

4. *Key maintenance* (e.g., key archiving, key destruction, key change, etc.)

5. *Maintenance personnel*. Who is allowed to perform maintenance? Do maintenance personnel require clearances, or do authorized users monitor maintenance activities? What must be removed from the system prior to maintenance? How is the correctness of the maintenance procedure ascertained?

Configuration management (CM) is needed for areas 1 and 2. CM ensures the management of system and security features through the control of changes made to a system's hardware, firmware, software, and documentation. The documentation may include user guidance, tests, test scripts and test documentation.

CHAPTER 8

8. PUTTING IT ALL TOGETHER - EXAMPLES

Included in this chapter are a few examples of projects that use cryptography. These projects illustrate how cryptography is used in the Federal government. In general, the projects have involved upgrading existing systems, while adding additional security and cryptographic capabilities. New systems that are being proposed and designed for Federal agencies focus on the PKI architecture and include digital signature, authentication, and encryption capabilities.

The descriptions in this chapter provide an overview of the projects and are not intended to include extensive detail. If more information is desired, contact the applicable agency.

8.1. Key Recovery Demonstration Project (KRDP)[44]

In May 1996, OMB released a white paper titled *Enabling Privacy, Commerce, Security, and Public Safety in the Global Information Infrastructure*. This paper stated that, "government and industry must work together to create a security management infrastructure and attendant products that incorporate robust cryptography without undermining national security and public safety." In support of this goal, a Key Recovery Demonstration Project (KRDP) was initiated in order to demonstrate the practicability of the recovery of keys that support data encryption in Federal government applications. Approximately ten Federal agencies participated in the pilot program in which different key recovery technologies were implemented, tested, and evaluated. Following are descriptions of two of these projects.

8.1.1. Department of Energy: EZ_ERA32 and the KRDP[45]

The KRDP seeks to test key recovery as part of a security component of an electronic commerce initiative with grantee organizations. This initiative has been underway since 1992, when a feasibility study conducted by Federal Information Exchange (now doing business as RAMS) found that grantee institutions were willing to migrate toward electronic research administration. Based on the findings, the Department of Energy (DOE) awarded a cooperative agreement to RAMS in 1994 to work on the NewERA project with research institutions and Federal funding agencies to demonstrate electronic research administration across the Internet and to provide "one face to government."

Electronic research administration has a number of important benefits including:

[44] KRDP information may be found at the web site: www.gits-sec.treas.gov.
[45] The information in this section was extracted from a report prepared by the Department of Energy.

- More efficient and timely proposal submissions;
- Improved data management capabilities;
- Integrated data functions among related operations;
- Reduced rekeying of data and administrative time; and
- Substantial paper savings.

Research applications may contain sensitive data, may involve considerable amounts of money, and can contain valuable intellectual property. Therefore, investigators are very concerned about the security of electronic transmissions. Failure to address security will inhibit widespread adoption of Electronic Research Administration and EDI. The requirements for cryptography include:

- Confidentiality: A confidentiality service to prevent the disclosure of information to unauthorized parties by means of strong encryption.

- Integrity: Providing cryptographic based integrity by means of message hashing to prevent undetected and unauthorized modifications of the information.

- Authentication and Non-repudiation: A user authentication service to provide verification of the sending identity by means of digital signature and the use of signature certificates.

- Key Recovery: Providing a means of retrieving a session key for the purposes of re-acquiring plaintext information from the ciphertext files in the event that the original decrypting key is no longer available.

The KRDP Pilot Project has provided the means to test the capability of emerging security technologies to meet the security requirements of electronic data interchange (EDI) over the Internet that is the heart of NewERA.

One benefit of the KRDP that has already been realized is the development of EZ_ERA32, a prototype security product by RAMS. In the absence of a commercial product that provided the complete set of security functionality, EZ_ERA32 was developed to tie together the different commercially available components. RAMS was responsible for seamlessly integrating the services and software of the vendor partners into the prototype software. EZ_ERA32 allowed the extraction of grant application data from user productivity tools such as Microsoft Excel and Word. The extracted data was then processed to generate a standard transaction set that was encrypted.

The software also provided functionality for public-private signature keys, public-private exchange keys, and Certificate Signing Request generation as well as other configuration utilities (email configuration, establishment of trading partner relationships and exchange key storage, etc.) that allowed the exchange of

secure EDI transactions to occur.

Partnering Vendors of Commercial Software and Services: Wherever possible, COTS products and services were utilized for the ERA project. The participating companies cooperated to provide the various security elements as described below.

Certificate Authority (CA): VeriSign provided CA services. A Web interface developed by VeriSign provided a site into which a user would paste a CSR generated by the EZ_ERA32 software. An EDI Certificate containing the public signature key was signed using the RSA algorithm and returned to the trading partner as a standard PKCS-7 file where it was stored by the EZ_ERA32 application. This certificate was then included within the encrypted package and sent to a recipient so the recipient could verify the identity of the sender and the integrity of the received EDI transaction

Key Recovery Agent (KRA): Requests for plaintext retrieval, whether from individual registered users or authorized law enforcement officials, must satisfy the procedures and rules of the designated KRA before a retrieval can be accomplished. Only authorized personnel can request a file recovery. All communications between the client and the Key Recovery Center (KRC) are encrypted. The KRC does not store (escrow) session keys, user keys, or user message files. Only an encrypted session key is returned to the client software where it is used to retrieve the plaintext data.

TIS Labs CryptALLProduct: CryptALL provides interfaces, dynamic link libraries, utilities, and sample code which can be utilized by application software to integrate the encryption and recovery capability into their products. The Client software runs on the application machine and also provides an interface for configuration and registration with a KRC. The software includes the cryptographic service provider module used for key generation and by the encryption/decryption processes of application software.

The System Architecture diagram in Figure 10 illustrates the relationship of EZ_ERA32 and the COTS services and software.

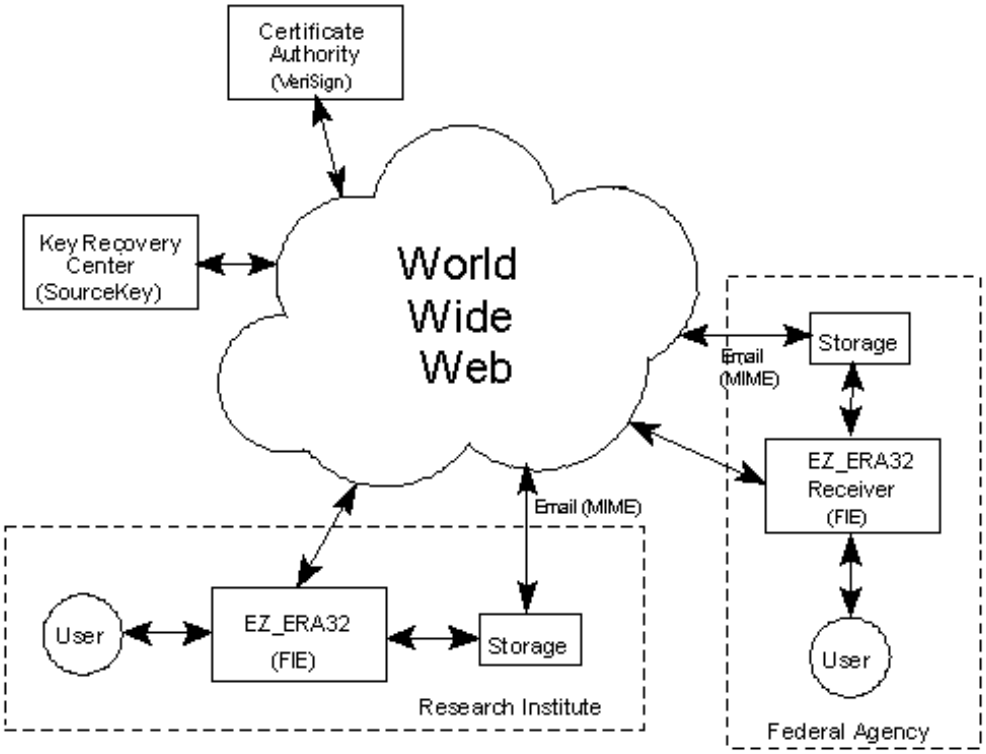


Figure 10. System Architecture

Findings:

1. Implementation of security functionality requires a whole new software setup and configuration. Many pieces must be set up (key generation, exchange partner agreements, obtaining a certificate, registration with a KRA, etc.) to create a complete environment. A basic understanding of how each of the security features works is necessary to get a system operational.

