

**NIST Special Publication 800-175B**

---

**Guideline for Using  
Cryptographic Standards in the  
Federal Government:**  
*Cryptographic Mechanisms*

---

Elaine Barker

---

C O M P U T E R   S E C U R I T Y

---

**NIST**  
National Institute of  
Standards and Technology  
U.S. Department of Commerce

**NIST Special Publication 800-175B**

**Guideline for Using  
Cryptographic Standards in the  
Federal Government:**

*Cryptographic Mechanisms*

Elaine Barker  
*Computer Security Division  
Information Technology Laboratory*

March 2016



U.S. Department of Commerce  
*Penny Pritzker, Secretary*

National Institute of Standards and Technology  
*Willie May, Under Secretary of Commerce for Standards and Technology and Director*

## Authority

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3541 *et seq.*, Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-175B  
Natl. Inst. Stand. Technol. Spec. Publ. 800-175B, 77 pages (March 2016)  
CODEN: NSPUE2

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at <http://csrc.nist.gov/publications>.

**Public comment period: *March 11, 2016* through *April 29, 2016***

National Institute of Standards and Technology  
Attn: Computer Security Division, Information Technology Laboratory  
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930  
Email: SP800-175@nist.gov

## **Reports on Computer Systems Technology**

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

### **Abstract**

This document is intended to provide guidance to the Federal Government for using cryptography and NIST's cryptographic standards to protect sensitive, but unclassified digitized information during transmission and while in storage. The cryptographic methods and services to be used are discussed.

### **Keywords**

Asymmetric-key algorithm, authentication, confidentiality, cryptography, digital signatures, encryption, integrity, key agreement, key derivation, key management, key transport, key wrapping, message authentication codes, non-repudiation, Public Key Infrastructure, random bit generation, symmetric-key algorithm.

## **Acknowledgments**

The author wishes to thank the authors of SP 800-21 from which this document was derived, Annabelle Lee and William C. Barker, along with those colleagues that reviewed drafts of this document and contributed to its development. The author also gratefully acknowledges and appreciates the many comments from the public and private sectors whose thoughtful and constructive comments improved the quality and usefulness of this publication.

**Table of Contents**

**SECTION 1: INTRODUCTION ..... 1**

- 1.1 Background and Purpose ..... 1
- 1.2 Audience ..... 2
- 1.3 Scope..... 2
- 1.4 Background..... 2
- 1.5 Terms and Definitions..... 3
- 1.6 Acronyms..... 9
- 1.7 Content..... 10

**SECTION 2: STANDARDS AND GUIDELINES ..... 12**

- 2.1 Benefits of Standards ..... 12
- 2.2 Federal Information Processing Standards and Special Publications ..... 13
  - 2.2.1 The Use of FIPS and SPs ..... 13
  - 2.2.2 FIPS Waivers..... 14
- 2.3 Other Standards Organizations ..... 14
  - 2.3.1 American National Standards Institute (ANSI)..... 14
  - 2.3.2 Institute of Electrical and Electronics Engineers (IEEE) Standards Association 15
  - 2.3.3 Internet Engineering Task Force (IETF) ..... 16
  - 2.3.4 International Organization for Standardization (ISO)..... 17
  - 2.3.5 Trusted Computing Group (TCG)..... 18

**SECTION 3: CRYPTOGRAPHIC ALGORITHMS ..... 19**

- 3.1 Cryptographic Hash Functions ..... 19
- 3.2 Symmetric-Key Algorithms..... 20
  - 3.2.1 Block Cipher Algorithms ..... 21
    - 3.2.1.1 Data Encryption Standard (DES).....22
    - 3.2.1.2 Triple Data Encryption Algorithm (TDEA) .....22
    - 3.2.1.3 SKIPJACK.....22
    - 3.2.1.4 Advanced Encryption Standard (AES).....23
    - 3.2.1.5 Modes of Operation.....23
  - 3.2.2 Hash-based Symmetric-key Algorithms ..... 23
- 3.3 Asymmetric-Key Algorithms..... 23
  - 3.3.1 DSA ..... 25
  - 3.3.2 ECDSA..... 25
  - 3.3.3 RSA ..... 26
  - 3.3.4 Diffie-Hellman and MQV ..... 26
- 3.4 Algorithm Security Strength ..... 26
- 3.5 Algorithm Lifetime ..... 27

**SECTION 4: CRYPTOGRAPHIC SERVICES ..... 28**

- 4.1 Data Confidentiality..... 28
- 4.2 Data Integrity and Source Authentication..... 29

- 4.2.1 Hash Functions ..... 29
- 4.2.2 Message Authentication Code Algorithms..... 30
  - 4.2.2.1 MACs Based on Block Cipher Algorithms.....31
  - 4.2.2.2 MACs Based on Hash Functions.....31
- 4.2.3 Digital Signature Algorithms ..... 32
- 4.3 Combining Confidentiality and Authentication in a Block-Cipher Mode of Operation..... 34
- 4.4 Random Bit Generation ..... 34
- 4.5 Symmetric vs. Asymmetric Cryptography ..... 36
- SECTION 5: KEY MANAGEMENT ..... 37**
- 5.1 General Key Management Guidance ..... 37
  - 5.1.1 Recommendation for Key Management..... 37
  - 5.1.2 Security Requirements for Cryptographic Modules..... 39
  - 5.1.3 Transitions to New Cryptographic Algorithms and Key Lengths..... 39
- 5.2 Cryptographic Key Management Systems..... 40
  - 5.2.1 Key Management Framework ..... 40
  - 5.2.2 Key Management System Profile..... 41
  - 5.2.3 Public Key Infrastructure ..... 41
    - 5.2.3.1 PKI Components, Relying Parties and Their Responsibilities.....42
    - 5.2.3.2 Basic Certificate Verification Process .....44
    - 5.2.3.3 CA Certificate Policies and Certificate Practice Statements .....44
    - 5.2.3.4 Federal Public Key Infrastructure .....45
- 5.3 Key Establishment ..... 45
  - 5.3.1 Key Generation ..... 45
  - 5.3.2 Key Derivation ..... 46
  - 5.3.3 Key Agreement ..... 47
  - 5.3.4 Key Transport..... 48
    - 5.3.4.1 SP 800-56A Key Transport .....48
    - 5.3.4.2 SP 800-56B Key Transport .....49
  - 5.3.5 Key Wrapping ..... 50
  - 5.3.6 Derivation of a Key from a Password ..... 51
- 5.4 Key Management Issues ..... 51
  - 5.4.1 Manual vs. Automated Key Establishment ..... 51
  - 5.4.2 Selecting and Operating a CKMS ..... 51
  - 5.4.3 Storing and Protecting Keys..... 51
  - 5.4.4 Cryptoperiods ..... 52
  - 5.4.5 Use Validated Algorithms and Cryptographic Modules ..... 52
  - 5.4.6 Control of Keying Material ..... 53
  - 5.4.7 Compromises..... 53

5.4.8 Accountability and Auditing ..... 54

**SECTION 6: OTHER ISSUES ..... 55**

6.1 Required Security Strength ..... 55

6.2 Interoperability ..... 55

6.3 When Algorithms are no Longer Approved ..... 56

6.4 Registration Authorities (RAs) ..... 56

6.5 Cross Certification ..... 56

**Appendix A: References ..... 57**



1

## SECTION 1: INTRODUCTION

### 2 1.1 Background and Purpose

3 In today's environment of increasingly open and interconnected systems and networks and  
4 the use of mobile devices, network and data security are essential for the optimum safe use  
5 of this information technology. Cryptographic techniques should be considered for the  
6 protection of data that is sensitive, has a high value, or is vulnerable to unauthorized  
7 disclosure or undetected modification during transmission or while in storage.

8 Cryptography is a branch of mathematics that is based on the transformation of data and  
9 can be used to provide several security services: confidentiality, data integrity  
10 authentication, and source authentication, and also to support non-repudiation.

- 11 • *Confidentiality* is the property whereby sensitive information is not disclosed to  
12 unauthorized entities. Confidentiality can be provided by a cryptographic process  
13 called *encryption*.
- 14 • *Data integrity* is a property whereby data has not been altered in an unauthorized  
15 manner since it was created, transmitted or stored. The process of determining the  
16 integrity of the data is called *data integrity authentication*.
- 17 • *Source authentication* is a process that provides assurance of the source of  
18 information to a receiving entity; source authentication can also be considered as  
19 identity authentication (i.e., providing assurance of an entity's identity). A special  
20 case of source authentication is called *non-repudiation*, whereby support for  
21 assurance of the source of the information is provided to a third party.

22 This document is one part in a series of documents intended to provide guidance to the  
23 Federal Government for using cryptography to protect its sensitive, but unclassified  
24 digitized information during transmission and while in storage; hereafter, the shortened  
25 term “sensitive” will be used to refer to this class of information. Other sectors are invited  
26 to use this guidance on a voluntary basis. The following are the initial publications to be  
27 included in the SP 800-175 series. Additional documents may be provided in the future.

- 28 • [SP 800-175A](#) will provide guidance on the determination of requirements for using  
29 cryptography. It will include the laws and regulations for the protection of the  
30 Federal Government’s sensitive information, guidance for the conduct of risk  
31 assessments to determine what needs to be protected and how best to protect that  
32 information, and a discussion of the required security-related documents (e.g.,  
33 various policy and practice documents). DOCUMENT UNDER  
34 DEVELOPMENT.
- 35 • SP 800-175B (this document) discusses the cryptographic methods and services  
36 available for the protection of the Federal Government’s sensitive information and  
37 provides an overview of NIST’s cryptographic standards.

## 38 1.2 Audience

39 This document is intended for federal employees and others who are responsible for  
40 providing and using cryptographic services to meet identified security requirements. This  
41 document might be used by:

- 42 • Program managers responsible for selecting and integrating cryptographic  
43 mechanisms into a system,
- 44 • A technical specialist requested to select one or more cryptographic  
45 methods/techniques to meet a specified requirement,
- 46 • A procurement specialist developing a solicitation for a system, network or service  
47 that will require cryptographic methods to perform security functionality, and
- 48 • Users of cryptographic services.

49 The goal is to provide these individuals with sufficient information to allow them to make  
50 informed decisions about the cryptographic methods that will meet their specific needs to  
51 protect the confidentiality and integrity of data that is transmitted and/or stored in a system  
52 or network, as well as to obtain assurance of its authenticity.

53 This document is not intended to provide information on the federal procurement process  
54 or to provide a technical discussion on the mathematics of cryptography and cryptographic  
55 algorithms.

## 56 1.3 Scope

57 This document limits its discussion of cryptographic methods to those that conform to  
58 Federal Information Processing Standards (FIPS) and NIST Special Publications (SPs),  
59 which are collectively discussed as NIST "standards" in this document. While the Federal  
60 Government is required to use these standards, when applicable, industry and national and  
61 international standards bodies have also adopted these cryptographic methods.

62 This document provides information on selecting and using cryptography in new or  
63 existing systems.

## 64 1.4 Background

65 The use of cryptography relies upon two basic components: an *algorithm* (or cryptographic  
66 methodology) and a *key*. The algorithm is a mathematical function, and the key is a  
67 parameter used during the cryptographic process. The algorithm and key are used together  
68 to apply cryptographic protection to data (e.g., to encrypt the data or to generate a digital  
69 signature) and to remove or check the protection (e.g., to decrypt the encrypted data or to  
70 verify the digital signature). The security of the cryptographic protection relies on the  
71 secrecy of the key, while the algorithm specification is publicly available.

72 In order to use a cryptographic algorithm, cryptographic keys must be “in place”, i.e., keys  
73 must be established for and/or between parties that intend to use cryptography. Keys may  
74 be established either manually (e.g., via a trusted courier) or using an automated method.  
75 However, when an automated method is used, authentication is required for the

76 participating entities that relies on an established trust infrastructure, such as a Public Key  
77 Infrastructure (PKI) or on a manually distributed authentication key.

78 In general, keys used for one purpose (e.g., the generation of digital signatures) must not  
79 be used for another purpose (e.g., for key establishment) because the use of the same key  
80 for two different cryptographic processes may weaken the security provided by one or both  
81 of the processes. See Section 5.2 in SP 800-57, Part 1<sup>1</sup> for further information.

82 **1.5 Terms and Definitions**

83 The following terms and definitions are used in this document. In general, the definitions  
84 are drawn from FIPS and NIST Special Publications.

Algorithm	A clearly specified mathematical process for computation; a set of rules that, if followed, will give a prescribed result.
Approved	FIPS-Approved and/or NIST-recommended. An <a href="#">algorithm</a> or technique that is either 1) specified in a FIPS or NIST recommendation, or 2) specified elsewhere and adopted by reference in a FIPS or NIST Recommendation.
Asymmetric-key algorithm	See <a href="#">public-key algorithm</a> .
Authentication	A process that provides assurance of the source and <a href="#">integrity</a> of information that is communicated or stored.
Bit string	An ordered sequence of 0's and 1's.
Block cipher algorithm	A family of functions and their inverse functions that is parameterized by <a href="#">cryptographic keys</a> ; the functions map <a href="#">bit strings</a> of a fixed length to bit strings of the same length.
Certificate (or public key certificate)	A set of data that uniquely identifies an <a href="#">entity</a> , contains the entity's <a href="#">public key</a> and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity. Additional information in the certificate could specify how the <a href="#">key</a> is used and the validity period of the certificate.
Certificate Revocation List (CRL)	A list of revoked but unexpired <a href="#">certificates</a> issued by a <a href="#">Certification Authority</a> .
Certification Authority (CA)	The <a href="#">entity</a> in a <a href="#">public key infrastructure</a> (PKI) that is responsible for issuing <a href="#">certificates</a> and exacting compliance to a PKI policy.

<sup>1</sup> SP 800-57, Part 1: Recommendation for Key Management: General Guideline.

Ciphertext	Data in its <a href="#">encrypted</a> form.
Compromise	The unauthorized disclosure, modification, substitution or use of sensitive data (e.g., <a href="#">keying material</a> and other security-related information).
Confidentiality	The property that sensitive information is not disclosed to unauthorized <a href="#">entities</a> .
Cross certify	The establishment of a trust relationship between two <a href="#">Certification Authorities</a> (CAs) through the signing of each other's <a href="#">public key</a> in a <a href="#">certificate</a> referred to as a "cross-certificate."
Cryptographic algorithm	A well-defined computational procedure that takes variable inputs, including a <a href="#">cryptographic key</a> (if applicable), and produces an output.
Cryptographic checksum	A mathematical value created using a <a href="#">cryptographic algorithm</a> that is assigned to data and later used to test the data to verify that the data has not changed.
Cryptographic hash function	A function that maps a bit string of arbitrary length to a fixed-length bit string. <a href="#">Approved hash functions</a> satisfy the following properties: <ol style="list-style-type: none"> <li>1. (One-way) It is computationally infeasible to find any input that maps to any pre-specified output, and</li> <li>2. (Collision resistant) It is computationally infeasible to find any two distinct inputs that map to the same output.</li> </ol>
Cryptographic key	A parameter used in conjunction with a <a href="#">cryptographic algorithm</a> that determines its operation in such a way that an <a href="#">entity</a> with knowledge of the key can reproduce or reverse the operation, while an entity without knowledge of the key cannot. Examples include: <ol style="list-style-type: none"> <li>1. The transformation of <a href="#">plaintext</a> data into <a href="#">ciphertext</a> data,</li> <li>2. The transformation of ciphertext data into plaintext data,</li> <li>3. The computation of a <a href="#">digital signature</a> from data,</li> <li>4. The verification of a digital signature,</li> <li>5. The computation of an <a href="#">authentication code</a> from data,</li> <li>6. The verification of an authentication code from data and a received authentication code, and</li> </ol>

	7. The computation of a <a href="#">shared secret</a> that is used to derive <a href="#">keying material</a> .
Cryptographic module	The set of hardware, software and/or firmware that implements <a href="#">approved</a> security functions (including <a href="#">cryptographic algorithms</a> and <a href="#">key</a> generation) and is contained within the cryptographic boundary.
Cryptographic primitive	A low-level <a href="#">cryptographic algorithm</a> used as a basic building block for higher-level cryptographic algorithms.
Cryptography	The discipline that embodies principles, means and methods for providing information security, including <a href="#">confidentiality</a> , <a href="#">data integrity</a> , and <a href="#">non-repudiation</a> .
Cryptoperiod	The time span during which a specific <a href="#">key</a> is authorized for use or in which the keys for a given system may remain in effect.
Data integrity	A property whereby data has not been altered in an unauthorized manner since it was created, transmitted or stored.
Decryption	The process of changing <a href="#">ciphertext</a> into <a href="#">plaintext</a> using a <a href="#">cryptographic algorithm</a> and <a href="#">key</a> .
Digital signature	The result of a cryptographic transformation of data that, when properly implemented, provides the services of: <ol style="list-style-type: none"> <li>1. <a href="#">Source authentication</a>,</li> <li>2. <a href="#">Data integrity</a>, and</li> <li>3. Supports signer <a href="#">non-repudiation</a>.</li> </ol>
Digital Signature Algorithm (DSA)	An <a href="#">algorithm</a> used by a <i>signatory</i> to generate a <a href="#">digital signature</a> on data and by a <i>verifier</i> to obtain assurance of the source and <a href="#">integrity</a> of the signed information.
Elliptic Curve Digital Signature Algorithm (ECDSA)	A <a href="#">digital signature algorithm</a> that is an analog of DSA using elliptic curve mathematics and specified in ANS <a href="#">X9.62</a> .
Encryption	The process of changing <a href="#">plaintext</a> into <a href="#">ciphertext</a> for the purpose of security or privacy.
Entity	An individual (person), organization, device or process.
Ephemeral key pair	A short-term <a href="#">key pair</a> that is generated when needed and used only once; the <a href="#">public key</a> is not certified.
Function	As used in this document, used interchangeability with <a href="#">algorithm</a> .

Hash function	See <a href="#">cryptographic hash function</a> .
Hash value	The result of applying a <a href="#">hash function</a> to information; also called a <a href="#">message digest</a> .
Initialization Vector (IV)	A vector used in defining the starting point of a cryptographic process.
Integrity	The property that protected data has not been modified or deleted in an unauthorized and undetected manner.
Interoperability	The ability of one <a href="#">entity</a> to communicate with another entity.
Key	See <a href="#">cryptographic key</a> .
Key agreement	A (pair-wise) <a href="#">key-establishment</a> procedure where the resultant secret <a href="#">keying material</a> is a function of information contributed by two participants, so that no party can predetermine the value of the secret keying material independently from the contributions of the other party. Contrast with <a href="#">key-transport</a> .
Key derivation	The process by which one or more <a href="#">keys</a> are derived from either a pre-shared key, or a <a href="#">shared secret</a> and other information.
Key establishment	The procedure that results in <a href="#">keying material</a> that is shared among different parties.
Key management	The activities involving the handling of <a href="#">cryptographic keys</a> and other related security parameters (e.g., <a href="#">IVs</a> , counters) during the entire life cycle of the keys, including the generation, storage, <a href="#">establishment</a> , entry and output, and destruction.
Key pair	A <a href="#">public key</a> and its corresponding <a href="#">private key</a> ; a key pair is used with a <a href="#">public key (asymmetric-key) algorithm</a> .
Key transport	A <a href="#">key-establishment</a> procedure whereby one party (the sender) selects a value for the secret <a href="#">keying material</a> and then securely distributes that value to another party (the receiver). Contrast with <a href="#">key agreement</a> .
Key-wrapping key	A <a href="#">symmetric key</a> used to provide <a href="#">confidentiality</a> and <a href="#">integrity</a> protection for other <a href="#">keys</a> .
Keying material	The data (e.g., <a href="#">keys</a> and <a href="#">IVs</a> ) necessary to establish and maintain cryptographic <a href="#">keying relationships</a> .
Keying relationship, cryptographic	The state existing between two <a href="#">entities</a> such that they share at least one <a href="#">cryptographic key</a> .

Message Authentication Code (MAC)	A <a href="#">cryptographic checksum</a> on data that uses a <a href="#">symmetric key</a> to detect both accidental and intentional modifications of data.
Message digest	See <a href="#">hash value</a> .
Mode of operation	An <a href="#">algorithm</a> that uses a lower-level algorithm to provide a cryptographic service, such as <a href="#">confidentiality</a> or <a href="#">Authentication</a> . The lower-level algorithm is typically a <a href="#">block cipher algorithm</a> , such as AES.
NIST standard	Federal Information Processing Standard (FIPS) or Special Publication (SP).
Non-repudiation	A service using a <a href="#">digital signature</a> that is used to support a determination of whether a message was actually signed by a given <a href="#">entity</a> .
Plaintext	Intelligible data that has meaning and can be understood without the application of <a href="#">decryption</a> .
Primitive	See <a href="#">Cryptographic primitive</a> .
Private key	<p>A cryptographic key, used with a <a href="#">public key cryptographic algorithm</a> that is uniquely associated with an <a href="#">entity</a> and is not made public. In an asymmetric (public) key cryptosystem, the private key is associated with a <a href="#">public key</a>. Depending on the algorithm, the private key may be used to:</p> <ol style="list-style-type: none"> <li>1. Compute the corresponding public key,</li> <li>2. Compute a <a href="#">digital signature</a> that may be verified by the corresponding public key,</li> <li>3. Decrypt data that was encrypted by the corresponding public key, or</li> <li>4. Compute a piece of common shared data, together with other information.</li> </ol>
Public key	<p>A cryptographic key used with a <a href="#">public key cryptographic algorithm</a>, that is uniquely associated with an <a href="#">entity</a> and that may be made public. In an asymmetric (public) key cryptosystem, the public key is associated with a <a href="#">private key</a>. The public key may be known by anyone and, depending on the algorithm, may be used to:</p> <ol style="list-style-type: none"> <li>1. Verify a <a href="#">digital signature</a> that is signed by the corresponding private key,</li> </ol>

	<p>2. Encrypt data that can be decrypted by the corresponding private key,</p> <p>3. Compute a piece of common shared data.</p>
Public key (asymmetric) cryptographic algorithm	A <a href="#">cryptographic algorithm</a> that uses two related keys, a <a href="#">public key</a> and a <a href="#">private key</a> . The two keys have the property that determining the private key from the public key is computationally infeasible.
Public Key Infrastructure (PKI)	A framework that is established to issue, maintain and revoke <a href="#">public key certificates</a> .
Relying party	An <a href="#">entity</a> that relies on the <a href="#">certificate</a> and the <a href="#">CA</a> that issued the certificate to verify the identity of the certificate owner, and the validity of the <a href="#">public key</a> , associated <a href="#">algorithms</a> and any relevant parameters in the certificate, as well as the owner’s possession of the corresponding <a href="#">private key</a> .
RSA	A <a href="#">public-key algorithm</a> that is used for <a href="#">key establishment</a> and the generation and verification of <a href="#">digital signatures</a> .
Secret key	A <a href="#">cryptographic key</a> that is used with a <a href="#">symmetric (secret key) cryptographic algorithm</a> and is not made public. The use of the term “secret” in this context does not imply a classification level, but rather implies the need to protect the key from disclosure.
Secret key (symmetric) cryptographic algorithm	See <a href="#">symmetric (secret key) algorithm</a> .
Sensitive (information)	Sensitive, but unclassified information.
Security strength	A number associated with the amount of work (that is, the number of operations) that is required to break a <a href="#">cryptographic algorithm</a> or system. In this Recommendation, the security strength is specified in bits and is a specific value from the set {80, 112, 128, 192, 256}. Note that the 80-bit security strength is no longer <a href="#">approved</a> , since it does not provide adequate protection.
Shared secret	A secret value that is computed during a <a href="#">key-agreement</a> process and is used as input to a derive a <a href="#">key</a> using a <a href="#">key-derivation</a> method.



Signature generation	The use of a <a href="#">digital signature algorithm</a> and a <a href="#">private key</a> to generate a <a href="#">digital signature</a> on data.
Signature verification	The use of a <a href="#">digital signature</a> and a <a href="#">public key</a> to verify a digital signature.
Source authentication	A process that provides assurance of the source of information.
Static key pair	A long-term <a href="#">key pair</a> for which the <a href="#">public key</a> is often provided in a public-key <a href="#">certificate</a> .
Symmetric key	A single <a href="#">cryptographic key</a> that is used with a <a href="#">symmetric (secret key) algorithm</a> .
Symmetric (secret key) algorithm	A <a href="#">cryptographic algorithm</a> that uses the same <a href="#">secret key</a> for an operation and its complement (e.g., <a href="#">encryption</a> and <a href="#">decryption</a> ).

## 85 1.6 Acronyms

86	AES	Advanced Encryption Standard; specified in <a href="#">FIPS 197</a> .
87	ANS	American National Standard.
88	ANSI	American National Standard Institute.
89	ASC	Accredited Standards Committee.
90	CA	Certification Authority.
91	CBC	Cipher Block Chaining mode; specified in <a href="#">SP 800-38A</a> .
92	CFB	Cipher Feedback mode; specified in <a href="#">SP 800-38A</a> .
93	CKMS	Cryptographic Key Management System.
94	CP	Certificate Policy.
95	CPS	Certification Practice Statement.
96	CRL	Certificate Revocation List.
97	CTR	Counter mode; specified in <a href="#">SP 800-38A</a> .
98	DES	Data Encryption Standard; originally specified in FIPS 46; now provided in <a href="#">SP 800-67</a> .
99		
100	DH	Diffie-Hellman algorithm.
101	DNSSEC	Domain Name System Security Extensions.
102	DRBG	Deterministic Random Bit Generator; specified in <a href="#">SP 800-90A</a> .
103	DSA	Digital Signature Algorithm; specified in <a href="#">FIPS 186</a> .
104	ECB	Electronic Codebook mode; specified in <a href="#">SP 800-38A</a> .
105	ECDSA	Elliptic Curve Digital Signature Algorithm.
106	EMC	Electromagnetic Compatibility.
107	FCKMS	Federal Cryptographic Key Management System.
108	FIPS	Federal Information Processing Standard.
109	FISMA	Federal Information Security Management Act.
110	GCM	Galois Counter Mode; specified in <a href="#">SP 800-38D</a> .
111	HMAC	Keyed-Hash Message Authentication Code; specified in <a href="#">FIPS 198</a> .
112	IEC	International Electrotechnical Commission.

113	IEEE	Institute of Electrical and Electronics Engineers.
114	IETF	Internet Engineering Task Force.
115	EMI	Electromagnetic Interference.
116	INCITS	International Committee for Information Technology Standards.
117	IPSEC	Internet Protocol Security.
118	ISO	International Standards Organization.
119	IT	Information Technology.
120	MAC	Message Authentication Code.
121	MQV	Menezes-Qu-Vanstone algorithm; specified in <a href="#">SP 800-56A</a> .
122	NRBG	Non-deterministic Random Bit Generator.
123	NIST	National Institute of Standards and Technology.
124	OFB	Output Feedback mode; specified in <a href="#">SP 800-38A</a> .
125	OTAR	Over-the-Air-Rekeying.
126	PKI	Public Key Infrastructure.
127	RA	Registration Authority.
128	RBG	Random Bit Generator.
129	RFC	Request for Comment.
130	RSA	Rivest, Shamir, Adleman.
131	SHA	Secure Hash Algorithm.
132	SP	Special Publication.
133	SSH	Secure Shell protocol.
134	TCG	Trusted Computing Group.
135	TDEA	Triple Data Encryption Algorithm; specified in <a href="#">SP 800-67</a> .
136	TLS	Transport Layer Security.

## 137 1.7 Content

138 This document is organized into the following sections:

- 139 • Section 1 provides an introduction to the SP 800-175 series of publications and to  
140 this document in particular, and provides a glossary of terms and a list of acronyms.
- 141 • Section 2 discusses the importance of standards, as well as the national and  
142 international standards bodies concerned with cryptography.
- 143 • Section 3 introduces the **approved** algorithms used for encryption, digital signature  
144 and key-establishment, and provides discussions on security strengths and  
145 algorithm lifetime.
- 146 • Section 4 discusses the services that cryptography can provide: data confidentiality,  
147 data integrity authentication, source authentication and support for non-repudiation.
- 148 • Section 5 discusses the key management required for the use of cryptography,  
149 providing general guidance and discussions on key-management systems, key-  
150 establishment mechanisms and random bit generation.
- 151 • Section 6 discusses additional issues associated with the use of cryptography.

152 There is one appendix in this document:

- 153 • Appendix A lists applicable Federal Information Processing Standards,  
154 recommendations, and guidelines.  
155

156

## SECTION 2: STANDARDS AND GUIDELINES

### 157 2.1 Benefits of Standards

158 Standards define common practices, methods, and measures/metrics. Standards provide  
159 solutions that have been evaluated by experts in relevant areas, reviewed by the public and  
160 subsequently accepted by a wide community of users. By using standards, organizations  
161 can reduce costs and protect their investments in technology.

162 Standards provide the following benefits:

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

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

173 • **Security.** Standards may be used to establish a common approved level of security.  
174 For example, most agency managers are not cryptographic security experts, and, by  
175 using an **approved** cryptographic algorithm and key length, a manager knows that  
176 the algorithm has been found to be adequate for the protection of sensitive  
177 government data and has been subjected to a significant period of public analysis  
178 and comment.

179 • **Quality.** Standards may be used to assure the quality of a product. Standards may:  
180     ○ Specify how a feature is to be implemented,  
181     ○ Require self-tests to ensure that the product is still functioning correctly,  
182     and  
183     ○ Require specific documentation to assure proper implementation and  
184     product-change management.

185 Many NIST standards have associated conformance tests and specify the  
186 conformance requirements. The conformance tests may be administered by NIST-  
187 accredited laboratories and provide validation that the NIST standard was correctly  
188 implemented.

189 • **Common Form of Reference.** A NIST standard may become a common form of  
190 reference to be used in testing/evaluating a vendor's product. For example, [FIPS](#)  
191 [140](#)<sup>2</sup> contains security and integrity requirements for *any* cryptographic module  
192 implementing cryptographic operations.

---

<sup>2</sup> FIPS 140, Security Requirements for Cryptographic Modules.

- 193       • **Cost Savings.** A standard can save money by providing a single commonly  
194       accepted specification. Without standards, users may be required to become  
195       experts in every information technology (IT) product that is being considered for  
196       procurement. Also, without standards, products may not interoperate with different  
197       products purchased by other users. This will result in a significant waste of money  
198       or in the delay of implementing IT.

## 199   **2.2 Federal Information Processing Standards and Special Publications**

### 200   **2.2.1 The Use of FIPS and SPs**

201   A Federal Information Processing Standard (FIPS) is *mandatory* for the Federal  
202   Government whenever the type of service provided by that standard is required by a federal  
203   agency for the protection of sensitive information. For example, [FIPS 197](#)<sup>3</sup> contains a  
204   specific set of technical security requirements for the AES algorithm. Whenever AES is  
205   used by an agency, the implementation and use must conform to FIPS 197. A FIPS is  
206   **approved** via a signature by the Secretary of Commerce.

207   A NIST Special Publication (SP) is similar to a FIPS, but is not mandatory unless a  
208   particular government agency (e.g., OMB) makes it so. An SP does not need the approval  
209   of the Secretary of Commerce.

210   Although the requirements for the use of a FIPS and an SP are different, both types of  
211   publications have been subjected to the same review process by the federal agencies and  
212   the public. The approval process for a FIPS is more formal than that of an SP, and  
213   subsequently takes longer for the initial approval and the approval of any subsequent  
214   revisions.

215   [When a federal agency requires the use of cryptography \(e.g., for encryption\), an approved](#)  
216   algorithm must be used; approval is indicated by inclusion in a FIPS or SP. For example,  
217   two **approved** algorithms for encryption are AES (as specified in [FIPS 197](#)) and TDEA  
218   (as specified in [SP 800-67](#)<sup>4</sup>). Whenever encryption is used by a federal agency for the  
219   protection of sensitive information, either AES or TDEA must be used. Whenever AES is  
220   to be used, it must be implemented as specified in FIPS 197; whenever TDEA is to be used,  
221   it must be implemented as specified in SP 800-67. In addition to using **approved**  
222   algorithms, federal agencies are required to use only implementations of these algorithms  
223   that have been validated and are included in validated cryptographic modules (see [Section](#)  
224   [5.4.5](#) for further discussion).

225   When developing a specification or the criteria for the selection of a cryptographic  
226   mechanism or service, cryptographic algorithms specified in FIPSs and SPs must be used,  
227   when available. Some guidelines may be used to specify the functions that the algorithm

---

<sup>3</sup> FIPS 197, the Advanced Encryption Standard.

<sup>4</sup> SP 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher.

228 will perform (e.g., [FIPS 199](#)<sup>5</sup> or [SP 800-53](#)<sup>6</sup>). Other NIST standards specify the operation  
229 and use of specific types of algorithms (e.g., AES, DSA) and the level of independent  
230 testing required for classes of security environments (e.g., [FIPS 140](#)).

231 [Appendix A](#) contains a list of FIPS and SPs that apply to the implementation of  
232 cryptography in the Federal Government. Note that when a FIPS is revised, its number is  
233 commonly followed by a revision number that indicates the number of times that it has  
234 been revised (e.g., "FIPS 186-4" is used to indicate the fourth revision of [FIPS 186](#)); this  
235 practice is not used in the main body of this document; the reader must refer to the latest  
236 version of the FIPS or SP that has been officially **approved** (see  
237 <http://csrc.nist.gov/publications/index.html>;  
238 note that this site may also contain clearly marked draft publications) .

### 239 **2.2.2 FIPS Waivers**

240 In the past, a waiver was sometimes issued by an agency to indicate that the use of a FIPS  
241 was not required by that agency. However, the Federal Information Security Management  
242 Act (FISMA) of 2002 (P.L. 107-347) eliminated previously authorized provisions for  
243 waivers from FIPS (see [SP 800-175A](#) for a discussion).

## 244 **2.3 Other Standards Organizations**

245 NIST develops standards, recommendations, and guidelines that are used by vendors who  
246 are developing security products, components, and modules. These products may be  
247 acquired and used by federal government agencies. In addition, there are other groups that  
248 develop and promulgate standards. These organizations are briefly described below.

### 249 **2.3.1 American National Standards Institute (ANSI)**<sup>7</sup>

250 The American National Standards Institute (ANSI) is the administrator and coordinator of  
251 the United States (U.S.) private-sector voluntary standardization system. ANSI does not  
252 develop American National Standards itself; rather, it facilitates the development of  
253 standards by establishing consensus among qualified groups.