2. Participants indicated that they found the installation and configuration moderately difficult to perform. Since the system must be maintained over time, basic training for operations personnel is highly desirable. Educated and qualified support for operations personnel who must perform these tasks is critical.

3. It takes time to install and set up security features since a number of different features must be configured (e.g., keys, certificates, trading partner relationships, key recovery center certificate and accounts). Operations personnel must be provided with sufficient time to perform these tasks.

4. Interoperability problems still remain.

5. Once installed and correctly configured, the security functionality was fairly invisible to the end-users. Encrypted messages were successfully sent and acknowledgments received. If the KRC was available, those who attempted to perform recoveries were able to do so.

8.1.2. U.S. Electronic Grants[46]

The U. S. Department of Transportation (DOT) developed the U.S. Electronic Grants System (USEGS). The production system, currently being tested, will permit grant customers to exchange required data and files with Federal agency databases (Extranet) through data screens running in a Web browser. To facilitate KRDP testing, DOT designed a special version of the grants system called the Secure Electronic Grants System (Secure EGS). The project involved developing a security approach and technology, extensive testing, evaluating the results, and identifying remaining tasks.

System security features were developed to provide authentication, confidentiality, integrity, non-repudiation and key recovery. These features were provided through a PKI based on digital certificates that bound users to their public and private keys. The keys were used to enable digital signature and encryption capabilities based on the RSA algorithm. The certificates and keys were stored on smartcards. Certificates were issued by a contracted CA and managed using a Lightweight Directory Access Protocol (LDAP) server.

Test and Results: DOT and its partners successfully tested signing, encrypting, validating, and decrypting grant documents sent over the Internet. These functions were tested through a Java user interface. Simulated recovery of lost data ("key recovery") was also tested and demonstrated live at the KRDP conference in November 1997. These tests also highlighted areas for improvement including easier user system set-up, a friendlier interface, improved CA interoperability, a comprehensive registration interface, and more specific policies and procedures.

Remaining Tasks: The KRDP test proved that the technology currently available for securing electronic grant transactions across the Internet works as anticipated. However, the test also indicated that significant policy, procedural, technical, and operational issues must be resolved before a Secure EGS based on PKI can be implemented. The areas which need to be addressed include security hardware and software set-up for users, the signing/encryption interface, the registration process and interface, CA interoperability and cross-certification, and security policy and procedures.

[46] The information in this section was extracted from a report prepared by the Department of Transportation.

Accomplishments: The KRDP project successfully demonstrated the capability to conduct secure electronic grant transactions over the Internet through a Java applet running in a Web browser. Key technology accomplishments include the following:

Information Broker Based Secure Sockets Layer (SSL) - developed and implemented SSL in an Information Broker system. SSL provides presentation layer confidentiality for all objects (data, documents, spreadsheets, etc.) and transactions. Information Broker middleware provides object communications services including routing, queuing and filtering.

SSL Client Authentication - developed SSL client authentication capabilities. Access to system objects (Information Broker events) is controlled by the identity and attributes contained in a digital certificate. This approach requires several enhancements before it can be fully implemented.

Client Authentication - implemented client authentication using ID/password.

PKI - tested and demonstrated application layer PKI (digital signature and encryption) technology using RSA to encrypt a DES session key. Encryption provides confidentiality for sensitive grant objects with access limited to selected recipients. Digital signature provides authentication, object integrity, and non-repudiation.

Multiple Certificate Directories - tested and demonstrated access to multiple LDAP directories using Secure EGS so that Federal agency and customer certificates could be used in transactions.

Tokens - tested and demonstrated PKI using tokens (DataKey smartcards and readers).

Java - developed, tested and demonstrated the first Java applet/smartcard interface.

Secure Object Transfer - tested and demonstrated the secure transfer of objects between clients and a Federal database. This capability enables grant customers to select files from their PC's file system, send them to the Federal database, and retrieve them from that database for review and modification.

User Interface - developed and tested a graphical user interface (GUI) designed to enable users to conduct secure object transfer activities along with digital signature and encryption functions.

Key Recovery (KR) - tested and demonstrated KR using key encapsulation. This approach streamlines the KR business process by eliminating the need for key administration and escrow.

KRDP Test Results: In addition to developing and demonstrating technology capabilities, the project involved testing the capabilities with grant customers

(States of Washington and Utah, Cornell University) and other Federal agencies (the National Technical Information Service (NTIS), NIST). The results indicate that the technology works as anticipated in a live transaction environment. The tests identified the following issues:

Organizational Registration - DOT established trading partner and trust relationships during the test with its customers through informal site visits. These relationships enabled DOT to trust the certificates issued by the customer's CA. However, an approach based on informal site visits is not feasible for widespread system implementation.

Individual Registration/Access Control - individual registration was also an informal process conducted by each customer organization. Based on phone conversations, certificates were provided to test participants by the Federal CA or a state designated CA (in Utah). System access rights were provided by DOT based on phone conversations and personal knowledge of the customers. Again, these approaches to registration and assigning access control are not feasible for widespread system implementation.

PKI Authentication/Single Sign-on - ID and password authentication cannot provide adequate security or single sign-on capabilities. PKI/SSL client authentication capabilities were developed during the project to provide strong authentication and single sign-on features. However, significant work needs to be done to implement and test these capabilities.

Key Recovery - the tested approach to "key recovery" involved the DOT KRA (NTIS) decrypting objects and providing the clear text back to DOT. Unfortunately, this approach permitted the KRA access to confidential agency information. In practice, the KRA should only decrypt the session key wrapper and pass the object back to the Federal agency for recovery of clear text.

Key Recovery Procedures - as with registration, the process used to request, authorize and recover information was based on informal relationships, and lacked specific procedures. This approach is inadequate for a secure KR operation.

Technology Implementation - although the technology worked well when implemented in Federal test systems, several issues arose when it was implemented at customer sites. Most of these issues are related to the immaturity of the technology, but they must be resolved for the system to be successful.

Remaining Tasks: The following tasks need to be completed in order to provide a truly Secure U.S. EGS based on PKI technology:

Organizational Registration - registration, including the establishment of trading partner and trust agreements, must be defined and tested.

Individual Registration/Access Control - individual registration and assignment of access control rights will involve electronic interaction between users and their CAs and Federal agencies. User registration policies and procedures must be defined and tested.

PKI Authentication/Single Sign-on - Significant work remains to be done to integrate and test the smartcard and SSL interface. This work is necessary to provide PKI authentication and single sign-on.

User Interface - a new GUI interface for object management, digital signature and encryption must be designed, developed and tested.