254 Several ANSI committees have developed standards that use cryptography, but the primary  
255 committee that has developed standards for the cryptographic algorithms themselves is  
256 Accredited Standards Committee (ASC) X9, which is a financial-industry committee<sup>8</sup>.  
257 Many of the standards developed within ASC X9 have been adopted within NIST standards  
258 (e.g., the Elliptic Curve Digital Signature Algorithm specified in American National  
259 Standard [X9.62](#)<sup>9</sup> [has been adopted in FIPS 186](#)); likewise, ASC X9 has approved the use

---

<sup>5</sup> FIPS 199, Standards for Security Categorization of Federal Information and Information Systems.

<sup>6</sup> SP 800-53, Recommended Security Controls for Federal Information Systems.

<sup>7</sup> Further information is available at the ANSI web site: [www.ansi.org](http://www.ansi.org).

<sup>8</sup> Further information is available at the ANSI X9 web site: [x9.org](http://x9.org).

<sup>9</sup> ANS X9.62, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).

260 of NIST standards via a registry of approved standards from non-ASC X9 sources (e.g.,  
261 AES, as specified in [FIPS 197](#)).

262 A number of ASC X9 standards have also been incorporated into the standards of other  
263 standards bodies, such as the International Standards Organization (ISO) (see [Section](#)  
264 [2.3.4](#)) via a Technical Advisory Group (TAG) called the International Committee on  
265 Information Technology Standards (INCITS). INCITS has been responsible for assuring  
266 that U.S. standards (e.g., both those developed by NIST and those developed within ASC  
267 X9) are incorporated within ISO standards.

### 268 **2.3.2 Institute of Electrical and Electronics Engineers (IEEE) Standards** 269 **Association**<sup>10</sup>

270 IEEE is an international, professional association that is dedicated to advancing  
271 technological innovation and excellence. The technical objectives of the IEEE focus on  
272 advancing the theory and practice of electrical, electronics and computer engineering, and  
273 computer science. IEEE develops and disseminates voluntary, consensus-based industry  
274 standards involving leading-edge electro-technology. IEEE supports international  
275 standardization and encourages the development of globally acceptable standards.

276 The Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA) is  
277 an organization within IEEE that develops global standards. It has more than one thousand  
278 active standards, some of which are related to cryptography.

279 [IEEE P1363](#)<sup>11</sup> is the only IEEE standard that focus on cryptography. It includes a series of  
280 standards on public-key cryptography. IEEE P1363 was developed at the same time as the  
281 ANSI public-key cryptographic standards, such as ANS [X9.31](#)<sup>12</sup>, [X9.42](#)<sup>13</sup>, [X9.44](#)<sup>14</sup>,  
282 [X9.62](#)<sup>15</sup>, and [X9.63](#)<sup>16</sup>, which were developed in ASC X9 (see [Section 2.4.1](#)).

- 283 • The first part of the [IEEE P1363](#) standard was published in 2000 and revised in  
284 2004 as [IEEE P1363a](#)<sup>17</sup>. It includes the basic public-key cryptography schemes,  
285 such as RSA encryption, signatures, the Digital Signature Algorithm (DSA), and  
286 key establishment using Diffie-Hellman (DH) and Menezes-Qu-Vanstone (MQV)  
287 over finite fields and elliptic curves.

---

<sup>10</sup> Further information is available at the IEEE-SA web site: [standards.ieee.org](http://standards.ieee.org).

<sup>11</sup> IEEE P1363: Standard Specifications for Public-Key Cryptography.

<sup>12</sup> ANS X9.31, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), which has now been withdrawn.

<sup>13</sup> ANS X9.42, Agreement of Symmetric Keys Using Discrete Logarithm Cryptography, which has now been withdrawn.

<sup>14</sup> ANS X9.44, Key Establishment Using Integer Factorization Cryptography.

<sup>15</sup> ANS X9.62, The Elliptic Curve Digital Signature Algorithm (ECDSA).

<sup>16</sup> ANS X9.63, *Key Agreement and Key Transport Using Elliptic Curve Cryptography*.

<sup>17</sup> IEEE P1363a, Standard Specifications for Public Key Cryptography - Amendment 1: Additional Techniques.



288 • [IEEE P1363.1](#)<sup>18</sup>, which was published in 2008, specifies NTRU encryption and  
289 signature schemes.

290 • [IEEE P1363.2](#)<sup>19</sup> was also published in 2008. It specifies password-authenticated  
291 key agreement and password-authenticated key retrieval schemes.

292 The schemes specified in IEEE P1363.1 and P1363.2 are not included in the NIST  
293 standards.

294 Cryptographic schemes are used in IEEE standards for different applications. One of the  
295 more notable is the IEEE 802 LAN/MAN group of standards, which are widely used  
296 computer networking standards for both wired (Ethernet) and wireless ([IEEE 802.11](#)<sup>20</sup>)  
297 networks. Cryptographic algorithms are used to protect wireless communications. The  
298 CCM mode for authentication and confidentiality specified in [SP 800-38C](#) was adopted  
299 from IEEE 802.11. Other AES modes of operations (e.g., GCM, which is specified in [SP](#)  
300 [800-38D](#)) are also used in IEEE 802 standards. IEEE 802 standards also use the SHA-1  
301 and SHA-2 family of hash functions specified in [FIPS 180](#) and used in HMAC, as specified  
302 in [FIPS 198](#).

303 XTS, a block cipher mode of operation specified in [SP 800-38E](#), was adopted from [IEEE](#)  
304 [P1619](#)<sup>21</sup>.

### 305 **2.3.3 Internet Engineering Task Force (IETF)**

306 The Internet Engineering Task Force (IETF) is an international community of network  
307 designers, operators, vendors, researchers, and technologists that work on the Internet  
308 architecture, and its techniques and protocols. The official technical specifications and  
309 recommendations of the IETF are called Request for Comments (RFCs).

310 The technical work of the IETF is done in its working groups, which are organized by topic  
311 into several areas, such as routing, transport and security. In the security area, different  
312 working groups work on security mechanisms for different protocols or applications. For  
313 example,

314 1. The PKIX (Public-Key Infrastructure X.509) Working Group (PKIX-WG)  
315 developed technical specifications and recommendations to support a Public Key  
316 Infrastructure, based on the [X.509](#) protocol, which is used to build a trust and  
317 authentication services infrastructure,

318 2. The IPSEC (Internet Protocol Security) working group developed a protocol and  
319 other technical recommendations for secure routing between network devices, and

320 3. The TLS (Transport Layer Security) working group has been specifying a  
321 communication protocol and technical recommendations to provide security  
322 services for communication between a server and a client, etc.

---

<sup>18</sup> IEEE P1363.1, Public-Key Cryptographic Techniques Based on Hard Problems over Lattices.

<sup>19</sup> IEEE P1363.2, Password-Based Public-Key Cryptography.

<sup>20</sup> IEEE 802.11, Wireless Local Area Networks.

<sup>21</sup> IEEE P1619, Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices.



323 NIST-approved cryptographic algorithms, such as block cipher modes of operation, hash  
324 functions, key establishment schemes, and digital signatures are used in various IETF  
325 protocols. For example, RFC 5288 specifies the AES Galois Counter Mode (GCM) Cipher  
326 Suites for TLS, based on [SP 800-38D](#).

327 Further information is available at the IETF web site: <http://ietf.org>.

### 328 **2.3.4 International Organization for Standardization (ISO)<sup>22</sup>**

329 ISO is a non-governmental, worldwide federation of national standards bodies. Its mission  
330 is to develop international standards that help to make industry more efficient and effective.  
331 ISO standards cover almost all aspects of technology and business, from food safety to  
332 computers, and from agriculture to healthcare. Experts from all over the world develop the  
333 standards that are required by their sector, using a consensus process.

334 ISO/IEC JTC 1 is a joint technical committee of the International Organization for  
335 Standardization (ISO) and the International Electrotechnical Commission (IEC). ISO/IEC  
336 JCT 1 SC 27 is the subcommittee for IT security. Working group 2 (WG2) is the group  
337 developing standards for cryptography and security mechanisms. It usually has more than  
338 twenty active projects to develop either a revision of an existing standard or a new standard.  
339 Each standard consists of multiple parts, and each part includes multiple algorithms and/or  
340 mechanisms.

341 The cryptographic algorithms and schemes in FIPS and SPs are usually included in  
342 ISO/IEC JTC 1 standards, along with many other algorithms submitted by other countries.  
343 The following is a list of ISO/IEC standards that include cryptographic algorithms and  
344 schemes specified in NIST standards.

- 345 1. [ISO/IEC 9797-1:2011](#), Information technology -- Security techniques -- Message  
346 Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher.
- 347 2. [ISO/IEC 9797-2:2011](#), Information technology -- Security techniques -- Message  
348 Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-  
349 function.
- 350 3. [ISO/IEC 10116:2006](#), Information technology -- Security techniques -- Modes of  
351 operation for an  $n$ -bit block cipher.
- 352 4. [ISO/IEC 10118-3:2004](#), Information technology -- Security techniques -- Hash-  
353 functions -- Part 3: Dedicated hash-functions.
- 354 5. [ISO/IEC 11770-3:2008](#), Information technology -- Security techniques -- Key  
355 management -- Part 3: Mechanisms using asymmetric techniques.
- 356 6. [ISO/IEC CD 11770-6](#) "Information technology -- Security techniques -- Key  
357 management -- Part 6: Key derivation.
- 358 7. [ISO/IEC 14888-2: 2008](#), Information technology -- Security techniques -- Digital  
359 signatures with appendix -- Part 2: Integer factorization based mechanisms.

---

<sup>22</sup> Further information is available at the ISO web site: [www.iso.org](http://www.iso.org).

- 360 8. [ISO/IEC CD 14888-3](#), Information technology -- Security techniques -- Digital  
361 signatures with appendix -- Part 3: Discrete logarithm based mechanisms.
- 362 9. [ISO/IEC 18033-3:2010](#), Information technology -- Security techniques --  
363 Encryption algorithms -- Part 3: Block ciphers.
- 364 10. [ISO/IEC 19772:2009](#), Information technology -- Security techniques --  
365 Authenticated encryption.

### 366 2.3.5 Trusted Computing Group (TCG)

367 The Trusted Computing Group (TCG) develops and promotes a set of industry standards  
368 that build upon roots of trust. Roots of Trust (RoTs) are hardware, firmware, and  
369 software components that are inherently trusted to perform specific, and vital, security  
370 functions. Because misbehavior by an RoT cannot be detected, they must be secure by  
371 design. To ensure that they are reliable and resistant to tampering, RoTs are often  
372 implemented in, or protected by, hardware.

373 Industry standards developed by the TCG define the capabilities of a set of fundamental  
374 roots of trust, and describe how to use those roots of trust in a variety of architectures and  
375 use cases. Many of the use cases supported by TCG technologies and specifications  
376 focus on one or more of the following areas: 1) device identity, 2) cryptographic key or  
377 credential storage, and 3) attestation of the system state.

378 Technologies supporting TCG-developed standards are deployed enterprise-class clients  
379 and servers, storage devices, embedded systems, and virtualized devices. Families of  
380 relevant TCG standards and specifications include:

- 381 • Trusted Platform Module (TPM): A TPM is a cryptographic module that can,  
382 among other features, establish device identity in a platform, provide secure  
383 storage for keys and credentials, and support the measurement and reporting of  
384 the system state. The TPM 2.0 Library Specification provides the general  
385 architecture and command set for TPMs, with platform-specific specifications  
386 detailing how a TPM can be implemented in a particular classes of systems.  
387 ISO/IEC JTC 1 has approved the TPM Library Specification as [ISO/IEC](#)  
388 [11889:2015 Parts 1-4](#).
- 389 • Trusted Network Connect (TNC): The TCG's TNC Work Group defines  
390 specifications that allow network administrators to enforce policies regarding  
391 endpoint integrity on devices connected to a network. These specifications were  
392 the basis for much of the work in the IETF's Network Endpoint Assessment  
393 (NEA) working group, and are highly complimentary to the on-going work in the  
394 IETF Security Automation and Continuous Monitoring (SACM) working group.
- 395 • Storage: The TCG's Storage Work Group defines specifications that enable  
396 standards-based mechanisms to protect data on storage devices, and manage these  
397 devices and capabilities. The TCG's storage specifications break out from a  
398 common core specification into two Security Subsystem Classes (SSCs): the Opal  
399 SSC, intended for client devices (e.g., tablets, notebooks, desktops), and the  
400 Enterprise SSC, intended for high-performance storage systems (e.g., servers).

401

### SECTION 3: CRYPTOGRAPHIC ALGORITHMS

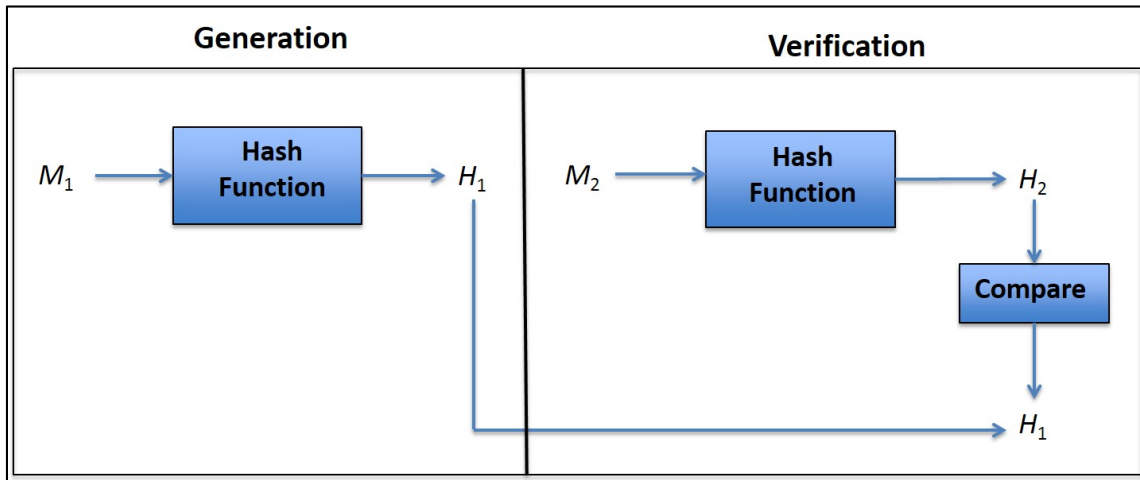
402 This document describes three types of cryptographic algorithms: cryptographic hash  
 403 functions, symmetric-key algorithms and asymmetric-key algorithms, discussed in  
 404 Sections 3.1, 3.2 and 3.3, respectively. Other topics to be introduced in this section include  
 405 the concept of algorithm security strength and algorithm lifetime (see Sections 3.4 and 3.5,  
 406 respectively).

#### 407 3.1 Cryptographic Hash Functions

408 A hash function (also called a hash algorithm) is a cryptographic primitive algorithm that  
 409 produces a condensed representation of its input (e.g., a message). A hash function takes  
 410 an input of arbitrary length and outputs a value with a predetermined length. Common  
 411 names for the output of a hash function include *hash value* and *message digest*.

412 A cryptographic hash function is a one-way function that is extremely difficult to invert.  
 413 That is, it is not practical to reverse the process from the hash value back to the input.

414 Figure 1 depicts the process of generating and verifying a hash value.



415

416 **Figure 1: Hash Function Generation and Verification**

417 A hash function is used as follows:

- 418 • Hash Generation:
  - 419 1. Hash value ( $H_1$ ) is generated on data ( $M_1$ ) using the hash function.
  - 420 2.  $M_1$  and  $H_1$  are then saved or transmitted.
- 421 • Hash Verification:
  - 422 1. Hash value ( $H_2$ ) is generated on the received or retrieved data ( $M_2$ ) using the  
 423 same hash function that generated  $H_1$ .
  - 424 2.  $H_1$  and  $H_2$  are compared. If  $H_1 = H_2$ , then it can be assumed that  $M_1$  has not  
 425 changed during storage or transmission.

426 The above description is for the simplest use of a hash function. Hash functions are usually  
427 used in higher-level algorithms, including:

- 428 • Keyed-hash message authentication code algorithms (Sections [3.2.2](#) and [4.2.2.2](#)),
- 429 • Digital signature algorithms ([Section 4.2.3](#)),
- 430 • Key derivation functions (e.g., for key establishment) ([Section 5.3.2](#)), and
- 431 • Random bit generators ([Section 4.4](#)).

432 When these higher-level algorithms are used with a key, they could be considered as  
433 symmetric-key algorithms (see [Section 3.2](#) for further discussion).

434 **Approved** hash functions for Federal Government use are specified in [FIPS 180](#)<sup>23</sup> and  
435 [FIPS 202](#)<sup>24</sup>.

- 436 • FIPS 180 specifies the SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-  
437 512/224 and SHA-512/256 hash functions. Additional guidance for the use of these  
438 hash functions is provided in [SP 800-106](#)<sup>25</sup> and [SP 800-107](#)<sup>26</sup>.

439 Note that attacks on SHA-1 have indicated that SHA-1 provides less security than  
440 originally thought when generating digital signatures (see [Section 4.2.3](#)) and is now  
441 disallowed for that purpose. However, SHA-1 may continue to be used for most  
442 other hash-function applications (see [SP 800-131A](#)<sup>27</sup>).