CA Interoperability - interoperability standards must be defined (most likely by an industry or standards group) and tested. Federal agencies should only procure and establish electronic trading partner agreements with compliant CAs and customer organizations.

Multiple Step KR - this approach to key recovery must be developed and tested. The multiple step approach will limit KRA access to the session key wrapper and not permit access to confidential agency information.

KR Procedures - KR procedures, including the multiple step approach, must be defined, documented, automated and tested.

Customer Testing - as we discovered in the KRDP project, testing by vendors and Federal agencies is necessary but not sufficient. Extensive customer testing, revision of system components, and further testing is necessary to ensure a successful implementation of the Secure U.S. EGS.

8.2. Army Corps of Engineers

The Army Corps of Engineers has implemented electronic signatures in the Corps of Engineers Financial Management System (CEFMS). CEFMS is a Corps-wide computerized system that uses databases to manage financial data. CEFMS migrates numerous financial applications - including purchase requests, obligations, disbursements, and travel order certification to a completely electronic format. Corps employees can generate unique electronic signatures on electronic forms, and other CEFMS users can electronically verify the correctness of those signatures, eliminating the need to generate paper-based forms with handwritten signatures. The electronic signature must have all of the following attributes:

- Unique to the signer,
- Verifiable by a third party,
- Under the sole control of the signer, and
- Linked to the signed data in such a manner that if the data is changed, the signature is invalidated.

The subsystem of CEFMS that provides this signature generation and verification capability is the Electronic Signature System (ESS). The ESS uses the DES

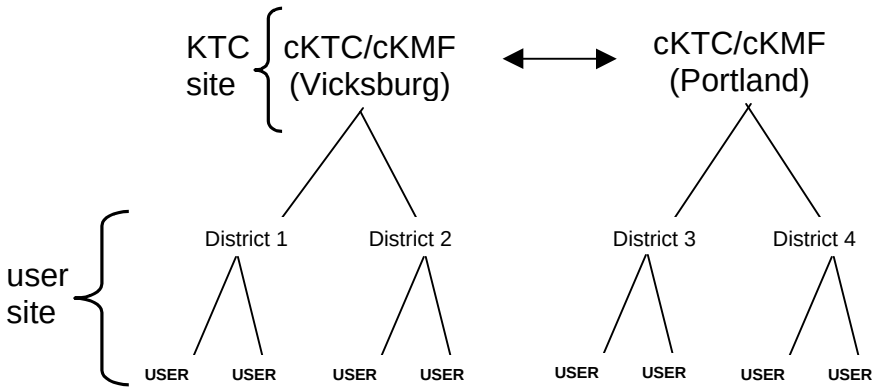
algorithm and key notarization techniques developed by NIST. The ESS was designed to be a modular system that could be plugged into various applications.

The ESS design is based on cryptography and is designed to implement split knowledge and dual control. Every user of the ESS must have a unique cryptographic key stored in a secure mode. The ESS allows for key management and token initialization, signature generation and verification, key translation, and message encryption and decryption. Key management and token initialization are functions of a Key Management Facility (KMF). Messages are signed and verified using MACs. Key translation, which allows one user to verify another's electronic signature, occurs at a Key Translation Center (KTC).

In the second phase, a software reference implementation was developed. This phase involved testing the cryptographic PC adapter used in the ESS. The cryptomodule incorporates the cryptographic service calls (CSCs) and application cryptographic commands (ACCs) used to perform the ESS functions. Testing was performed on the implementation of the ACCs and CSCs to ensure that they were implemented correctly. Testing was executed before the ESS went operational. By finding some cryptographic bugs in the adapter before thousands were distributed to ESS users, the dissemination of weak cryptography that could have led to incorrect signatures was prevented.

8.2.1. ESS Architecture

In general, the ESS has two sites that each contains a user key database, which mirrors the other site's database and software, that acts as a central KMF/central KTC (cKMF/cKTC[47]). A backup computer is stored in the immediate vicinity of the primary KTC machine, along with several cryptomodules that can be used if those in the primary computer fail. Figure 11 illustrated the ESS design.



[47] This will be referred to as the KTC, since both functions are centrally provided on a single machine at each KTC site.

Figure 11. ESS Architecture

Both KTCs maintain a complete user key database containing keys for all valid users of the ESS. Central Security Officers are responsible for using the KTC software to initialize tokens for all users of the ESS. Each KTC remains activated twenty-four hours a day, seven days a week.

The ESS has multiple user sites, which are connected to one of the main processing centers via the CEFMS Wide Area Network (WAN), depending on their location. District Security Officers (dSOs), Security Administrators (SAs), and Users may reside at the user sites. Their functions range from managing tokens to providing split knowledge and dual control at all sites. Each user site has a user information database that contains personnel information pertaining to users possessing tokens.

8.2.2. Key Management

cSOs and KTCs are the cornerstones of the ESS because they are needed to manage cryptographic keys for the entire ESS. Key management is an essential component of the CEFMS ESS. It provides the foundation necessary to securely generate, store, distribute and translate cryptographic keys within CEFMS. One of the fundamental principles for protecting keys is the practice of split knowledge and dual control.

8.2.2.1. Key Generation

cSOs are responsible for initializing all types of tokens for all users of the ESS. This includes cSO, dSO, SA, and User tokens. In CEFMS, tokens are used to securely store cryptographic keys that enable the tokens to be used to initialize cryptomodules, initiate KTCs, and allow users to access the ESS. It is important that the cSOs who are generating the tokens maintain split knowledge and dual control during token generation for all types of users.

8.2.2.2. Key Storage

Once keys are generated in the ESS, they must be stored securely, to prevent unauthorized persons from accessing and using them. This is particularly critical in a system like CEFMS because the secure key storage is necessary for controlling financial transactions. Secure storage is needed for both the data keys(KDs) and the key encrypting keys (*KKs). Keys are never removed from the module without first being encrypted by some value.

8.2.2.3. Key Destruction

Key destruction involves a variety of mechanisms, including the zeroizing of keys in a cryptomodule, the reinitialization of a token, and the removal and/or disabling of a key in the user key database.

8.2.3. Signature Generation and Verification

The signature generation process of the CEFMS ESS involves the calculation of a MAC. This MAC serves as an electronic signature when it is created using a notarized key. When an ESS user generates a signature on selected data, the key used to generate the MAC is stored in encrypted form by the CEFMS database. This key is then used in the signature verification process.

8.3. Treasury Electronic Certification System

The number one priority of the Department of the Treasury Financial Management Service (FMS) is payment operations and the Electronic Certification System (ECS) project is the most critical of the basic ongoing payment operations. The ECS Project's sole purpose is to aid in achieving the goal of ensuring a world-class delivery of all Federal government payments and associated information to their ultimate destination.

The ECS eliminated a paper-based, error-prone and time-consuming process of entering, certifying and making payments. The system also gives small agencies that cannot afford mainframe computers the capability to use the Service's Vendor Express and other Automated Clearing House (ACH) payment processes, as well as the Same-Day-Pay-Request (Fedwire) payment process. Formerly, only agencies that submitted magnetic tapes to the financial centers could use the ACH payment mechanism. Paper forms were required for Same Day Payments.

The ECS is the only method currently available for creating and processing electronic certifications. The ECS is operating in all six Regional Financial Centers and is used by over 500 agency end-points to prepare, certify, and submit to the FMS an average of over 30,000 payment schedules each month. Over 20,000 of these schedules, each month, are ones that contain both individual payment data and the certification (Type A Schedules). These agencies make over 75 million payments, valued at over 65 billion dollars, each month.

Additionally, the ECS, through its use by the Federal Emergency Management Administration (FEMA), has speeded responses to major disasters such as hurricanes, floods, earthquakes, and other national emergencies. The ECS is now being used worldwide in 38 countries.

8.3.1. Program History

In 1986, a small staff began with an idea, a new concept. The FMS was making payments using a very labor-intensive and time-consuming process that entailed requiring Federal Program Agencies (FPAs) to manually submit paper SF-1166 Voucher and Schedule of Payments (vouchers) to certify payment requests, and

in many cases to provide payment data. These vouchers were submitted to FMS' Regional Financial Centers by mail, express and special courier. At the Regional Financial Centers, vouchers containing payment data were scanned using Optical Character Recognition (OCR) equipment, which had a very high "read" error rate. This could mean the voucher had to be returned to the agency to be redone. The voucher would then repeat the paper-based process that is very prone to error and is time consuming. The ECS concept provided a much faster, more efficient, more economical, and more secure method for certifying and making payments. The ECS added electronic signatures to payment requests submitted to FMS by FPAs. Vouchers containing certifications for payment data submitted on magnetic tape were manually verified and used to certify the production of payments.

The ECS electronic signature certification operation began early in 1987, with the ECS Prototype. NIST and the General Accounting Office participated in the approval of the concept to ensure that the system satisfied the Federal requirements concerning voucher certification and payments. Initially, the ECS provided the facilities to submit only schedules for small volume vendor check payments. Subsequently, over the next 10 years, the ECS was improved and extended through continued development to provide features that support processing of all payment types processed by the FMS. The ECS is the only electronic system that is available for submission and certification of payments to the FMS. Additionally, it is the only means of timely certification for bulk payment files telecommunicated to FMS.

8.3.2. ECS Process

The ECS provides FPAs with the ability to create, certify and transmit two classes of payment schedules (Type A and Type B) to FMS using FMS ECS software and Message Authentication Security hardware. Type A schedules are those that contain both the payment data for individual payments as well as the required certification for them. Type A schedules can contain 1 to 60 payments per schedule (average of 7.8 payments per schedule). Type B schedules are those that are used to certify payment data provided to FMS in a bulk file (through either magnetic tape or telecommunication of a bulk file). Bulk files typically contain large numbers of payments (from hundreds to millions).

Agencies enter the schedule data on their microcomputers, a unique message authentication code (MAC)[48] is created using the cryptographic signature of the agency Certifying Officer and ECS Security Administrator. The MAC is attached to the data and transmitted to FMS' Regional Financial Center computer. At the Regional Financial Center, the MAC is recalculated, again using the

[48] The MAC is generated on the current data and compared with the previously generated MAC. If the two values are equal, the integrity (i.e., authenticity) of the data is verified.

cryptographic signatures, and is checked to see if the code is different from the one sent to the center. If there is no difference, the payments are made; otherwise, they are rejected.

The ECS electronic signature design uses DES, smartcard technology, a separation-of-duties concept, and requires a dual-role-process to provide a level of technology and security previously unemployed by government payment processes.

8.3.3. Future Plans: Windows-Based ECS (WECS)

The electronic signature certification and MAC for payment schedules are based upon cryptographic keys that are unique to each user of the system. The cryptographic keys form the basis of the ECS security and integrity. The current system of keys is based upon DES. The DES will not be supported in its current form by NIST in the near future. Therefore, the FMS has had to look at new technologies to provide this security and integrity function for systems where the use of cryptography is indicated.

At this time, the most appropriate solution identified involves the use of PKI and on-card cryptographic processors (smartcards that provide the necessary cryptographic functions within the card), in conjunction with telecommunications modems incorporating smartcard cryptographic functions to secure telephone connections used to transmit data. This solution is the same one that has been selected for the FMS Electronic Check (E-Check) project, and would strengthen the security and integrity of the ECS, while allowing for a common cryptographic platform and the cryptographic hardware to support any current or future microcomputer hardware and operating system.

The pilot E-Check system will implement cryptography based on the DSS and will include PKI cryptography. The CA, cryptographic components, and much of the cryptographic processing and support software would be usable in a redesigned ECS. Use of the E-Check CA and cryptographic components would reduce the cost and lead time for development of a new ECS. This would also have the advantage of employing a common FMS cryptographic platform.

For the proposed windows-based ECS (WECS), with the use of Windows NT (with its built-in C2 level security system) and client/server concepts, it will be possible to make the overall security and integrity of the system much more robust, while simplifying log-on procedures for users. Additional security and integrity will be provided by an architecture that provides communications encryption and authentication for all ECS electronic transmissions.

CHAPTER 9

9. WHAT'S NEXT?

Currently available cryptographic methods provide users with valuable services. With the constant change in computer technology, the increased internetworking of systems, and the significant advance in computing power, new and more powerful cryptographic methods will be needed. Following is a summary of the new initiatives that are being evaluated by NIST.

9.1. Advanced Encryption Standard (AES)

For interoperability and other purposes, NIST strongly desires to select a single block encryption algorithm to be specified in the AES with strength equal to or better than that of Triple DES and significantly improved efficiency.

It is intended that the AES will specify an unclassified, publicly disclosed encryption algorithm available royalty-free worldwide that is capable of protecting sensitive government information well into the next century. (There is a possibility that the AES will specify multiple algorithms.)

Encryption algorithms that have been submitted for consideration as the Advanced Encryption Algorithm (AEA) are being reviewed by NIST and the public on the basis of evaluation criteria. The selected AEA will be included in the FIPS for AES.

9.1.1. Minimum Acceptability Requirements

All of the candidate algorithms met the following minimum acceptability requirements:

1. The algorithm must implement symmetric (secret) key cryptography.

2. The algorithm must be a block cipher.

3. The candidate algorithm shall be capable of supporting *key-block combinations with sizes of* ***128-128, 192-128,*** *and* ***256-128*** *bits*. A submitted algorithm may support other key-block sizes and combinations, and such features will be taken into consideration during analysis and evaluation.

9.1.2. Evaluation Criteria

The following evaluation criteria are being used to review candidate algorithms.

Security (i.e., the effort to cryptanalyze).

Cost (licensing requirements, computational efficiency, memory requirements).

Algorithm and Implementation Characteristics (flexibility, hardware and software suitability, simplicity).

9.1.3. AES Finalists

On August 28, 1998, NIST announced a group of fifteen AES candidate algorithms. Members of the cryptographic community from around the world had submitted the algorithms. Using the public comments and analyses conducted by the global cryptographic community, NIST selected five algorithms from the fifteen. The AES finalist candidate algorithms are MARS, RC6, Rijndael, Serpent, and Twofish. These finalist algorithms will receive further analysis during a second, more in-depth review period prior to the selection of the final algorithm(s) for the AES FIPS. Following the close of the Round 2 public analysis period, NIST intends to study all available information and propose the AES, which will incorporate one or more AES algorithms selected from the finalists. If all steps of the AES development process proceed as planned, it is anticipated that the standard will be completed by the summer of 2001. Upon publication of the standard, NIST intends to have a validation testing program in place to test the algorithm(s).

9.2. Key Agreement or Exchange

Cryptographic services depend on the secure generation and distribution of keys (public and private). Since there is no existing FIPS in this area, a standard is needed for the design and implementation of Federal key agreement and exchange systems. NIST has solicited public comments on potential technologies that could be considered for a future standard for public key-based cryptographic key agreement and exchange. NIST has specifically asked for comments on RSA, Elliptic Curve, and Diffie-Hellman technologies. More than one algorithm could be specified, consistent with sound security practices to give Federal organizations more flexibility in using cryptographic systems.