- 443 • [FIPS 202](#) specifies SHA3-224, SHA3-256, SHA3-384 and SHA3-512. This FIPS  
444 also specifies two extendable-output functions (SHAKE128 and SHAKE256),  
445 which are not, in themselves, considered to be hash functions; guidance on their  
446 use will be provided in the future.

## 447 **3.2 Symmetric-Key Algorithms**

448 Symmetric-key algorithms (often called secret-key algorithms) use a single key to both  
449 apply cryptographic protection and to remove or check the protection. For example, the  
450 key used to encrypt data (i.e., apply protection) is also used to decrypt the encrypted data  
451 (i.e., remove the protection); in the case of encryption, the original data is called the  
452 plaintext, while the encrypted form of the data is called the ciphertext. The key must be  
453 kept secret if the data is to remain protected.

454 Several classes of symmetric-key algorithms have been approved: those based on block  
455 cipher algorithms (e.g., AES) and those based on the use of hash functions (e.g., a keyed-  
456 hash message authentication code based on SHA-1).

---

<sup>23</sup> FIPS 180: Secure Hash Standard.

<sup>24</sup> FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable Output Functions.

<sup>25</sup> SP 800-106: Randomized Hashing for Digital Signatures.

<sup>26</sup> SP 800-107: Recommendations for Applications Using Approved Hash Algorithms.

<sup>27</sup> SP 800-131A: Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.

457 Symmetric-key algorithms are used for:

- 458 • Encryption to provide data confidentiality (see [Section 4.1](#)),
- 459 • Authentication to provide assurance of data integrity and the source of the data  
460 (see [Section 4.2](#)),
- 461 • Key derivation (see [Section 5.3.2](#)),
- 462 • Key wrapping (see [Section 5.3.5](#)), and
- 463 • Random bit generation (see [Section 4.4](#)).

464 When using a symmetric-key algorithm, a unique key needs to be generated for each  
465 cryptographic relationship<sup>28</sup> and used for each purpose (e.g., encryption, data integrity  
466 authentication and key wrapping). Technically, the same key can be used for multiple  
467 purposes when the same algorithm is used, but this is usually ill-advised, as the use of the  
468 same key for two different cryptographic processes (e.g., HMAC and key derivation using  
469 the same hash function) may weaken the security provided by one or both of the processes.  
470 However, exceptions to this rule have been approved (see [Section 4.3](#)).

471 As an example of the number of keys required for the use of symmetric-key algorithms,  
472 suppose that there are four entities (A, B, C, and D) that need to communicate using  
473 encryption, with each pair of entities using a different encryption key. There are six  
474 possible pair-wise relationships (A-B, A-C, A-D, B-C, B-D, and C-D), so, at least six keys  
475 are required<sup>29</sup>. If, instead, there are 1000 entities that wish to communicate with each other,  
476 there are 499,500 possible pair-wise relationships, and at least one unique key would be  
477 required for each relationship. If more than one algorithm, key length or purpose is to be  
478 supported (e.g., both encryption and key wrapping), then additional keys will be needed.  
479 Each entity must keep all its symmetric keys secret and protect their integrity. The  
480 requirement for a large number of keying relationships is a significant problem; methods  
481 for mitigating this problem are discussed in [Section 5](#).

482 Several symmetric-key algorithms have been approved by NIST for the protection of  
483 sensitive data. However, some of these algorithms are no longer approved for applying  
484 cryptographic protection (e.g., encryption), but may continue to be used for processing  
485 already-protected information (e.g., decryption), providing that the risk of doing so is  
486 acceptable (e.g., there is reason to believe that a key was not compromised). See [SP 800-  
487 57, Part 1](#) and [SP 800-131A](#) for more information about the acceptability of using different  
488 cryptographic algorithms.

### 489 3.2.1 Block Cipher Algorithms

490 A block cipher algorithm is used with a single key in an **approved** mode of operation to both apply  
491 cryptographic protection (e.g., encrypt) and to subsequently process the protected information (e.g.,

---

<sup>28</sup> A cryptographic relationship exists when two or more parties can communicate using the same key and algorithm. A relationship may be one-to-one or one-to-many (e.g., broadcast).

<sup>29</sup> Although only six cryptographic relationships are used in the example, different keys may be required by some protocols for each communication direction, i.e., a different key may be required for communications sent from A to B than is used for communications sent from B to A.

492 decrypt). Several block cipher algorithms have been approved by NIST as cryptographic primitives,  
493 some of which may no longer be approved for applying cryptographic protection. However, they  
494 may still be needed for processing information that was previously protected (e.g., they may be  
495 needed for decrypting previously encrypted information).

496 The block cipher algorithms are discussed in Sections [3.2.1.1](#) through [3.2.1.4](#). The  
497 **approved** modes of operation are discussed in [Section 3.2.1.5](#).

### 498 **3.2.1.1 Data Encryption Standard (DES)**

499 The Data Encryption Standard (DES) became effective in July 1977, and was the first  
500 NIST-**approved** cryptographic algorithm. It was reaffirmed several times, but due to  
501 advances in computer power and speeds, the strength of the DES algorithm is no longer  
502 sufficient to adequately protect Federal Government information. Therefore, DES was  
503 withdrawn as an **approved** algorithm in 2005 (i.e., the use of DES is no longer approved  
504 for encryption or otherwise applying cryptographic protection). However, the DES  
505 “cryptographic engine” continues to be used as a component function of TDEA (see the  
506 next section).

### 507 **3.2.1.2 Triple Data Encryption Algorithm (TDEA)**

508 The Triple Data Encryption Algorithm (TDEA), also known as Triple DES, uses the DES  
509 cryptographic engine to transform data in three operations. TDEA is specified in [SP 800-](#)  
510 [67](#)<sup>30</sup>.

511 TDEA encrypts data in blocks of 64 bits, using three keys that define a key bundle. The  
512 use of TDEA using three distinctly different (i.e., mathematically independent) keys is  
513 **approved** and is commonly known as three-key TDEA (also referred to as 3TDEA or  
514 3TDES).

515 Other variations of TDEA, where two or three of the keys are identical, are no longer  
516 approved for applying cryptographic protection because of increased computing power or  
517 weaknesses in the algorithm.

### 518 **3.2.1.3 SKIPJACK**

519 SKIPJACK is referenced in [FIPS 185](#)<sup>31</sup> and specified in a classified document. SKIPJACK  
520 is no longer considered adequate for the protection of federal information and has been  
521 withdrawn as a FIPS. The use of SKIPJACK for applying cryptographic protection (e.g.,  
522 encryption) is **not approved**, although it is permissible to use the algorithm for decrypting  
523 information.

---

<sup>30</sup> SP 800-67: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher.

<sup>31</sup> FIPS 185: Escrowed Encryption Standard.



#### 524 **3.2.1.4 Advanced Encryption Standard (AES)**

525 The Advanced Encryption Standard (AES) was developed as a replacement for DES and  
526 is the preferred block cipher algorithm for new products. AES is specified in [FIPS 197](#)<sup>32</sup>.  
527 AES operates on 128-bit blocks of data, using 128, 192 or 256-bit keys.

528 Note that the use of the longer key lengths affects algorithm performance (e.g., the speed),  
529 though not by very much. Also, note that the performance of AES is significantly better  
530 than that of TDEA.

#### 531 **3.2.1.5 Modes of Operation**

532 With a symmetric-key block cipher algorithm, the same input block will always produce  
533 the same output block when the same key is used. For example, if the multiple blocks in a  
534 typical message are encrypted without using a mode designed for the purpose, an adversary  
535 could easily substitute individual blocks, possibly without detection. Furthermore, certain  
536 kinds of data patterns in the plaintext, such as repeated blocks, would be apparent in the  
537 ciphertext.

538 Therefore, block cipher modes-of-operation have been specified to address this problem  
539 by combining the cryptographic primitive algorithm with variable starting values  
540 (commonly known as initialization vectors) and rules that successively use the block cipher  
541 algorithm to perform a cryptographic service (e.g., the encryption of a message).  
542 **Approved** modes for block cipher algorithms have been specified in the [SP 800-38](#) series  
543 of publications and include modes for:

- 544 • Encryption, as specified in [SP 800-38A](#), [SP 800-38E](#) and [SP 800-38G](#) (see [Section](#)  
545 [4.1](#)),
- 546 • Authentication, as specified in [SP 800-38B](#) (see [Section 4.2.2.1](#)),
- 547 • Authenticated encryption, as specified in [SP 800-38C](#) and [SP 800-38D](#) (see [Section](#)  
548 [4.3](#)), and
- 549 • Key wrapping, as specified in [SP 800-38F](#) (see [Section 5.3.5](#)).

#### 550 **3.2.2 Hash-based Symmetric-key Algorithms**

551 A symmetric-key algorithm based on the use of a hash function has been specified in [FIPS](#)  
552 [198](#)<sup>33</sup>. This algorithm, known as HMAC, has been **approved** for use with any **approved**  
553 hash function specified in [FIPS 180](#) or [FIPS 202](#). Guidance on the use of the hash functions  
554 specified in FIPS 180 for HMAC is provided in [SP 800-107](#).

### 555 **3.3 Asymmetric-Key Algorithms**

556 Asymmetric-key algorithms (often called public-key algorithms) use a pair of keys (i.e., a  
557 key pair): a public key and a private key that are mathematically related to each other. The  
558 public key may be made public without reducing the security of the process, but the private

---

<sup>32</sup> FIPS 197: Advanced Encryption Standard.

<sup>33</sup> FIPS 198: Keyed Hash Message Authentication Code (HMAC).

559 key must remain secret if the data is to retain its cryptographic protection. Even though  
560 there is a relationship between the two keys, the private key cannot easily be determined  
561 based on knowledge of the public key.

562 One of the keys of the key pair is used to apply cryptographic protection, and the other key  
563 is used to remove or verify that protection. The key to use depends on the algorithm used  
564 and the service to be provided. For example, a digital signature is computed using a private  
565 key, and the signature is verified using the public key (i.e., the protection is applied using  
566 the private key and verified using the corresponding public key). For those asymmetric  
567 algorithms also capable of encryption<sup>34</sup>, the encryption is performed using the public key,  
568 and the decryption is performed using the private key (i.e., the protection is applied using  
569 the public key and removed using the private key).

570 Asymmetric-key algorithms are used primarily for data integrity authentication and source  
571 authentication (see [Section 4.2](#)), and for key establishment (see [Section 5.3](#)). These  
572 algorithms tend to be much slower than symmetric-key algorithms, so are not used to  
573 process large amounts of data. However, when used for key establishment (see [Section 5](#)),  
574 there are methods that combine the use of symmetric and asymmetric algorithms to reduce  
575 the number of keys required for establishing cryptographic relationships.

576 Like symmetric-key algorithms, the key pair for an asymmetric-key should be generated  
577 for each purpose (e.g., one key pair for generating and verifying digital signatures, and a  
578 different key pair for key establishment). Technically, it is sometimes possible to use the  
579 same key pair for more than one purpose, but this is ill-advised, as the use of the same key  
580 pair for two different cryptographic purposes (e.g., digital signatures and key  
581 establishment) may weaken the security provided by one or both of the processes.

582 The use of asymmetric-key algorithms requires the establishment of fewer initial keys than  
583 the use of symmetric-key algorithms. As an example, suppose that an entity wants to  
584 generate digital signatures and participate in a key-establishment process using its own key  
585 pair<sup>35</sup>; a key pair needs to be generated for each purpose. If there are six entities that intend  
586 to both generate digital signatures and participate in the key-establishment process, then  
587 six key pairs are needed for digital signature generation, and another six key pairs are  
588 needed for key establishment, for a total of twelve key pairs. For 1000 entities, 1000 key  
589 pairs of each would be needed for each purpose, for a total of 2000 key pairs. A unique key  
590 pair does not need to be generated for each relationship; recall that for symmetric-key  
591 algorithms, a unique key needs to be generated for each relationship (see [Section 3.2](#)). If  
592 multiple public-key algorithms or key lengths are to be used for either process, then  
593 additional key pairs will be required.

594 The private key is retained by the entity who “owns” the key pair; it must be kept secret  
595 and its integrity protected. The public key is usually distributed to other entities and  
596 requires integrity protection; this is often accomplished by using a public-key certificate,

---

<sup>34</sup> Not all public-key algorithms are capable of multiple functions, e.g., both encryption and decryption, and the generation and verification of digital signatures.

<sup>35</sup> Note that some key-establishment schemes do not require that all parties have key pairs, so some parties will not need a key pair for key establishment.



597 as discussed in [Section 5.2.3](#). When a public-key certificate is used, the certificate provides  
598 the integrity protection for the public key, so the burden of key protection by each entity is  
599 limited to only those private keys owned by the entity.

600 Some asymmetric-key algorithms use domain parameters, which are additional values  
601 necessary for the use of the cryptographic algorithm. These values are mathematically  
602 related to each other and to the keys with which they will be used. Domain parameters are  
603 usually public and are used by a community of users for a substantial period of time. These  
604 domain parameters are either contained within or referenced by a certificate containing a  
605 public key.

606 The secure use of asymmetric-key algorithms is dependent on users obtaining certain  
607 assurances:

- 608 • Assurance of domain-parameter validity (for those algorithms requiring domain  
609 parameters) provides confidence that the domain parameters are mathematically  
610 correct,
- 611 • Assurance of public-key validity provides confidence that the public key appears  
612 to be a suitable key, and
- 613 • Assurance of private-key possession provides confidence that the party that is  
614 supposedly the owner of the private key really has the key.

### 615 **3.3.1 DSA**

616 The Digital Signature Algorithm (DSA) is **approved** and specified in [FIPS 186](#). This  
617 algorithm is used to generate and verify digital signatures using finite-field mathematics  
618 (i.e., the mathematics that most of us are familiar with). FIPS 186 defines methods for  
619 generating DSA domain parameters and key pairs, and specifies the key lengths to be used  
620 for secure interoperability and the algorithms to be used for digital-signature generation  
621 and verification.

### 622 **3.3.2 ECDSA**

623 The Elliptic Curve Digital Signature Algorithm (ECDSA) is **approved** within [FIPS 186](#),  
624 but actually specified within American National Standard (ANS) [X9.62](#)<sup>36</sup>. The basic  
625 signature and verification algorithms are the same as those used for DSA, except that the  
626 mathematics is based on the use of elliptic curves, rather than finite fields (i.e., the rules  
627 for combining numbers is different than commonly used). FIPS 186 provides guidance for  
628 the use of ECDSA within the Federal Government, as well as providing recommended  
629 elliptic curves to facilitate interoperability and security. An advantage of using ECDSA is  
630 that the key lengths are considerably shorter than those used for DSA and RSA, requiring  
631 less storage space and transmission bandwidth, and the execution of the algorithm is  
632 generally faster than DSA and RSA

633 [ANS X9.62](#) includes specifications for the generation of the ECDSA domain parameters  
634 and key pairs, as well as the algorithms for digital signature generation and verification.

---

<sup>36</sup> ANS X9.62: Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA).

635 [FIPS 186](#) defines the key lengths to be used for secure interoperability, provides additional  
636 guidance on the use of random bit generators to generate the key pairs, and recommends  
637 elliptic curves for use by the Federal Government. Note that the same elliptic curves are  
638 also included in ANS X9.62.

### 639 **3.3.3 RSA**

640 The RSA algorithm is **approved** for the generation and verification of digital signatures in  
641 [FIPS 186](#)] and specified in [PKCS 1](#)<sup>37</sup> and [ANS X9.31](#)<sup>38</sup>. FIPS 186 includes restrictions on  
642 the use of RSA to generate digital signatures, methods to generate RSA key pairs, and  
643 defines the key lengths to be used for secure interoperability.

644 The RSA primitive can be used for key establishment, as well as for the generation and  
645 verification of digital signatures. Its use for key establishment is specified in [SP 800-](#)  
646 [56B](#)<sup>39</sup>; that publication specifies **approved** methods for both key agreement and key  
647 transport (see [Section 5.3](#) for further information on key establishment, key agreement and  
648 key transport).

649 The key pairs used for RSA digital-signature generation and verification, and for RSA key  
650 establishment are generated in the same way, but need to be different for each purpose.

### 651 **3.3.4 Diffie-Hellman and MQV**

652 Diffie-Hellman (DH) and MQV<sup>40</sup> are two classes of key-establishment algorithms used for  
653 key agreement (see [Section 5.3.3](#)). The use of these algorithms for key agreement is  
654 specified in [SP 800-56A](#)<sup>41</sup> using both finite-field and elliptic-curve mathematics for each.  
655 For elliptic-curve key pairs and domain parameters, the methods for generating those key  
656 pairs and domain parameters are specified in ANS [X9.62](#) using the same methods used to  
657 generate ECDSA key pairs and domain parameters.

658

659 The recommended elliptic curves for elliptic-curve DH and MQV are the same as those  
660 provided in [FIPS 186](#) for ECDSA.

## 661 **3.4 Algorithm Security Strength**

662 The security strength of a cryptographic algorithm is measured by an attacker's difficulty  
663 in breaking the algorithm. Breaking a cryptographic algorithm can be defined as defeating  
664 some aspect of the protection that the algorithm is intended to provide. For example, a

---

<sup>37</sup> Public Key Cryptography Standard #1.

<sup>38</sup> ANS X9.31, Digital Signatures Using Reversible Public Key Cryptography For The Financial Services Industry (RDSA). This standard has been withdrawn as an ANSI standard.

<sup>39</sup> SP 800-56B: Recommendation for Pair-wise Key Establishment Schemes Using Integer Factorization Cryptography.

<sup>40</sup> Menezes–Qu–Vanstone

<sup>41</sup> SP 800-56A: Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography.

665 block cipher encryption algorithm that is used to protect the confidentiality of data is  
666 broken if, with an acceptable amount of work, it is possible to determine the value of its  
667 key or to recover the plaintext from the ciphertext without knowledge of the key.

668 [SP 800-57, Part 1](#) provides the current estimates for the security strengths that can be  
669 provided by the **approved** cryptographic algorithms; these strengths have been determined  
670 with respect to specific key lengths.

671 The **approved** security strengths for federal applications are 112, 128, 192 and 256 bits.  
672 Appropriate algorithms, key lengths, and key generation and handling methods need to be  
673 used to actually support those security strengths, and is further discussed in [Section 5.1.4](#).

### 674 **3.5 Algorithm Lifetime**

675 Over time, algorithms may be successfully attacked so that the algorithm no longer  
676 provides the desired protection. The attack could be on the algorithm itself, or could be on  
677 the algorithm with a specific key length. In the latter case, the use of a longer key may  
678 prevent a successful attack, or at least delay it for a period of time.

679 When selecting the algorithms and key lengths to be used for an application, the length of  
680 time for which the data needs to be protected should be taken into account so that a suitable  
681 algorithm and key length is used. [SP 800-57, Part 1](#) provides a current estimate of the time  
682 frames during which the **approved** algorithms and key lengths are considered to be secure.  
683 The algorithms and key lengths used for cryptographic protection need to fall within the  
684 estimated time frame. However, these estimates are just that – estimates. It is possible that  
685 an advance in technology or cryptanalysis could occur prior to the end date of that time  
686 frame (e.g., the use of quantum computers and algorithms). It is often the case that these  
687 advances are initially impractical or limited in their threat. It is recommended that an  
688 organization have a transition strategy for addressing this problem if it occurs, including  
689 assessing the risk for the compromise of the organization's data, and transitioning to a new  
690 algorithm or key length, as appropriate.

691

692

693

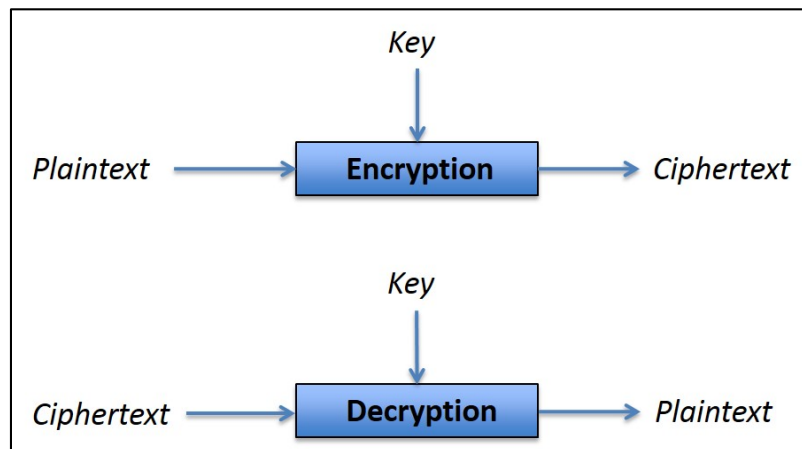
## SECTION 4: CRYPTOGRAPHIC SERVICES

694 All sensitive information requires integrity protection, and confidentiality protection may  
695 be required as well. This section discusses the cryptographic services that can be provided  
696 for the protection of sensitive data other than keys. These services include data  
697 confidentiality, data integrity authentication and source authentication, including non-  
698 repudiation. The protection and management of the keys used while providing these  
699 cryptographic services are discussed in [Section 5](#).

700 Ideally, cryptographic services would be provided using as few algorithms as possible. For  
701 example, AES could be used to provide confidentiality ([Section 4.1](#)), data integrity  
702 authentication ([Section 4.2](#)), key wrapping ([Section 5.3.5](#)) and as the basis for a random bit  
703 generator (see [Section 4.4](#)). However, this may not be as practical as it first appears, as  
704 other algorithms may also be available that are needed for different applications and that  
705 provide other security properties.

### 706 4.1 Data Confidentiality

707 Encryption is used to provide confidentiality for data. The unprotected form of the data is  
708 called plaintext. Encryption transforms the data into ciphertext, and ciphertext can be  
709 transformed back into plaintext using decryption. Data encryption and decryption are  
710 provided using symmetric-key block cipher algorithms. The **approved** symmetric-key  
711 algorithms for data encryption are: AES and TDEA (see [Section 3.2.1.4](#) and [Section](#)  
712 [3.2.1.2](#), respectively). Decryption of the ciphertext is performed using the algorithm and  
713 key that were used to encrypt the plaintext. Unauthorized recipients of the ciphertext who  
714 know the cryptographic algorithm but do not have the correct key should not be able to  
715 decrypt the ciphertext. However, anyone who has the key and the cryptographic algorithm  
716 can easily decrypt the ciphertext and obtain the original plaintext.



717

718

**Figure 2: Encryption and Decryption**

719 Figure 2 depicts the encryption and decryption processes. The plaintext and a key are used  
720 by the encryption process to produce the ciphertext. To decrypt, the ciphertext and the same  
721 key are used by the decryption process to recover the plaintext data.

722 Note that asymmetric-key algorithms could also be used to encrypt and decrypt data, but  
723 because these algorithms are slow in comparison to block cipher algorithms, they are not  
724 normally used to encrypt and decrypt general data; they can, however, be used to protect  
725 keys, as discussed in [Section 5](#).

726 As discussed in [Section 3.2.1.5](#), encryption is performed using a block cipher algorithm  
727 and a mode of operation. The **approved** modes of operation for encryption are specified  
728 in:

- 729 • [SP 800-38A](#) for AES and TDEA: the Electronic Codebook (ECB), Cipher Block  
730 Chaining (CBC), Cipher Feedback (CFB), Counter (CTR), and Output Feedback  
731 (OFB) modes,
- 732 • [SP 800-38E](#) for AES: the XTS-AES mode (for protecting the confidentiality of  
733 data on storage devices only), and
- 734 • [SP 800-38G](#) for AES: the FF1 and FF3 modes for Format Preserving Encryption.

735 Additional modes that provide both confidentiality and authentication (as discussed in  
736 [Section 4.2](#)) are discussed in [Section 4.3](#).

## 737 **4.2 Data Integrity and Source Authentication**

738 Data integrity (often referred to as simply *integrity*) is concerned with whether or not  
739 something (e.g., some data) has changed between two specified times (e.g., between the  
740 time when the data was created, stored and/or transmitted, and the time when it was  
741 retrieved and/or received). The absolute integrity of the data cannot be guaranteed, but the  
742 computation of a data integrity code on the data when it is created, before storage or before  
743 transmission will allow the detection of any changes with a high probability when that code  
744 is later verified, thus providing a measure of assurance of data integrity. In cryptographic  
745 literature, this process is called *message* (or data) *authentication*.

746 Source authentication is a process used to provide assurance of the source of information.  
747 Source authentication includes identity authentication, which provides assurance to one of  
748 the parties in a communication (say, Bob) that he is receiving data from or providing data  
749 to another specific party (say, Alice). Depending on the method used, source authentication  
750 could also support non-repudiation, whereby both Bob and some third party (say, Carl)  
751 have some assurance that the data came from Alice.

752 Cryptography can be used to provide these services, but the same algorithm may not  
753 provide all of them. Hash functions, as discussed in [Section 4.2.1](#), can be used to provide  
754 some assurance of data integrity. Message Authentication Code (MAC) algorithms, as  
755 discussed in [Section 4.2.2](#), can provide both data integrity and source authentication  
756 services. Digital signature algorithms can be used to provide data integrity and source  
757 authentication services, as well as supporting non-repudiation, but at a higher performance  
758 cost (see [Section 4.2.3](#)).

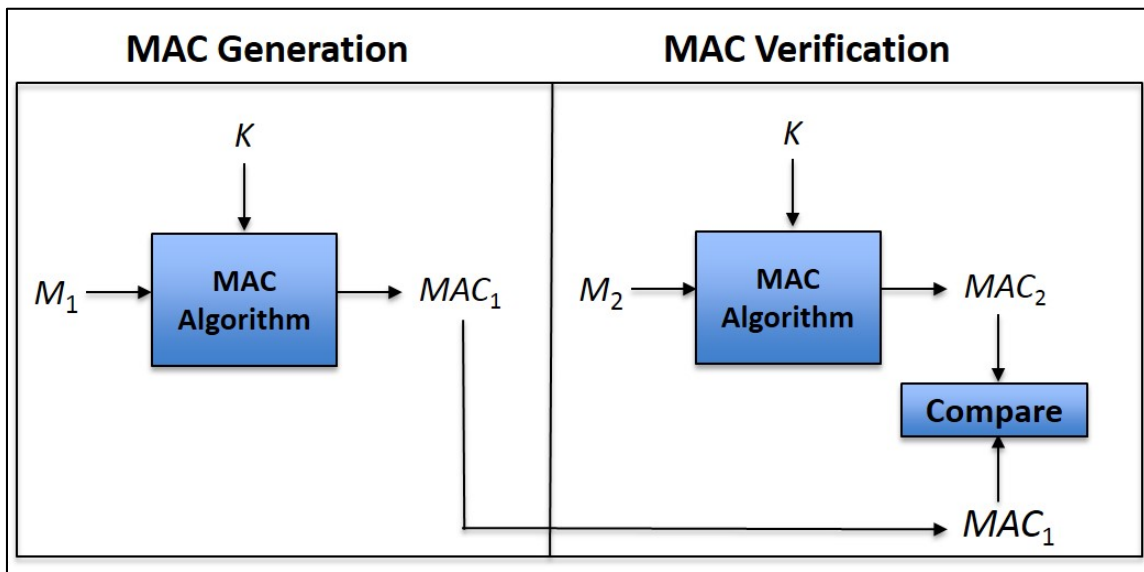
### 759 **4.2.1 Hash Functions**

760 A hash function is used to generate a hash value that can provide some assurance of the  
761 integrity of the data over which the hash value is generated. However, since no

762 cryptographic key is used, there is no assurance that the data has not been altered by an  
 763 adversary and a new hash value computed. This method for providing integrity protection  
 764 is not recommended unless there is a very low risk of this scenario (e.g., when data is  
 765 provided by a trusted source, and the hash value is used to determine changes that may  
 766 occur because of a degraded transmission medium).

767 **4.2.2 Message Authentication Code Algorithms**

768 A Message Authentication Code algorithm and a cryptographic key are used to generate a  
 769 message authentication code (MAC) that can be used to provide assurance of data integrity  
 770 and source authentication. A MAC is a cryptographic checksum on the data that can  
 771 provide assurance that the data has not changed or been altered since some point in time,  
 772 and that the MAC was computed by the party or parties sharing the key. Typically, MACs  
 773 are used between two or more parties that share the same secret key to authenticate  
 774 information exchanged between those parties; the use of MACs to provide data integrity  
 775 and source authentication depends on limiting knowledge of the secret key to only those  
 776 parties. Since a MAC key is shared among a community of users (e.g., two or more parties),  
 777 only those parties sharing the key can compute a correct MAC on given data.



778 **Figure 3: Message Authentication and Verification**

779 Figure 3 depicts the use of MACs:

- 780
- 781 • A MAC ( $MAC_1$ ) is computed on data ( $M_1$ ) using a key ( $K$ ).  $M_1$  and  $MAC_1$  are then
  - 782 saved or transmitted.
  - 783 • At a later time, the integrity of the retrieved or received data is checked by labeling
  - 784 the retrieved or received data as  $M_2$  and computing a MAC ( $MAC_2$ ) on  $M_2$  using
  - 785 the same key ( $K$ ).
  - 786 • If  $MAC_1$  is the same as  $MAC_2$ , then it can be assumed that  $M_2$  is the same as the
  - 787 original data ( $M_1$ ) (i.e.,  $M_1 = M_2$ ). The verifying party also knows that only a party
  - 788 that shares the key could have correctly generated the MAC.



789 For example, if two parties (e.g., parties A and B) share a key, party A generates the MAC  
790 and sends it to party B, and party B successfully verifies the received MAC, then party B  
791 knows that party A generated the original MAC, and source authentication has been  
792 accomplished. However, if three parties share the key (e.g., A, B and C), party A generates  
793 the MAC to be sent to party B, and party B successfully verifies the received MAC; party  
794 B knows that either party A or party C generated the original MAC, but has no proof of  
795 which one. Note that this may be acceptable for some applications.

796 MACs are used to detect data modifications that occur between the initial generation of the  
797 MAC and the verification of the received MAC. They do not detect errors that occur before  
798 the MAC is originally generated.

799 Assurance of data integrity is frequently provided using non-cryptographic techniques  
800 known as error detection codes. However, these codes can be altered by an adversary to  
801 the adversary's benefit. The use of an **approved** cryptographic mechanism, such as a  
802 MAC, addresses this problem. That is, the assurance of integrity provided by a MAC is  
803 based on the assumption that it is not likely that anyone could correctly generate a MAC  
804 without knowing the cryptographic key. An adversary without knowledge of the key will  
805 be unable to modify data and then generate a verifiable MAC on the modified data. It is  
806 therefore crucial that MAC keys be kept secret.

807 Two types of algorithms for computing a MAC have been **approved** for Federal  
808 Government use: MAC algorithms that are based on symmetric-key block cipher  
809 algorithms, and MAC algorithms that are based on hash functions.

#### 810 4.2.2.1 MACs Based on Block Cipher Algorithms

811 The SP 800-38 series of publications includes modes for the generation of MACs:

- 812 • [SP 800-38B](#)<sup>42</sup> defines the CMAC mode for computing a MAC using the NIST-  
813 **approved** block-cipher algorithms: AES and TDEA.
- 814 • [SP 800-38D](#)<sup>43</sup> defines the GMAC mode for the computation of a MAC using AES.
- 815 • Modes providing both confidentiality (i.e., encryption) and authentication (i.e.,  
816 computing a MAC) in a single operation are also defined (see [Section 4.3](#)).

#### 817 4.2.2.2 MACs Based on Hash Functions

818 [FIPS 198](#)<sup>44</sup> defines a MAC (HMAC) that uses a cryptographic hash function in  
819 combination with a secret key. HMAC must be used with an **approved** cryptographic hash

---

<sup>42</sup> SP 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication.

<sup>43</sup> SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.

<sup>44</sup> FIPS 198-1: The Keyed-Hash Message Authentication Code (HMAC).

820 function (see [Section 4.2.1](#)). The security associated with the use of HMAC is discussed in  
821 [SP 800-107](#)<sup>45</sup>.

### 822 **4.2.3 Digital Signature Algorithms**

823 A digital signature algorithm is used with a pair of keys – a private key and a public key –  
824 to generate and verify digital signatures. The private key is used to generate signatures and  
825 must be known only by the signer (the key-pair owner); the public key is used to verify the  
826 signatures. Because of the design of the algorithm, and the methods for generating key  
827 pairs, the public key cannot easily be used to determine the private key. Because two keys  
828 are required for the generation and verification process, digital signature algorithms are  
829 classified as asymmetric-key algorithms.

830 A digital signature is represented in a computer as a string of bits and is an electronic  
831 analogue of a hand-written signature that can be verified by anyone with access to the  
832 public key. The signature can be used to provide assurance of data integrity and source  
833 authentication, and to support non-repudiation.

834 Each signer possesses a private and public key pair. Signature generation (with a verifiable  
835 digital signature) can be performed only by the party that has access to the private key.  
836 Anyone that knows the public key can verify the signature by employing the associated  
837 public key. The security of a digital-signature system is dependent on maintaining the  
838 secrecy of the signer's private key. Therefore, signers must guard against the unauthorized  
839 acquisition of their private keys.

840 Digital signatures offer protection that is not available by using alternative signature  
841 techniques. One such alternative is a digitized signature. A digitized signature is generated  
842 by converting a visual form of a handwritten signature to an electronic image (e.g., by  
843 scanning it into a computer). Although a digitized signature resembles its handwritten  
844 counterpart when printed, it does not provide the same protection as a digital signature.  
845 Digitized signatures can be forged and can be duplicated and appended to other electronic  
846 data; digitized signatures cannot be used to determine if information has been altered after  
847 it is signed. Digital signatures, however, are computed on each message using a private key  
848 known only by the signer. Each different message signed by the signer will have a different  
849 digital signature. Even small changes to the message will result in a different signature. If  
850 an adversary does not know the private key, the adversary cannot generate a valid signature  
851 (i.e., a signature that can be verified using the public key that corresponds to the private  
852 key used to generate the signature).

853 Figure 4 depicts the generation and verification of digital signatures. A digital signature  
854 algorithm includes a signature generation process and a signature verification process:

- 855 • Signature generation:
  - 856 ○ A hash function (see [Section 3.1](#)) is used in the signature generation process to
  - 857 obtain a hash value, which is a condensed version of the data to be signed (i.e.,
  - 858 shown as  $M_1$  for signature generation in Figure 4).