9.3. Key Recovery

NIST is exploring the use of key recovery technology through a broad agency agreement for several agency pilots and with the help of a special advisory committee. A Key Recovery Demonstration Project has been established involving several government agencies to demonstrate the practicality of techniques to recover keys used in data encryption and to identify, test, and evaluate different key recovery products and services. This effort supports an

Administration white paper entitled *Enabling Privacy, Commerce, Security, and Public Safety in the Global Information Infrastructure*.

9.4. Technical Advisory Committee

The Technical Advisory Committee to Develop a Federal Information Processing Standard for the Federal Key Management Infrastructure (TACDFIPSFKMI) was established by the Department of Commerce in July 1996. The Committee will advise the Secretary of Commerce on the development of a draft FIPS for the Federal Key Management Infrastructure. The purpose of the standard is to specify requirements for key recovery components, and to enable the validation of components claiming conformance. The standard encompasses the security (from an implementation, managerial and operation perspective) and the availability of key recovery components, as well as defining interoperability requirements.

Key recovery is motivated by three primary scenarios:

1. Recovery of stored data on behalf of an organization (or individual) e.g., in response to accidental loss of keys;

2. Recovery of stored or communicated data on behalf of an organization (e.g., for the purposes of monitoring or auditing activities); and

3. Recovery of communicated or stored data by lawfully authorized authorities.

The first scenario supports the ability to regain access to data that would otherwise be lost. The second scenario encompasses internal investigation authorized by an organization. The final scenario encompasses data acquired under the authorization of court orders for wiretaps, search and seizure orders, civil suit subpoenas, etc.

A Key Recovery System (KRS) manages cryptographic keys in support of data recovery when normal key access mechanisms fail. These systems must be carefully designed so that plaintext may be recovered in a timely manner, and so that only authorized recoveries are permitted. Therefore, security is a critical factor in any KRS design.

9.5. FIPS 140-2

FIPS 140-1, first published in 1994, specified that it be reviewed within five years. In 1998, NIST solicited public comments on reaffirming the standard. The comments received by NIST supported maintaining the standard. The comments also supported updating the standard due to advances in technology. The proposed revision (FIPS 140-2) is currently being prepared.

APPENDIX A

ACRONYMS

ACC	Application Cryptographic Commands
ACH	Automated Clearing House
AEA	Advanced Encryption Algorithm
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
BCA	Bridge Certification Authority
CA	Certification Authority
CAPI	Cryptographic Application Programming Interface
CBC	Cipher Block Chaining Mode
CC	Common Criteria
CEFMS	Corps of Engineers Financial Management System
CFB	Cipher Feedback Mode
cKMF	central Key Management Facility
CKMS	Centralized Key Management System
cKTC	central Key Translation Center
CM	Configuration Management
CMT	Cryptographic Module Testing
CMV	Cryptographic Module Validation
CMVP	Cryptographic Module Validation Program
COTS	Commercial-Off-The-Shelf
CPS	Certification Practice Statement
CRADA	Cooperative Research and Development Agreement
CRL	Certificate Revocation List
CSA	Computer Security Act
CSC	Cryptographic Service Call
CSE	Communications Security Establishment
cSO	central Security Officer
CSPs	Critical Security Parameters
CSR	Certificate Status Responder
DAA	Data Authentication Algorithm
DAC	Data Authentication Code
DEA	Data Encryption Algorithm
DES	Data Encryption Standard
DID	Data Item Identifier
DII	Defense Information Infrastructure
DISN	Defense Information Systems Network
DOE	Department of Energy

DOT	Department of Transportation
DPMA	Domain Policy Management Authority
DSA	Digital Signature Algorithm
dSO	district Security Officer
DSS	Digital Signature Standard
DSSV	Digital Signature Storage and Verification
DTR	Derived Test Requirement
EAL	Evaluation Assurance Level
EC	Elliptic Curve
ECB	Electronic Codebook Mode
ECDSA	Elliptic Curve Digital Signature Algorithm
ECS	Electronic Check System
EDI	Electronic Data Interchange
EFP	Environmental Failure Protection
EFT	Environmental Failure Testing
EGS	Electronic Grants System
EMI/EMC	Electromagnetic Interference/Electromagnetic Compatibility
ESS	Electronic Signature System
FCC	Federal Communications Commission
FEMA	Federal Emergency Management Administration
FIPS	Federal Information Processing Standard
FIPS PUB	Federal Information Processing Standard Publication
FMS	Financial Management Service
FPA	Federal Program Agencies
FPMA	Federal Policy Management Authority
GAO	General Accounting Office
GCS-API	Generic Cryptographic Services - API
GUI	Graphical User Interface
I&A	Identification and Authentication
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IT	Information Technology
ISO	International Organization for Standardization
IV	Initialization Vector
IV&V	Independent Verification and Validation
*KK	Key Encrypting Key
$*KK_{cms}$	Central Master Storage Key Encrypting Key
KAT	Known Answer Test
KD_{com}	Data Communications Data Key
KEA	Key Exchange Algorithm
KMF	Key Management Facility

KR	Key Recovery
KRA	Key Recovery Agent
KRC	Key Recovery Center
KRDP	Key Recovery Demonstration Project
KRS	Key Recovery System
KTC	Key Translation Center
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MCT	Monte Carlo Tests
MID	Message Identifier
MISPC	Minimum Interoperability Specification for PKI Components
MCT	Monte Carlo Test
NIAP	National Information Assurance Partnership
NII	National Information Infrastructure
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NTIS	National Technical Information Service
NVLAP	National Voluntary Laboratory Accreditation Program
OCP	Optical Character Recognition
OFB	Output Feedback Mode
OMB	Office of Management and Budget
PC	Personal Computer
PCA	Policy Certification Authority
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POC	Point-of-Contact
PP	Protection Profile
PRNG	PseudoRandom Number Generator
RA	Registration Authority
RFP	Request for Proposal
RNG	Random Number Generator
SA	Security Administrator
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	Special Publication
SSL	Secure Socket Layer
SUT	System Under Test
TCBC	TDEA Cipher Block Chaining Mode

TCBC-I	TDEA Cipher Block Chaining Mode - Interleaved
TDEA	Triple DEA
TECB	TDEA Electronic Codebook Mode
TCFB	TDEA Cipher Feedback Mode
TCFB-P	TDEA Cipher Feedback Mode - Pipelined
TOE	Target of Evaluation
TOFB	TDEA Output Feedback Mode
TOFB-I	TDEA Output Feedback Mode - Interleaved
TSF	TOE Security Functions
TSP	TOE Security Policy
U.S.	United States
USEGS	U.S. Electronic Grants System
WAN	Wide Area Network
WECS	Windows-based ECS
WWW	World Wide Web

APPENDIX B

TERMS AND DEFINITIONS

This section includes terms and definitions that are commonly used in or associated with cryptography. In general, the definitions are drawn from FIPS PUBs, related documents, and other standards. The source of each definition is included with the definition and the full references are included in Appendix C. The source is listed at the end of the definition in square brackets [ ]. Some terms include more than one definition - multiple definitions are included to illustrate the variations in the use of a term or to provide a more detailed definition.