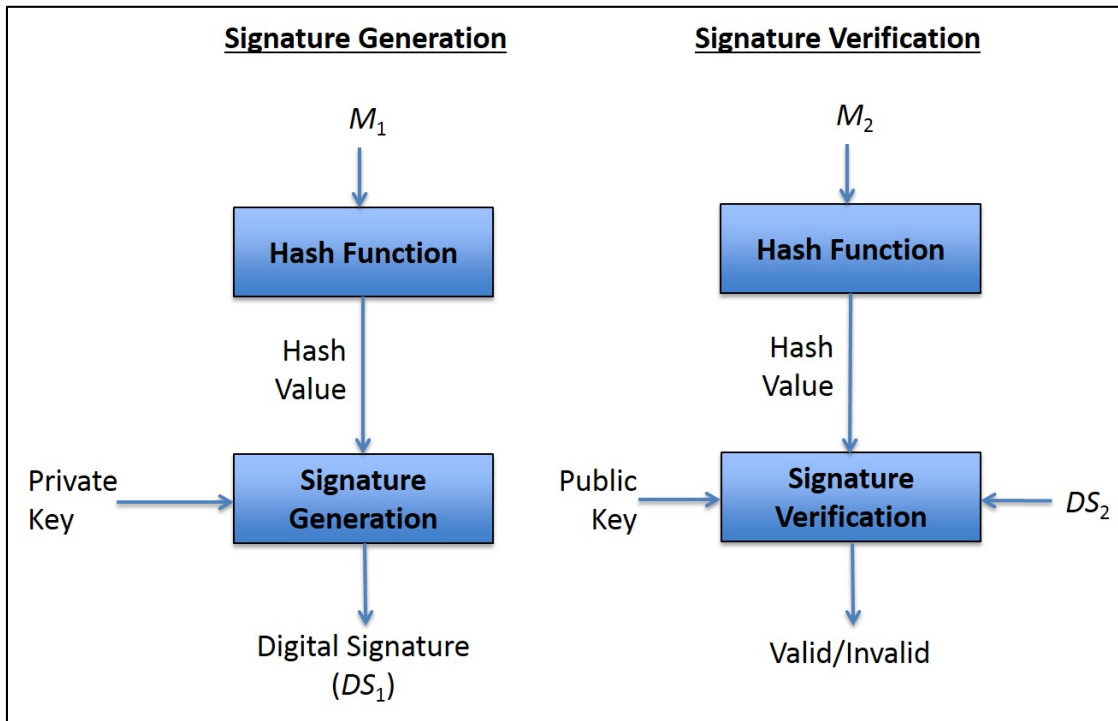
---

<sup>45</sup> SP 800-107: Recommendation for Applications Using Approved Hash Algorithms.



- 859           ○ The hash value is then input to the signature generation process, along with a
- 860           private key, to generate the digital signature (shown as  $DS_1$  in Figure 4).
- 861           ○ The digital signature ( $DS_1$ ) is provided to the verifier, along with the signed data
- 862           ( $M_1$ ).
- 863           • **Signature verification:** The receiver of the data and signature verifies the signature
- 864           as follows using the signatory's public key to process the received signature:
  - 865           ○ The received data ( $M_2$ ) is hashed using the same hash function to produce
  - 866           another hash value.
  - 867           ○ The newly computed hash value and the received signature ( $DS_2$ ) are input to
  - 868           the signature verification process, along with the the signer's public key. The
  - 869           output of this process is an indication of whether or not the signature is valid or
  - 870           invalid for the received message ( $M_2$ ).

871 Note that the details of the signature generation and verification processes are different for  
 872 each approved algorithm. Also, note that  $M_2$  is used in the verification process rather than  
 873  $M_1$ , and  $D_2$  is used rather than  $D_1$  because of the possibility that  $M_1$  and  $D_1$  could have been  
 874 deliberately or accidentally modified before the verification process performed by the  
 875 receiver.



876

877

**Figure 4: Digital Signature Generation and Verification**

878 [FIPS 186](#) specifies methods for generating and verifying digital signatures using  
 879 asymmetric (public-key) cryptography. The FIPS includes three digital signature  
 880 algorithms:

881

- The Digital Signature Algorithm (DSA) (see [Section 3.3.1](#)),

- 882       • The Elliptic Curve Digital Signature Algorithm (ECDSA) (see [Section 3.3.2](#)), and  
883       • RSA (see [Section 3.3.3](#)).

884 The digital signature algorithms are used in conjunction with the hash functions specified  
885 in [FIPS 180](#)<sup>46</sup> and [FIPS 202](#). Each of these algorithms requires obtaining assurances about  
886 the domain parameters and/or keys used, as discussed in [Section 3.3](#); [SP 800-89](#)<sup>47</sup> provides  
887 methods for obtaining these required assurances when using digital signatures.

888 In many cases, determining when a digital signature was generated is important. For  
889 example, it may be important to determine whether a document was signed before a certain  
890 date, e.g., which of two wills was signed closest to and prior to the date that a person died.  
891 [SP 800-102](#)<sup>48</sup> provides guidance on establishing when a digital signature was generated.

### 892 **4.3 Combining Confidentiality and Authentication in a Block-Cipher Mode** 893 **of Operation**

894 Confidentiality and authentication can be provided using either two separate block-cipher  
895 algorithms (e.g., AES in the CBC mode for encryption and HMAC for authentication) or  
896 in a single block-cipher mode of operation. Note that in this discussion, authentication is  
897 used to obtain both an assurance of data integrity and of the source of the data that has been  
898 cryptographically protected.

899 If encryption and authentication are performed as two separate operations (see Sections [4.1](#)  
900 and [4.2](#), respectively), two distinct keys are required. If care is not taken in performing  
901 these operations (e.g., performing the operations in the right order), vulnerabilities can be  
902 introduced that may allow attacks.

903 An alternative is to use modes that both encrypt and authenticate in a single operation using  
904 a single key; such a mode is called an “authenticated-encryption” mode. Using such modes  
905 requires fewer keys and is generally faster than using two separate operations. Two  
906 authenticated-encryption modes have been defined for AES (no such mode has been  
907 defined for TDEA):

- 908       • [SP 800-38C](#)<sup>49</sup> specifies the CCM mode, and  
909       • [SP 800-38D](#)<sup>50</sup> defines the Galois/Counter mode (GCM).

### 910 **4.4 Random Bit Generation**

911 Cryptography and security applications make extensive use of random numbers and  
912 random bits. For cryptography, random values are needed to generate cryptographic keys.

---

<sup>46</sup> FIPS 180: Secure Hash Standard.

<sup>47</sup> SP 800-89: Recommendation for Obtaining Assurances for Digital Signature Applications.

<sup>48</sup> SP 800-102: Recommendation for Digital Signature Timeliness.

<sup>49</sup> SP 800-38C: Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality.

<sup>50</sup> SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.

913 The term “entropy” is used to describe the amount of randomness in a value, and the  
914 amount of entropy determines how hard it is to guess that value.

915 There are two classes of random bit generators (RBGs): Non-Deterministic Random Bit  
916 Generators (NRBGs), sometimes called true random number (or bit) generators, and  
917 Deterministic Random Bit Generators (DRBGs), sometimes called pseudorandom bit (or  
918 number) generators. Each RBG is dependent on the use of an entropy source to provide  
919 unpredictable bits that are outside of human control; these bits are acquired from some  
920 physical source, such as thermal noise, ring oscillators or hard-drive seek times. An NRBG  
921 is dependent on the availability of new, unused entropy bits produced by the entropy source  
922 for every NRBG output. A DRBG is initially “seeded” with entropy produced by an  
923 entropy source or using an **approved** method that depends on an entropy source (e.g., an  
924 NRBG); depending on the application, the DRBG may or may not receive additional  
925 entropy (e.g., by being reseeded).

926 Several publications have been developed or are currently under development for random-  
927 bit generation:

- 928 • [SP 800-90A](#)<sup>51</sup> specifies **approved** DRBG algorithms, based on the use of hash  
929 functions and block-cipher algorithms; DRBGs must be initialized from a  
930 randomness source that provides sufficient entropy for the security strength(s) to  
931 be supported by the DRBG.
- 932 • [SP 800-90B](#)<sup>52</sup>, which is currently under development, discusses entropy sources,  
933 including health tests to determine that the entropy source has not failed and tests  
934 to estimate how much entropy that the entropy source can provide reliably.
- 935 • [SP 800-90C](#)<sup>53</sup> provides constructions for the design and implementation of NRBGs  
936 and DRBGs from the algorithms in SP 800-90A and the entropy sources designed  
937 in accordance with SP 800-90B. Note that the NRBGs are constructed to include a  
938 DRBG algorithm from SP 800-90A to provide a fallback capability if an entropy  
939 source failure is not immediately detected.
- 940 • [SP 800-22](#)<sup>54</sup> discusses some aspects of selecting and testing random and  
941 pseudorandom number generators. This document includes some criteria for  
942 characterizing and selecting appropriate generators, discusses statistical testing and  
943 its relation to cryptanalysis and provides some recommended statistical tests. These  
944 tests may be useful as a first step in determining whether or not a generator is  
945 suitable for a particular cryptographic application. However, for federal  
946 applications, the RBGs must be validated for compliance to [FIPS 140](#) and the  
947 appropriate parts of SP 800-90.

---

<sup>51</sup> SP 800-90A: Random Number Generation Using Deterministic Random Bit Generator Mechanisms.

<sup>52</sup> SP 800-90B: Entropy Sources.

<sup>53</sup> SP 800-90C: Random Bit Generator (RBG) Constructions.

<sup>54</sup> SP 800-22: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.

#### 948 **4.5 Symmetric vs. Asymmetric Cryptography**

949 As discussed in Sections [3.2](#) and [3.3](#), when large numbers of cryptographic relationships  
950 are required, the number of initial symmetric keys that will be required may be significantly  
951 larger than the number of public/private key pairs required.

952 However, the primary advantage of symmetric-key cryptography is speed. Symmetric-key  
953 algorithms are generally significantly faster than asymmetric-key algorithms, and the keys  
954 are shorter in length for the same security strength; the key length may be an important  
955 consideration if memory for storing the keys, or the bandwidth for transporting the keys is  
956 limited. In addition, advances in cryptanalysis and computational efficiency have tended  
957 to reduce the level of protection provided by public-key cryptography more rapidly than  
958 that provided by symmetric-key cryptography. Also, in a potential post-quantum “world”,  
959 the currently approved asymmetric-key algorithms will not provide adequate protection.

960 Since asymmetric-key (i.e., public-key) cryptography requires fewer keys overall, and  
961 symmetric-key cryptography is significantly faster, a hybrid approach is often used,  
962 whereby asymmetric-key algorithms are used for the generation and verification of digital  
963 signatures and for key establishment, while symmetric-key algorithms are used for all other  
964 purposes (e.g., encryption), especially those involving the protection of large amounts of  
965 data. For example, an asymmetric-key system can be used to establish a symmetric key via  
966 a key-agreement or key-transport process (see Sections [5.3.3](#) and [5.3.4](#), respectively), after  
967 which the symmetric key is used to encrypt files or messages.

968 In some situations, asymmetric-key cryptography is not necessary, and symmetric-key  
969 cryptography alone is sufficient. This includes environments where secure symmetric-key  
970 establishment can take place using symmetric keys already shared between entities,  
971 environments where a single authority knows and manages all the keys, and in single-user  
972 environments.

973 In general, asymmetric cryptography is best suited for an open, multi-user environment.

974

## SECTION 5: KEY MANAGEMENT

975 The proper management of cryptographic keys is essential to the effective use of  
976 cryptography for security. Keys are analogous to the combination of a safe. If a safe  
977 combination becomes known by an adversary, that safe provides no security against  
978 penetration by that adversary. Similarly, poor key management may easily compromise  
979 strong algorithms. Ultimately, the security of information protected by cryptography  
980 directly depends on the strength of the keys, the effectiveness of mechanisms and protocols  
981 associated with keys, and the protection afforded to the keys themselves. All keys need to  
982 be protected against modification (i.e., their integrity needs to be preserved), and secret and  
983 private keys (i.e., keys used by symmetric and asymmetric algorithms, respectively) need  
984 to be protected against unauthorized disclosure (i.e., their confidentiality needs to be  
985 maintained).

986 Key management provides the foundation for the secure generation, storage,  
987 distribution/establishment, use and destruction of keys, and is essential at all phases of a  
988 key's life. Cryptography can be used to protect large amounts of data. If a strong algorithm  
989 is used to encrypt the data using keys that are properly generated, then the protection of  
990 that data can subsequently be reduced to just protecting the keys, i.e. the security of  
991 information protected by cryptography directly depends on the protection afforded the  
992 keys. Therefore, a Cryptographic Key Management System (CKMS) is required for  
993 managing the keys.

### 994 5.1 General Key Management Guidance

995 Several publications have been developed to provide general key-management guidance:  
996 SP 800-57 (see [Section 5.1.1](#)), FIPS 140 (see [Section 51.2](#)), and SP 800-131A (see [Section](#)  
997 [5.1.3](#)).

#### 998 5.1.1 Recommendation for Key Management

999 SP 800-57<sup>55</sup> provides general guidance on the management of cryptographic keys: their  
1000 generation, use, and eventual destruction. Related topics, such as algorithm selection and  
1001 appropriate key size, and cryptographic policy are also included in SP 800-57, which  
1002 consists of three parts:

- 1003 • [SP 800-57, Part 1](#), *General Guidance*, contains basic key-management guidance,  
1004 including:
  - 1005 ○ The protection required for keying material;
  - 1006 ○ Key life-cycle responsibilities;
  - 1007 ○ Key backup, archiving and recovery;
  - 1008 ○ Changing keys;
  - 1009 ○ Cryptoperiods (i.e., the appropriate lengths of time that keys are to be used);

---

<sup>55</sup> SP 800-57: Recommendation for Key Management.

- 1010           ○ Accountability and auditing;
- 1011           ○ Contingency planning; and
- 1012           ○ Key compromise recovery (e.g., by generating new keys).
- 1013           Federal agencies have a variety of information that they have determined to require  
1014           cryptographic protection; the sensitivity of the information and the periods of time  
1015           that the protection is required also vary. To this end, NIST has established four<sup>56</sup>  
1016           security strengths for the protection of information: 112, 128, 192 and 256 bits<sup>57</sup>.  
1017           These security strengths have been assigned to the **approved** cryptographic  
1018           algorithms and key sizes, and dates have been projected during which the use of  
1019           these algorithms and key sizes is anticipated to be secure. For further information,  
1020           see [SP 800-131A](#).
- 1021           Agencies need to determine the length of time that cryptographic protection is  
1022           required before selecting an algorithm and key size with the appropriate security  
1023           strength.
- 1024           • [SP 800-57, Part 2, Best Practices for Key Management Organizations](#), contains:
- 1025           ○ A generic key-management infrastructure,
- 1026           ○ Guidance for the development of organizational key-management policy  
1027           statements and key-management practices statements,
- 1028           ○ An identification of key-management information that needs to be  
1029           incorporated into security plans for general support systems and major  
1030           applications that employ cryptography, and
- 1031           ○ An identification of key-management information that needs to be  
1032           documented for all federal applications of cryptography.
- 1033           • [SP 800-57, Part 3, Application-Specific Key Management Guidance](#), addresses the  
1034           key management issues associated with currently available cryptographic  
1035           mechanisms, such as the Public Key infrastructure (PKI), Internet Protocol Security  
1036           (IPsec), the Transport Layer Security protocol (TLS), Secure/Multipart Internet  
1037           Mail Extensions (S/MIME), Kerberos, Over-the-Air Rekeying (OTAR), Domain  
1038           Name System Security Extensions (DNSSEC), Encrypted File Systems and the  
1039           Secure Shell (SSH) protocol.
- 1040           Specific guidance is provided regarding:
- 1041           ○ The recommended and/or allowable algorithm suites and key sizes,
- 1042           ○ Recommendations for the use of the mechanism in its current form for the  
1043           protections of federal government information, and

---

<sup>56</sup> A fifth security strength was originally defined to provide 80 bits of security strength, but this strength is no longer adequate for the protection of Federal information.

<sup>57</sup> A fifth security strength (i.e., 80 bits of security) was acceptable for applying cryptographic protection (e.g., encryption) prior to 2014. However, this strength is no adequate.



- 1044           ○ Security considerations that may affect the effectiveness of key-  
1045 management processes and the cryptographic mechanisms using keys that  
1046 are generated and managed by those key-management processes.

1047 Note that in the case of TLS, a reference is provided to a separate publication – [SP](#)  
1048 [800-52](#)<sup>58</sup> – that provides extensive details for using TLS.

1049 New key-management techniques and mechanisms are constantly being developed,  
1050 and existing key-management mechanisms and techniques are constantly being  
1051 refined. While the security-guidance information contained in Part 3 will be  
1052 updated as mechanisms and techniques evolve, new products and technical  
1053 specifications can always be expected that are not reflected in the current version  
1054 of the document. Therefore, the context provided may include status information,  
1055 such as version numbers or implementation status at the time that the document was  
1056 published.

### 1057 **5.1.2 Security Requirements for Cryptographic Modules**

1058 [FIPS 140](#)<sup>59</sup> provides minimum security requirements for cryptographic modules that  
1059 embody or support cryptography in federal information systems. A cryptographic module  
1060 performs the actual cryptographic computations for a security system protecting sensitive  
1061 information. The security requirements cover areas related to the secure design and  
1062 implementation of a cryptographic module, including the module specification;  
1063 cryptographic module ports and interfaces; roles, services and authentication; finite-state  
1064 models; physical security; the operational environment; cryptographic key management;  
1065 electromagnetic interference/electromagnetic compatibility (EMI/EMC); self-tests; design  
1066 assurance; and the mitigation of attacks.

1067 FIPS 140 is applicable to all federal agencies that use cryptography to protect sensitive  
1068 information in computer and telecommunications systems. Further information about FIPS  
1069 40 and the validation of cryptographic modules is available at  
1070 <http://csrc.nist.gov/groups/STM/cmvp/index.html>.

### 1071 **5.1.3 Transitions to New Cryptographic Algorithms and Key Lengths**

1072 With the development and publication of [SP 800-57, Part 1](#), NIST provided  
1073 recommendations for transitioning to new cryptographic algorithms and key lengths  
1074 because of algorithm breaks or the availability of more powerful computers that could be  
1075 used to efficiently search for cryptographic keys. [SP 800-131A](#) was developed to provide  
1076 more specific guidance for such transitions. Each algorithm and service is addressed in SP

---

<sup>58</sup> SP 800-52: Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations.

<sup>59</sup> FIPS 140: Security Requirements for Cryptographic Modules.

1077 800-131A, indicating whether its use is acceptable<sup>60</sup>, deprecated<sup>61</sup>, restricted<sup>62</sup>, allowed  
1078 only for legacy applications<sup>63</sup>, or disallowed.

## 1079 **5.2 Cryptographic Key Management Systems**

1080 Several publications have been developed for the development of key-management  
1081 systems: [SP 800-130](#)<sup>64</sup> (see [Section 5.2.1](#)), [SP 800-152](#)<sup>65</sup> (see [Section 5.2.2](#)) and  
1082 documents relating to the Public Key Infrastructure used for asymmetric-key cryptography  
1083 (see [Section 5.2.3](#)).

1084 A CKMS includes policies, procedures, components and devices that are used to protect,  
1085 manage and distribute cryptographic keys and associated information (called metadata). A  
1086 CKMS includes all devices or subsystems that can access a key or its metadata. The  
1087 devices could be computers, cell phones, tablets, or other smart devices, such as cars, alarm  
1088 systems, or refrigerators.

### 1089 **5.2.1 Key Management Framework**

1090 [SP 800-130](#) contains topics that should be considered by a CKMS designer when  
1091 developing a CKMS design specification. Topics include security policies, cryptographic  
1092 keys and metadata, interoperability and transitioning, security controls, testing and system  
1093 assurances, disaster recovery, and security assessments.

1094 For each topic, SP 800-130 specifies one or more documentation requirements that need to  
1095 be addressed by the designer. SP 800-130 is intended to assist in:

- 1096 • The definition of the CKMS design by requiring the specification of significant  
1097 CKMS capabilities,
- 1098 • Encouraging CKMS designers to consider the factors needed in a comprehensive  
1099 CKMS,
- 1100 • Logically comparing different CKMSs and their capabilities,
- 1101 • Performing security assessments by requiring the specification of implemented and  
1102 supported CKMS capabilities, and
- 1103 • Forming the basis for the development of Profiles that specify the specific  
1104 requirements for the CKMS to be used by an organization.

---

<sup>60</sup> No security risk is known at present.

<sup>61</sup> The use of the algorithm and key length is allowed, but the user must accept some risk.

<sup>62</sup> The use of the algorithm is discouraged, and there are additional restrictions required for use.

<sup>63</sup> The algorithm and key length may be used to process already-protected information, but there may be a risk in doing so.

<sup>64</sup> SP 800-130: A Framework for Designing Cryptographic Key Management Systems.

<sup>65</sup> SP 800-152: A Profile for U. S. Federal Cryptographic Key Management Systems (CKMS).



## 1105 **5.2.2 Key Management System Profile**

1106 [SP 800-152](#) contains requirements for the design, implementation, procurement,  
1107 installation, configuration, management, operation and use of a CKMS by and for U.S.  
1108 federal organizations and their contractors. The Profile is based on SP 800-130 (see [Section](#)  
1109 [5.2.1](#)). SP 800-152 specifies requirements, makes recommendations for federal  
1110 organizations having special security needs and desiring to augment the base security and  
1111 key-management services, and suggests additional features that may be desirable to  
1112 implement and use.

1113 In addition to providing design requirements to be incorporated into a CKMS design, SP  
1114 800-152 provides requirements for a Federal CKMS (FCKMS) to be operated by a service  
1115 provider that may be a federal agency or a third party operating an FCKMS under contract  
1116 for one or more federal agencies and their contractors.

1117 This Profile is intended to:

- 1118 • Assist CKMS designers and implementers in supporting appropriate cryptographic  
1119 algorithms and keys, selecting the metadata associated with the keys, and selecting  
1120 protocols for protecting sensitive U.S. federal computing applications and data;
- 1121 • Establish requirements for testing, procurement, installation, configuration,  
1122 administration, operation, maintenance and usage of the FCKMS;
- 1123 • Facilitate an easy comparison of one CKMS with another by analyzing their designs  
1124 and implementations in order to understand how each meets the Framework and  
1125 Profile requirements; and
- 1126 • Assist in understanding what is needed to evaluate, procure, install, configure,  
1127 administer, operate, and use an FCKMS that manages the cryptographic keys that  
1128 protect sensitive and valuable data obtained, processed, stored, and used by U.S.  
1129 federal organizations and their contractors.

## 1130 **5.2.3 Public Key Infrastructure**

1131 A PKI is a security infrastructure that creates and manages public-key certificates to  
1132 facilitate the use of public-key (i.e., asymmetric-key) cryptography. To achieve this goal,  
1133 a PKI needs to perform two basic tasks:

- 1134 1. Generate and distribute public key certificates that bind public keys to the identifier  
1135 associated with the owner of the corresponding private key<sup>66</sup> and to other required  
1136 information *after* validating the accuracy of the information to be bound, and
- 1137 2. Maintain and distribute certificate-status information for unexpired and revoked  
1138 certificates.

1139 Two types of certificates are commonly used: certificates used to distribute the public keys  
1140 that are used to verify digital signatures, and certificates used to distribute public keys used  
1141 for key management (i.e., key establishment). Each certificate associated with digital

---

<sup>66</sup> The identifier could be the true identity of the owner, or could be an alias or a pseudonym used to represent the owner.

1142 signatures provides the public keys of one of the three digital-signature algorithms  
1143 approved in [FIPS 186](#): DSA, ECDSA or RSA (see [Section 3.3](#)). Certificates that convey  
1144 the public keys to be used for key establishment may be of two types: those that provide a  
1145 key-agreement public key (see [Section 5.3.3](#)), and those that provide a key-transport public  
1146 key (see [Section 5.3.4](#)). Key-usage bits in a certificate indicate the purpose for which the  
1147 public key is intended to be used.

1148 As discussed in [Section 3.3](#), public keys can be made available to anyone. However, a  
1149 private key must be maintained under the exclusive control of the owner of that private  
1150 key<sup>67</sup> (i.e., the user that is authorized to use the private key).

- 1151 • If a private key that is used to generate digital signatures is lost, the owner can no  
1152 longer generate digital signatures; some policies may permit users to maintain  
1153 backup copies of the private key for continuity of operations, but this is not  
1154 encouraged, so an alternative is to simply generate new key pairs and certificates.
- 1155 • If the private key used to generate digital signatures is compromised, relying parties  
1156 can no longer trust the digital signatures generated using that private key (e.g.,  
1157 someone may be using the signature to provide false information).
- 1158 • If a private key used for key establishment is lost (e.g., a key used for key transport  
1159 or key agreement), then further key establishment processes cannot be  
1160 accomplished until the key is recovered or replaced; if the key is needed to recover  
1161 data protected by the key, then that data is lost unless the key can be recovered. For  
1162 example, if the key is used to transport a decryption key for encrypted data, and the  
1163 key is lost, then the encrypted data cannot be decrypted. To ensure that access to  
1164 critical data is not lost, PKIs often backup the private key-establishment key for  
1165 possible recovery.
- 1166 • If a private key used for key establishment is compromised, then any transactions  
1167 involving that key cannot be trusted (e.g., someone other than the true owner of the  
1168 private key may be attempting to enter into a supposedly "secure" transaction for  
1169 some illicit purpose).

### 1170 **5.2.3.1 PKI Components, Relying Parties and Their Responsibilities**

1171 For scalability, PKIs are usually implemented with a set of complementary components,  
1172 each focused on specific aspects of the PKI process. The main PKI tasks are assigned to  
1173 the following logical components; other components are also used to support the PKI, but  
1174 are not discussed here (see [SP 800-32](#)<sup>68</sup> for further discussion):

- 1175 • *Certification authorities* (CAs) generate certificates and certificate-status  
1176 information, and

---

<sup>67</sup> An exception could be some other trusted entity, such as the owner's organization. In these cases, the organization could be considered to be the *real* owner of the key.

<sup>68</sup> SP 800-32: Introduction to Public Key Technology and the Federal PKI Infrastructure.

- 1177       • *Registration authorities* (RAs) verify the identity of users applying for a  
1178       certificate<sup>69</sup> and authenticate other information to be included in the certificate.

1179   In general, a PKI operates as follows:

- 1180       1. An entity applies to an RA to request a certificate.
- 1181       2. The RA verifies the identity of the applicant, and 2) verifies the information to be  
1182       inserted in the certificate.
- 1183       3. If the checks made by the RA in step 2 indicate that the information to be inserted  
1184       in the certificate is valid, then the RA sends the public key and other relevant  
1185       information to the CA to request that a certificate be generated.
- 1186       4. Upon receiving the certificate request from the RA, the CA creates a digital  
1187       certificate, returns the certificate to the RA and deposits the certificate in a  
1188       repository.
- 1189       5. When a relying party interacts with another entity that has a public-key certificate,  
1190       the relying party needs to obtain the other entity's certificate, either directly or from  
1191       the CA's repository. After acquiring the certificate, the relying entity verifies the  
1192       signature on the certificate. Assuming that the certificate is "good," then the relying  
1193       party can proceed safely with its interaction with the certificate's owner.

1194   Most of the interaction involved with using a certificate is transparent to the user. However,  
1195   a user or a system administrator may be responsible for obtaining and installing a  
1196   certificate. Thereafter, an application (e.g., a browser) uses the certificate to interact with  
1197   other entities, and the user may not be aware of these actions. An exception might be when  
1198   a certificate has expired or been revoked, in which case a message may be displayed to  
1199   indicate this status.

1200   Certificates expire at a predetermined time unless revoked prior to the expiration date.  
1201   Certificates can be revoked for a variety of reasons, including the compromise of the  
1202   private key corresponding to the public key in the certificate, and the owner of the  
1203   certificate leaving the organization. When a certificate has been revoked, a system will  
1204   quite often display the certificate-revocation message and perhaps include the reason for  
1205   the revocation. Depending on the application implementation and the revocation reason,  
1206   the application could disallow further actions, or could allow the user to indicate whether  
1207   to ignore the warning and continue operations, or to simply discontinue operations. This  
1208   warning must not be taken lightly. Ignoring the warning means that the user is accepting  
1209   the risks associated with doing so. For example, if a warning indicates a compromised  
1210   digital signature certificate, there is a possibility that someone other than the claimed owner  
1211   of the certificate actually used the private key corresponding to the public key to sign data.  
1212   Depending on the data, it may not be prudent to ignore the warning. A user should consult  
1213   with his organization to determine how to respond to this warning.

---

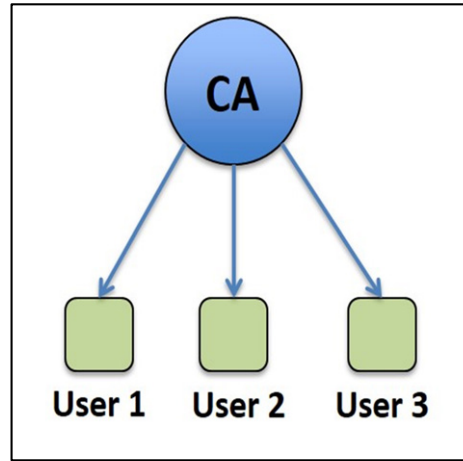
<sup>69</sup> The certificate could be for the user or for a device for which the user is authorized to obtain a certificate.

1214 **5.2.3.2 Basic Certificate Verification Process**

1215 A PKI consists of at least one CA with its subscribers, as shown in Figure 5. Each of the  
1216 subscribers (e.g., User 1, User 2 and User 3) obtains a certificate containing their public  
1217 key and other information, which is signed by their CA. All CA subscribers are provided  
1218 with the public key of the CA.

1219 As a basic example of how this works, suppose that User 3 signs a document and sends it  
1220 to User 1, who needs to verify the contents and source of the signed document. This is  
1221 accomplished as follows:

- 1222 1. User 1 obtains the certificate containing the  
1223 public key that corresponds to the private key  
1224 used to sign the document, i.e., User 1 obtains  
1225 User 3’s certificate. Either User 3 supplies  
1226 that certificate, or the certificate is obtained  
1227 from some other source, e.g., the CA.
- 1228 2. User 1 verifies User 3’s certificate using the  
1229 CA’s public key.
- 1230 3. User 1 then employs the public key in User  
1231 3’s certificate to verify the signature on the  
1232 document received from User 3. If the  
1233 signature is successfully verified, then User 1  
1234 knows that User 3 generated the signature,  
1235 and no unauthorized modifications were  
1236 made to the document after the signature was  
1237 generated.



**Figure 5: Basic Certificate Verification Example**

1238 Note that other more-complicated scenarios exist when users subscribing to different CAs  
1239 need to interact using CAs that have cross certified by signing a certificate for each other.

1240 **5.2.3.3 CA Certificate Policies and Certificate Practice Statements**

1241 Each CA has a Certificate Policy and a Certificate Practices Statement. As defined by  
1242 ITU<sup>70</sup> Recommendation [X.509](#), a Certificate Policy (CP) is “a named set of rules that  
1243 indicates the applicability of a certificate to a particular community and/or class of  
1244 application with common security requirements.” The CP defines the expectations and  
1245 requirements of the relying party community that will trust the certificates issued by the  
1246 CAs using that policy. A CP addresses such issues as key generation and storage; certificate  
1247 generation; key escrow<sup>71</sup> and recovery; certificate status services, including Certificate  
1248 Revocation List (CRL) generation and distribution; and system management functions,  
1249 such as security audits, configuration management, and archiving.

<sup>70</sup> ITU is the abbreviation of the International Telecommunication Union.

<sup>71</sup> Saving a key or information that allows the key to be reconstructed so that the key can be recovered if ever needed (e.g., by being lost or corrupted).

1250 A Certification Practice Statement (CPS) describes how a specific CA issues and manages  
1251 public-key certificates. The CPS is derived from the applicable CP for the community or  
1252 application in which the CA participates.

1253 A Federal Public Key Infrastructure (FPKI) has been established for use by the Federal  
1254 Government (see [Section 5.2.3.4](#) for further information).

1255 DRAFT [NISTIR 7924](#)<sup>72</sup> identifies a baseline set of security controls and practices to  
1256 support the secure issuance of certificates. NISTIR 7924 is designed to be used as a  
1257 template and guide for writing a CP for a specific community, or a CPS for a specific CA.

### 1258 **5.2.3.4 Federal Public Key Infrastructure**

1259 A Federal Public Key Infrastructure (FPKI) provides the Federal Government with a  
1260 common infrastructure to administer digital certificates and public-private key pairs. The  
1261 network portion of the FPKI (commonly referred to as the “Bridge”) consists of “Principal  
1262 CAs” designated by various agencies. Each CA within the bridge is cross-certified with  
1263 every other CA within the bridge, thus establishing a conduit for trust relationships among  
1264 all CAs within the FPKI. Each Principal CA may also be associated with other CAs that  
1265 are not part of the bridge. For more information about the FPKI, including its certificate  
1266 policy and certificate practices statement, see [http://www.idmanagement.gov/federal-  
1267 public-key-infrastructure](http://www.idmanagement.gov/federal-public-key-infrastructure).

## 1268 **5.3 Key Establishment**

1269 Key establishment is the means by which keys are generated and provided to the entities  
1270 that are authorized to use them. An entity may be a person, organization, device or process.  
1271 Scenarios for which key establishment could be performed include the following:

- 1272 • A single entity could generate a key (see [Section 5.3.1](#)) and use it without providing  
1273 it to other entities (e.g., for protecting locally stored data),
- 1274 • A key could be derived from a key that is already shared between two or more  
1275 entities (see [Section 5.3.2](#)),
- 1276 • Two entities could generate a key using contributions (i.e., data) from each entity  
1277 using an automated protocol that incorporates a key-agreement scheme (see [Section  
1278 5.3.3](#)), or
- 1279 • A single entity could generate a key and provide it to one or more other entities,  
1280 either by a manual means (e.g., a courier or a face-to-face meeting, with the key in  
1281 either printed or electronic form, such as on a flash drive) or using automated  
1282 protocols that incorporate a key-transport scheme (see [Sections 5.3.4](#) and [5.3.5](#)).

### 1283 **5.3.1 Key Generation**

1284 Cryptographic keys are required by most cryptographic algorithms, the exception being  
1285 hash functions when not used as a component of another cryptographic process (e.g.,

---

<sup>72</sup> NISTIR 7924: Reference Certificate Policy (Second Draft).