Application Programming Interface (API): The interface between the application software and the application platform, across which all services are provided. The API is primarily in support of application portability, but system and application interoperability is also supported by a communication API. [X/Open Preliminary Specification]

asset: Information resources that support an organization's mission. [NIST Special Publication (SP) 800-12]

Information or resources to be protected by the countermeasures of a Target of Evaluation (TOE). [Common Criteria]

asymmetric algorithm: See public-key algorithm.

authentication: The broadest definition of authentication within computing systems encompasses identity verification, message origin authentication, and message content authentication. [FIPS PUB 190]

Authenticity refers to validating the source of a message i.e., that it was transmitted by a properly identified sender and is not a replay of a previously transmitted message. [NIST SP 800-2]

automated key distribution: The distribution of cryptographic keys, usually in encrypted form, using electronic means, such as a computer network. [FIPS PUB 140-1]

binding: An acknowledgment by a trusted third party that associates an entity's identity with its public key. This may take place through (1) a certification authority's generation of a public key certificate, (2) a security officer's verification of an entity's credentials and placement of the entity's public key and identifier in a secure database, or (3) an analogous method. [FIPS PUB 196]

certificate (or public key certificate): A digitally signed data structure defined in the X.509 standard that binds the identity of a certificate holder (or subject) to a public key. [MISPC]

certificate revocation list (CRL): a list of revoked but unexpired certificates issued by a CA. [MISPC]

certification authority (CA): A trusted entity that issues certificates to end entities and other CAs. CAs issue CRLs periodically, and post certificates and CRLs to a repository. [MISPC]

certification path: An ordered sequence of certificates, leading from a certificate whose public key is known by a client, to a certificate whose public key is to be validated by the client. [MISPC]

ciphertext: Encrypted (enciphered) data. [FIPS PUB 46-3]

claimant: An entity which is or represents a principal for the purposes of authentication, together with the functions involved in an authentication exchange on behalf of that entity. A claimant acting on behalf of a principal must include the functions necessary for engaging in an authentication exchange (e.g., a smartcard (claimant) can act on behalf of a human user (principal)). [FIPS PUB 196]

compromise: The unauthorized disclosure, modification, substitution or use of sensitive data (including plaintext cryptographic keys and other critical security parameters). [FIPS PUB 190]

confidentiality: The property that sensitive information is not disclosed to unauthorized individuals, entities or processes. [FIPS PUB 140-1]

countermeasure: Any action, device, procedure, technique, or other measure that reduces a system's vulnerability to a threat. [NIST SP 800-12]

critical security parameters: Security-related information (e.g., cryptographic keys, authentication data such as passwords and PINs) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module. [FIPS PUB 140-1]

cryptographic hash function: A (mathematical) function that maps values from a large (possibly very large) domain into a smaller range. The function satisfies the following properties:

1. (One-way) It is computationally infeasible to find any input which maps to any pre-specified output;

2. (Collision free) It is computationally infeasible to find any two distinct inputs that map to the same output. [ANSI X9.42]

cryptographic key: A parameter used in conjunction with a cryptographic algorithm that determines:
- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a data authentication code (DAC) computed from data.

[FIPS PUB 140-1]

cryptographic key component: A parameter which is combined (e.g., via a bit-wise exclusive-OR operation) with one or more other identically sized key component(s) to form a plaintext cryptographic key. [FIPS PUB 140-1]

cryptographic module: The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module. [FIPS PUB 140-1]

cryptography: The discipline which embodies principles, means and methods for the transformation of data to hide its information content, prevent its undetected modification, prevent its unauthorized use or a combination thereof. [ANSI X9.31]

Cryptography deals with the transformation of ordinary text (plaintext) into coded form (ciphertext) by encryption and transformation of ciphertext into plaintext by decryption. [NIST SP 800-2]

cryptoperiod: The time span during which a specific key is authorized for use or in which the keys for a given system may remain in effect. [ANSI X9.31]

Data Authentication Code (DAC): Applying the Data Authentication Algorithm (DAA) to data generates a DAC. The DAC is a mathematical function of both the data and a cryptographic key. When the integrity of the data is to be verified, the DAC is generated on the current data and compared with the previously generated DAC. If the two values are equal, the integrity (i.e., authenticity) of the data is verified. [FIPS PUB 113]

A DAC is also known as a Message Authentication Code (MAC) in ANSI standards. [FIPS PUB 140-1]

data integrity: The state that exists when computerized data is the same as that in the source documents and has not been exposed to accidental or malicious alteration or destruction. [FIPS PUB 39]

data key: A cryptographic key which is used to cryptographically process data (e.g., encrypt, decrypt, authenticate). [FIPS PUB 140-1]

decryption: The process of changing ciphertext into plaintext. [FIPS PUB 81]

DES: The symmetric encryption algorithm defined by the Data Encryption Standard (FIPS PUB 46-3). [MISPC]

DES MAC: An algorithm for generating a message authentication code (MAC) using the symmetric encryption algorithm DES. [MISPC]

digital signature: The result of a cryptographic transformation of data which, when properly implemented, provides the services of:

1. origin authentication,
2. data integrity, and
3. signer non-repudiation. [ANSI X9.31]

The digital signature is computed using a set of rules (e.g., the Digital Signature Algorithm (DSA)) and a set of parameters such that the identity of the signatory and integrity of the data can be verified. [FIPS PUB 186-2]

A data unit that allows a recipient of a message to verify the identity of the signatory and integrity of the message. [MISPC]

A nonforgeable transformation of data that allows the proof of the source (with non-repudiation) and the verification of the integrity of that data. [FIPS PUB 196]

Digital Signature Algorithm (DSA): The DSA is used by a *signatory* to generate a digital signature on data and by a *verifier* to verify the authenticity of the signature. [FIPS PUB 186-2]

dual control: A process of utilizing two or more separate entities (usually persons) operating in concert, to protect sensitive functions or information. [ANSI X9.17]

electronic signature: A method of signing an electronic message that --

(A) Identifies and authenticates a particular person as the source of the electronic message; and

(B) Indicates such person's approval of the information contained in the electronic message. [GPEA]

Elliptic Curve Digital Signature Algorithm (ECDSA): A digital signature algorithm that is an analog of DSA using elliptic curve mathematics and specified in ANSI standard X9.62. [MISPC]

encrypted key (ciphertext key): A cryptographic key that has been encrypted with a key encrypting key, a PIN or a password to disguise the value of the underlying plaintext key. [FIPS PUB 140-1]

encryption: The process of changing plaintext into ciphertext for the purpose of security or privacy. [FIPS PUB 81]

entity: Any participant in an authentication exchange such a participant may be human or non-human, and may take the role of a claimant and/or verifier. [FIPS PUB 196]

ephemeral data: In ANSI X9.42, ephemeral data is data specific to a particular execution of a cryptographic scheme. Ephemeral data includes ephemeral private/public keys and may include ephemeral-key domain parameters. [ANSI X9.42]

error detection code: A code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data. [FIPS PUB 140-1]

FIPS approved security method: A security method (e.g., cryptographic algorithm, random number generator) that is either a) specified in a FIPS, or b) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS. [FIPS PUB 196]

hash: The SHA-1 produces a 160-bit condensed representation of the message called a message digest. The message digest is used during generation of a signature for the message. The SHA-1 is also used to compute a message digest for the received version of the message during the process of verifying the signature. Any change to the message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify. [FIPS PUB180-1]

hash code: The string of bits that is the output of a hash function. [MISPC]

initialization vector (IV): A vector used in defining the starting point of an encryption process within a cryptographic algorithm. [FIPS PUB 140-1]

integrity: The property that sensitive data has not been modified or deleted in an unauthorized and undetected manner. [FIPS PUB 140-1]

Integrity refers to assurance that a message was not modified accidentally or deliberately in transit, by replacement, insertion or deletion. [NIST SP 800-2]

key: See cryptographic key

key encrypting key: A cryptographic key that is used for the encryption or decryption of other keys. [FIPS PUB 140-1]

key management: The activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs, counters) during the entire life cycle of the keys, including the generation, storage, distribution, entry and use, deletion or destruction, and archiving. [FIPS PUB 140-1]

The generation, storage, secure distribution and application of keying material in accordance with a security policy that prevents its modification, unauthorized use, or a combination thereof. [ANSI X9.42]

keying material: The data (e.g., keys and IVs) necessary to establish and maintain cryptographic keying relationships. [ANSI X9.17]

message: The data to be signed. [ANSI X9.31]

message identifier (MID): A field that may be used to identify a message. Typically, this field is a sequence number. [ANSI X9.31]

message authentication code (MAC): A data authenticator generated from the message, usually through cryptographic techniques. In general, a cryptographic key is also required as input. [MISPC]

Note: the MAC as specified in ANSI X9.9 is computed in the same manner as the DAC specified in FIPS PUB 113.

message digest: The fixed size result of hashing a message. [MISPC]

non-repudiation: This service provides proof of the integrity and origin of data that can be verified by a third party. [ANSI X9.31]

Non-repudiation of origin is protection against a sender of a message later denying transmission. [NIST SP 800-2]

password: A string of characters used to authenticate an identity or to verify access authorization. [FIPS PUB 140-1]

Personal Identification Number (PIN): A 4 to 12 character alphanumeric code or password used to authenticate an identity, commonly used in banking applications. [FIPS PUB 140-1]

plaintext: Unencrypted (unenciphered) data. [FIPS PUB 81]

principal: An entity whose identity can be authenticated. [FIPS PUB 196]

private key: A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and not made public. [FIPS PUB 140-1]

In an asymmetric (public) key cryptosystem, the key of an entity's key pair that is known only by that entity. A private key may be used to:

(1) Compute the corresponding public key,

(2) Make a digital signature that may be verified by the corresponding public signature,

(3) Decrypt data encrypted by the corresponding public key, or

(4) Compute a piece of common shared secret information together with other information.

[ANSI X9.42]

The private key is used to generate a digital signature. This key is mathematically linked with a corresponding public key. [FIPS PUB 196]

public key: A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public. [FIPS PUB 140-1]

In an asymmetric (public) key cryptosystem that key of an entity's key pair that may be publicly known. A public key may be used to:

(1) Verify a digital signature that is signed by the corresponding private key,

(2) Encrypt data that may be decrypted by the corresponding private key,

(3) Compute a piece of shared information by other parties.

[ANSI X9.42]

The public key is used to verify a digital signature. This key is mathematically linked with a corresponding private key. [FIPS PUB 196]

public key certificate (certificate): A set of data that unambiguously identifies an entity, contains the entity's public key, and is digitally signed by a trusted third party (certification authority). [FIPS PUB 196]

public key cryptography (reversible): Reversible public key cryptography is an asymmetric cryptographic algorithm where data encrypted using the public key can only be decrypted using the private key and conversely, data encrypted using the private key can only be decrypted using the public key. [ANSI X9.31]

public key (asymmetric) cryptographic algorithm: A cryptographic algorithm that uses two related keys, a public key and a private key; the two keys have the property that, given the public key, it is computationally infeasible to derive the private key. [FIPS PUB 140-1]

These algorithms are referred to as "asymmetric" because they rely on two different keys to perform cryptographic processing of data. [FIPS PUB 190]

public key infrastructure (PKI): An architecture which is used to bind public keys to entities, enable other entities to verify public key bindings, revoke such bindings, and provide other services critical to managing public keys. [FIPS PUB 196]

role: A predefined set of rules establishing the allowed interactions between a user and the TOE. [Common Criteria]

RSA: A public-key signature algorithm specified in the *RSA Encryption Standard*, Version 1.4, RSA Data Security, Inc., 3 June 1991.

secrecy: Secrecy refers to denial of access to information by unauthorized individuals. [NIST SP 800-2]

secret key: A cryptographic key used with a secret key cryptographic algorithm, uniquely associated with one or more entities, and which shall not be made public. The use of the term "secret" in this context does not imply a classification level, rather the term implies the need to protect the key from disclosure or substitution. [FIPS PUB 140-1]

secret key (symmetric) cryptographic algorithm: A cryptographic algorithm that uses a single, secret key for both encryption and decryption. [FIPS PUB 140-1]

secure hash algorithm: An algorithm that can generate a condensed message representation of a message or a data file, called a message digest. **[**FIPS PUB 190]

session key: The cryptographic key used by a device [module] to encrypt and decrypt data during a session. [FIPS PUB 185]

signature generation: Makes use of a private key to generate a digital signature. Only the possessor of the user's private key can perform signature generation. [FIPS PUB 186-2]

signature verification: Makes use of a public key that corresponds to, but is not the same as, the private key. Anyone can verify the signature of a user by employing that user's public key. [FIPS PUB 186-2]

split knowledge: A condition under which two or more entities separately have key components which individually convey no knowledge of the plaintext key which will be produced when the key components are combined in the cryptographic module. [FIPS PUB 140-1]

Note: in more general terms, split knowledge applies to splitting knowledge of the secret *S* among two or more individuals.

static data: Data that is relatively long-lived. In ANSI X9.42, static data is data common to a number of executions of a cryptographic scheme. Static data includes static-key domain parameters and static private/public keys. [ANSI X9.42]

symmetric key: A cryptographic key that is used in symmetric cryptographic algorithms. The same symmetric key that is used for encryption is also used for decryption. [ANSI X9.42]

threat: An entity or event with the potential to harm a system. [NIST SP 800-12]

trusted path: A mechanism by which a person or process can communicate directly with a cryptographic module and which can only be activated by the person, process or module, and cannot be imitated by untrusted software within the module. [FIPS PUB 140-1]

A means by which a user and a Target of Evaluation Security Functions (TSF) can communicate with necessary confidence to support the Target of Evaluation Security Policy (TSP). [Common Criteria]

verifier: An entity that is or represents the entity requiring an authenticated identity. A verifier includes the functions necessary for engaging in authentication exchanges. [FIPS PUB 196]

vulnerability: A condition or weakness in (or absence of) security procedures, technical controls, physical controls, or other controls that could be exploited by a threat. [NIST SP 800-12]

zeroization/zeroisation: A method of erasing electronically stored data by altering the contents of the data storage so as to prevent the recovery of the data. [FIPS PUB 140-1]

APPENDIX C

REFERENCE LIST

American Bankers Association, *American National Standard for Financial Institution Key Management (Wholesale),* ANSI X9.17-1995, Washington, DC, 1995.

American Bankers Association, *American National Standard for Financial Institution Message Authentication,* ANSI X9.9-1986 (R1994), Washington, DC, Approved August 15, 1986, (Revised, 1994).

American Bankers Association, *Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)),* ANSI X9.31-1998, Washington, DC, 1998.

American Bankers Association, *Key Agreement and Key Transport Using Elliptic Curve-Based Cryptography*, Working Draft ANSI X9.63,1997.

American Bankers Association, *Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS): Requirements and Procedures, ANSI TG-19,* Washington, DC, 1999.

American Bankers Association, *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, Working Draft ANSI X9.42-2000.

American Bankers Association, *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm*, ANSI X9.62-1998.

American Bankers Association, *Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry,* Part 1: *The Digital Signature Algorithm (DSA)*, ANSI X9.30-1997*.*

American Bankers Association, *Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: The Secure Hash Algorithm (SHA-1) (Revised)*, ANSI X9.30-1993, Part 2*.*

American Bankers Association, *The Transport of Symmetric Algorithms Keys Using Reversible Public Key Cryptography*, Working DRAFT ANSI X9.44-1998.

American Bankers Association, *Triple Data Encryption Algorithm Modes of Operation*, ANSI X9.52-1998, Washington, DC, 1998.

Burr, William E., *Public Key Infrastructure (PKI) Version 1 Technical Specifications - Part C: Concept of Operations*, Federal PKI Technical Working Group, Nov. 16, 1995.

Burr, William, Donna Dodson, Noel Nazario, W. Timothy Polk, *Minimum Interoperability Specification for PKI Components, Version 1*, Special Publication 800-15, Gaithersburg, MD, National Institute of Standards and Technology, January 1998.

Common Criteria for Information Technology Security Evaluation, Version 2.0, May 1998, International Standard ISO/IEC 15408 *Evaluation Criteria for Information Technology Security*, ISO/IEC JTC1 and Common Criteria Implementation Board.

Computer Security Act of 1987, 40 U.S. Code 759, (Public Law 100-235), 8 January 1988.

Department of Defense, *Department of Defense Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD, December 1985.

Diffie, W. and M.E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, v. IT-22, n. 6, Nov. 1976, pp. 644-654.

Dodson, D., Keller, S. S., Chang, S., and Smid. M. E., *Technical Component for the CEFMS Electronic Signature System RFP*, NIST September 22, 1992.

Foti, J., Keller, S., and Dodson, Donna, *Security Review of the CEFMS Electronic Signature System*, NIST, May 17, 1996.

General Accounting Office (GAO), *Corps of Engineers Electronic Signature System*, GAO/AIMD-97-18R, November 19, 1996.

Government Paperwork Elimination Act (GPEA), Title XVII of Public Law 105-277, October 21, 1998.

Guttman, Barbara, *Computer Security Considerations in Federal Procurements: A Guide for Procurement Initiators, Contracting Officers, and Computer Security Officials*, Special Publication 800-4, Gaithersburg, MD, National Institute of Standards and Technology, March 1992.

Guttman, Barbara and Edward Roback, *An Introduction to Computer Security: The NIST Handbook*, Special Publication 800-12, Gaithersburg, MD, National Institute of Standards and Technology, February 6, 1996.

Keller, Sharon and Miles Smid, *Modes of Operation Validation System (MOVS): Requirements and Procedures*, Special Publication 800-17, Gaithersburg, MD, National Institute of Standards and Technology, February 1998.

MacLean, Abby, *EZ_ERA32 and the Key Recovery Demonstration Project (KRDP)*, Office of Energy Research, Department of Energy, 1998.

Menezes, Alfred J., vanOorschot, Paul C, and Vanstone, Scott A., *Handbook of Applied Cryptography*, CRC Press, Inc., New York, 1997.

MITRE Corporation, *Public Key Infrastructure Study, Final Report,* National Institute of Standards and Technology, April 1994.

Myers, M, R. Ankney, A. Malpani, S. Galperin, and C. Adams, *Internet Public key Infrastructure Online Certificate Status Protocol - OSCP*, draft, Sept. 1998.

National Institute of Standards and Technology, *Advanced Authentication Technology,* NIST ITL Bulletin, 1991-12.

National Institute of Standards and Technology, *Advanced Encryption Standard*, NIST ITL Bulletin, 1997-02.

National Institute of Standards and Technology, *Computer Data Authentication*, Federal Information Processing Standards Publication 113, May 30, 1985.

National Institute of Standards and Technology, *Computer Security Publications*, NIST Publication List 91, Revised October 1999.

National Institute of Standards and Technology, *Cryptographic Standards and Supporting Infrastructures: A Status Report*, NIST ITL Bulletin, 1997-09.

National Institute of Standards and Technology, *Data Encryption Standard*, Federal Information Processing Standards Publication 46-2, December 30, 1993.

National Institute of Standards and Technology, *DES Modes of Operation*, Federal Information Processing Standards Publication 81, December 2, 1980.

National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186, May 19, 1994.

National Institute of Standards and Technology, *Digital Signature Standard,* NIST ITL Bulletin, 1994-12.

National Institute of Standards and Technology, *Digital Signature Standard Validation System (DSSVS) User's Guide*, June 20, 1997.

National Institute of Standards and Technology, *Entity Authentication Using Public Key Cryptography*, Federal Information Processing Standards Publication 196, February 18, 1997.

National Institute of Standards and Technology, *Escrowed Encryption Standard (EES)*, Federal Information Processing Standards Publication 185, February 1994.

National Institute of Standards and Technology, *A Framework for Cryptographic Standards,* NIST ITL Bulletin, 1995-08.

National Institute of Standards and Technology, *Guideline for the Use of Advanced Authentication Technology Alternatives*, Federal Information Processing Standards Publication 190, September 28, 1994.

National Institute of Standards and Technology, *Implementation Issues for Cryptography*, NIST ITL Bulletin, 1996-08.

National Institute of Standards and Technology, *Key Management Using ANSI X9.17*, Federal Information Processing Standards Publication 171, April 27, 1992.

National Institute of Standards and Technology, *Public Key Infrastructure Technology*, NIST ITL Bulletin, 1997-07.

National Institute of Standards and Technology, *Secure Hash Standard*, Federal Information Processing Standards Publication 180-1, April 17, 1995.

National Institute of Standards and Technology, *Security Requirements for Cryptographic Modules*, Federal Information Processing Standards Publication 140-1, January 11, 1994.

National Security Agency (NSA), *Security Service API: Cryptographic API Recommendation Second Edition*, NSA Cross Organization CAPI Team, July 1, 1996.

National Security Telecommunications and Information Systems Security Committee, *National Information Systems Security Glossary*, NSTISSI No. 4009, 5 June 1992.

Nechvatal, James, *Public-Key Cryptography*, Special Publication 800-2, Gaithersburg, MD, National Institute of Standards and Technology, April 1991.

Office of Management and Budget, *Guidance for Preparation of Security Plans for Federal Computer Systems That Contain Sensitive Information*, OMB Bulletin No. 90-08, 9 July 1990.

Office of Management and Budget, *Security of Federal Automated Information Resources*, Appendix III to OMB Circular No. A-130, February 8, 1996.

RSA Laboratories, *Diffie-Hellman Key-Agreement Standard*, Technical Note Version 1.4, PKCS #3, November 1, 1993.

Schneier, Bruce, *Applied Cryptography*, *Second Edition*, John Wiley & Sons, Inc., New York, c. 1996.

Smith, B. H., *Secure Electronic Grants: Key Recovery Demonstration Project Phase I Accomplishments, Test Results and Remaining Tasks*, U.S. Department of Transportation, July 24, 1998.

Swanson, Marianne and Barbara Guttman, *Generally Accepted Principles and Practices for Securing Information Technology Systems*, Special Publication 800-14, Gaithersburg, MD, National Institute of Standards and Technology, September 20, 1995.

X/Open, *X/Open Preliminary Specification: Generic Cryptographic Service API*, draft 8, April 20, 1996.

West, John P., *Electronic Certification System (ECS) Development and Enhancement*, Financial Management Service, Department of the Treasury, April 8, 1998.

West, John P. and Chris Shanefelter, *Decision Fact Sheet, Electronic Certification System (ECS)*, Financial Management Service, Department of the Treasury, March 18, 1998.