1286 HMAC). [SP 800-133](#)<sup>73</sup> discusses the generation of the keys to be used with the **approved**  
1287 cryptographic algorithms.

1288 All keys must be based directly or indirectly on the output of an **approved** Random Bit  
1289 Generator (RBG) and must be generated within FIPS 140-compliant cryptographic  
1290 modules (see [FIPS 140](#)). Any random value required by the module must be generated  
1291 within a cryptographic module.

1292 [SP 800-133](#) provides guidance on generating a key directly from an RBG, and references  
1293 other publications for additional information required for the generation of keys for specific  
1294 algorithms:

- 1295 • [FIPS 186](#) provides rules for the generation of the key pairs to be used for the  
1296 generation of digital signatures,
- 1297 • [SP 800-108](#) provides methods for the generation of keys from an already-shared  
1298 key (see [Section 5.3.2](#)),
- 1299 • [SP 800-56A](#) specifies the rules for the generation of key pairs for Diffie-Hellman  
1300 and MQV key-agreement schemes (see [Section 5.3.3](#)),
- 1301 • [SP 800-56B](#) specifies the rules for the generation of key pairs for RSA key-  
1302 agreement and key-transport schemes (see [Sections 5.3.3](#) and [5.3.4](#), respectively),  
1303 and
- 1304 • [SP 800-132](#) specifies the rules for the generation of keys from passwords (see  
1305 [Section 5.3.6](#)).

### 1306 **5.3.2 Key Derivation**

1307 Key derivation is concerned with the generation of a key from secret information, although  
1308 non-secret information may also be used in the generation process in addition to the secret  
1309 information. Typically, the secret information is shared among entities that need to derive  
1310 the same key for subsequent interactions. The secret information could be a key that is  
1311 already shared between the entities (i.e., a pre-shared key), or could be a shared secret that  
1312 is derived during a key-agreement scheme (see [Section 5.3.3](#)).

1313 [SP 800-108](#)<sup>74</sup> specifies several key-derivation functions that use pre-shared keys. A pre-  
1314 shared key could have been

- 1315 • Generated by one entity and provided to one or more other entities by some manual  
1316 means (e.g., a courier or face-to-face meeting),
- 1317 • Agreed upon by the entities using an automated key-agreement scheme (see [Section](#)  
1318 [5.3.3](#)), or
- 1319 • Generated by one entity and provided to another entity using an automated key-  
1320 transport scheme (see [Section 5.3.4](#)).

---

<sup>73</sup> SP 800-133: Recommendation for Cryptographic Key Generation.

<sup>74</sup> SP 800-108: Recommendation for Key Derivation Using Pseudorandom Functions.



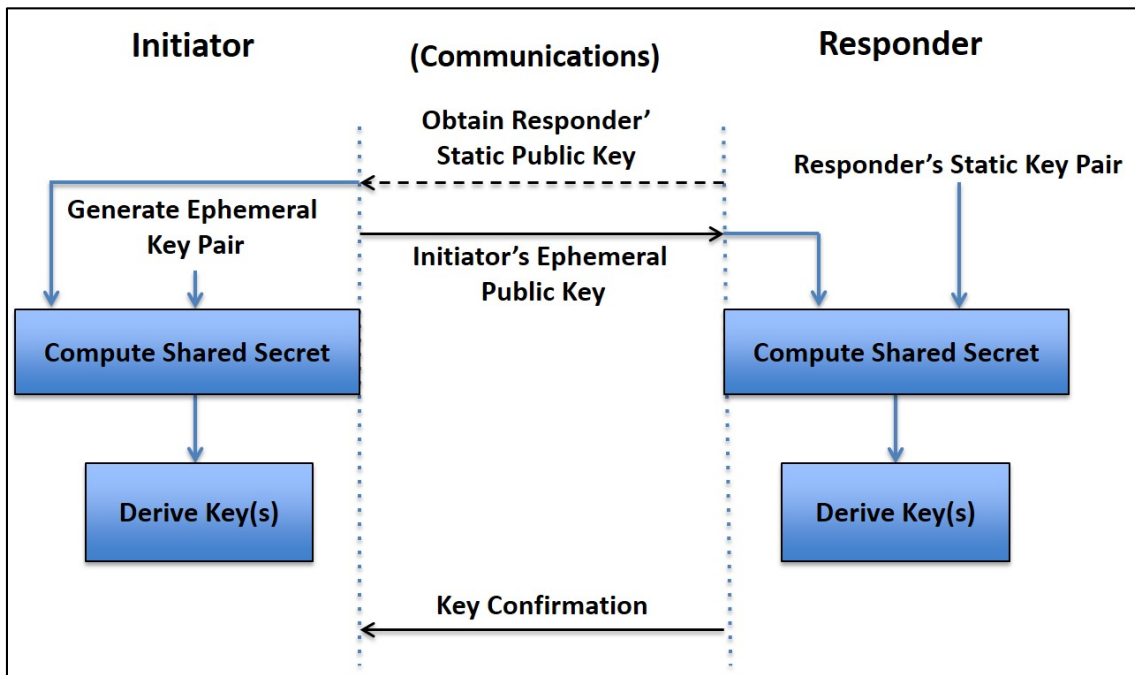
1321 [SP 800-56A](#), [SP 800-56B](#) and [SP 800-56C](#)<sup>75</sup> provide methods for deriving keys from the  
 1322 shared secrets generated during key agreement (see [Section 5.3.3](#)). SP 800-56A and SP  
 1323 800-56 B specify two key-derivation methods for this purpose, and refer to SP 800-56C  
 1324 and [SP 800-135](#)<sup>76</sup> for additional approved methods.

1325 **5.3.3 Key Agreement**

1326 Key agreement is a key-establishment procedure in which the resultant keying material is  
 1327 a function of information contributed by all participants in the key-agreement process so  
 1328 that no participant can predetermine the value of the resulting keying material  
 1329 independently of the contributions of the other participants. Key agreement is usually  
 1330 performed using automated protocols.

1331 [SP 800-56A](#) and [SP 800-56B](#) provide several automated pair-wise key-agreement schemes,  
 1332 i.e., key-agreement schemes involving two parties. For each scheme, a shared secret is  
 1333 generated, and keying material is derived from the shared secret using a key-derivation  
 1334 method specified or approved by reference in SP 800-56A, SP 800-56B or [SP 800-56C](#).

1335 SP 800-56A and SP 800-56B include variations of key-agreement schemes, differing in  
 1336 the number of keys used and whether the keys are long term (i.e., static) or an ephemeral  
 1337 value (e.g., a nonce or a short-term key pair). The key-agreement schemes have two  
 1338 participating entities: an initiator and a responder.



1339  
1340

**Figure 6: Key Agreement Example**

<sup>75</sup> SP 800-56C: Recommendation for Key Derivation through Extraction-then-Expansion.

<sup>76</sup> SP 800-135: Recommendation for Existing Application-Specific Key Derivation Functions.

1341 Figure 6 provides an example of a scheme where the responder uses a static key pair during  
1342 the scheme, and the initiator uses an ephemeral key pair. Note that other key-agreement  
1343 schemes may use other arrangements of key pairs (e.g., each party could use a static key  
1344 pair or each party could use an ephemeral key pair). In the example provided in the figure  
1345 above, the responder's private key is retained by the responder (who is the owner of the key  
1346 pair), but the responder's public key may be provided to anyone. In this example, the public  
1347 key is provided to the initiator:

- 1348 1. The initiator obtains the responder's public key (e.g., from a CA or directly from the  
1349 responder); this public key is the responder's contribution to the key-agreement  
1350 process.
- 1351 2. The initiator then generates a short-term key pair (i.e., an ephemeral key pair), and  
1352 sends the ephemeral public key to the responder, retaining the ephemeral private  
1353 key. The ephemeral public key is the initiator's contribution to the key-agreement  
1354 process.
- 1355 3. Both parties use their own key pair and the other party's public key to generate a  
1356 shared secret.
- 1357 4. Both parties then use their copy of the shared secret to derive one or more keys that  
1358 are (hopefully) identical.

1359 Key confirmation is an optional, but highly recommended, step that provides assurance  
1360 that both parties now have the same (identical) key(s), and is shown in Figure 6 for the case  
1361 that the initiator receives key confirmation from the responder. See [SP 800-56A](#) and [SP](#)  
1362 [800-56B](#) for further information.

1363 SP 800-56A specifies Diffie-Hellman (DH) and MQV key-agreement schemes using finite  
1364 field or elliptic curve mathematics and asymmetric key pairs to generate the shared secret,  
1365 and SP 800-56B specifies two RSA key-agreement schemes. SP 800-56A and SP 800-56B  
1366 also provide an analysis of the merits of each key-agreement scheme.

### 1367 **5.3.4 Key Transport**

1368 Key transport is a method whereby one party (the sender) generates a key and distributes  
1369 it to one or more other parties (the receiver(s)). Key transport could be accomplished using  
1370 manual methods (e.g., using a courier) or performed using automated protocols. [SP 800-](#)  
1371 [56A](#) and [SP 800-56B](#) provide automated pair-wise key-transport schemes, and an analysis  
1372 of the merits of each key-transport scheme.

#### 1373 **5.3.4.1 SP 800-56A Key Transport**

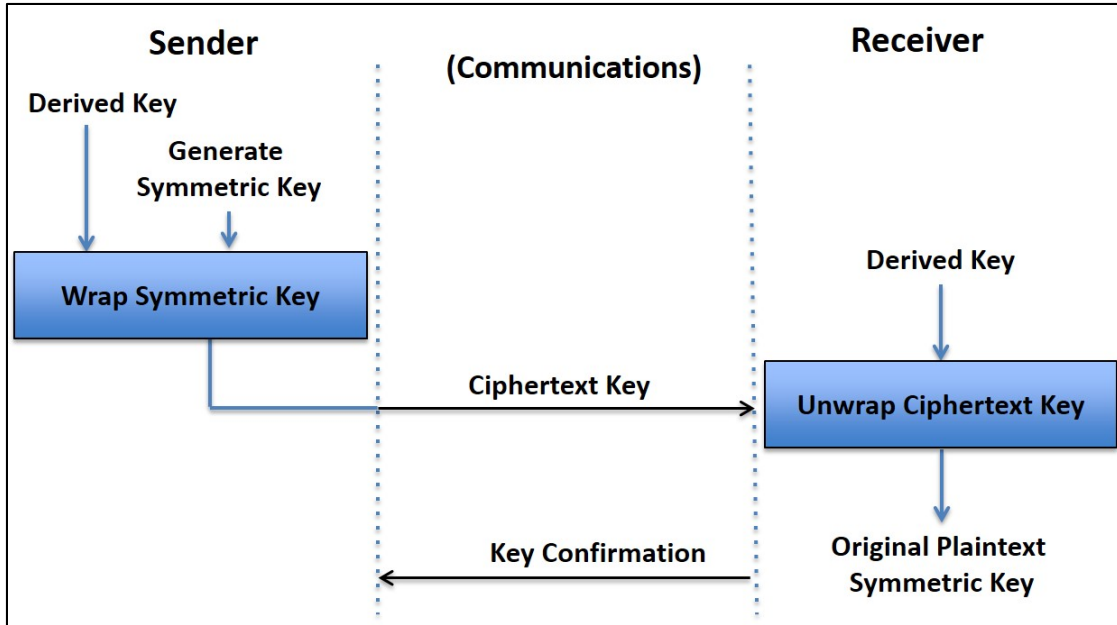
1374 [SP 800-56A](#) specifies a key-transport method whereby a key-establishment transaction  
1375 includes both a key-agreement process and a key-wrapping process. Key wrapping is a  
1376 process that provides both confidentiality and integrity protection for keying material using  
1377 a symmetric-key algorithm (see [Section 5.3.5](#) for further information about key wrapping).

1378 During the transaction, the key generated during the key-agreement part of the transaction  
1379 is used as a key-wrapping key with a symmetric-key algorithm (e.g., AES) by the sending



1380 party to wrap a key to be sent to the other party (the receiver). Note that the sender can be  
1381 either the initiator or the responder in the key-agreement process.

1382 Figure 7 illustrates the key transport process that follows the key-agreement discussed in  
1383 [Section 5.3.3](#) and shown in [Figure 6](#). After the key-agreement part of the transaction, the  
1384 initiator and responder share a symmetric key-wrapping key, which is then used as follows:



1385  
1386

**Figure 7: SP 800-56A Key Transport Example**

1387 The sender:

- 1388 1. Generates (or otherwise obtains) a symmetric key to be transported (note that  
1389 the sender could have been either the initiator or the responder in the key-  
1390 agreement part of the transaction),
- 1391 2. Wraps the symmetric key from step 1 using the key-wrapping key, and
- 1392 3. Sends the resulting ciphertext (i.e., the wrapped key) to the intended receiver.

1393 The receiver:

- 1394 4. Unwraps the ciphertext using his copy of the key-wrapping key to obtain the  
1395 original plaintext symmetric key, and
- 1396 5. Optionally performs key confirmation; although this step is optional, it is highly  
1397 recommended to provide assurance that both parties now have the same  
1398 symmetric key.

1399 **5.3.4.2 SP 800-56B Key Transport**

1400 [SP 800-56B](#) specifies two very different methods for transporting keys whereby the sender  
1401 uses the receiver’s public key to securely transport keying material to the receiver.

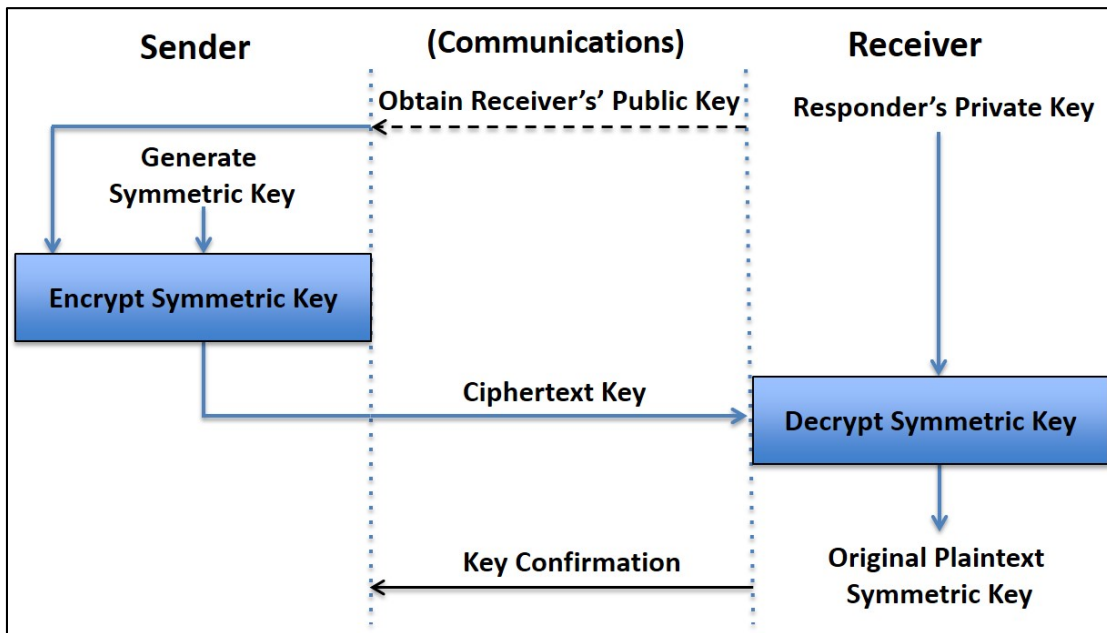
1402 Figure 8 provides a simplified example of one of the key-transport methods in SP 800-56B.  
1403 The receiver must have a key pair that is used during a key-transport transaction. Key  
1404 transport is accomplished as follows.

1405 The sender:

- 1406 1. Obtains the public key of the intended receiver,
- 1407 2. Generates a symmetric key to be transported,
- 1408 3. Encrypts the symmetric key using the receiver's public key, and
- 1409 4. Sends the resulting ciphertext key to the receiver.

1410 The receiver:

- 1411 5. Uses his private key to decrypt the ciphertext key, thus obtaining the original  
1412 plaintext key.
- 1413 6. Optionally performs key confirmation; although this step is optional, it is  
1414 highly recommended to provide assurance that both parties now have the same  
1415 symmetric key.



1416

1417

**Figure 8: SP 800-56B Key Transport Example**

1418 **5.3.5 Key Wrapping**

1419 Key wrapping is a method used to provide confidentiality and integrity protection to keys  
1420 (and possibly other information) using a symmetric key-wrapping key and a symmetric-  
1421 key block cipher algorithm. The wrapped keying material can then be stored or transmitted  
1422 securely. Unwrapping the keying material requires the use of the same same algorithm and  
1423 key-wrapping key that was used during the original wrapping process.

1424 Key wrapping differs from simple encryption in that the wrapping process includes an  
1425 integrity feature. During the unwrapping process, this integrity feature is used to detect  
1426 accidental or intentional modifications to the wrapped keying material.

1427 Three methods have been specified in [SP 800-38F](#)<sup>77</sup> for key wrapping, and other SP 800-  
1428 38 modes (or combination of modes) that can also be used for key wrapping are also  
1429 **approved** in SP 800-38F. Depending on the method or mode, either AES or TDEA can be  
1430 used.

### 1431 **5.3.6 Derivation of a Key from a Password**

1432 Keys can be derived from passwords. Due to the ease of guessing most passwords, keys  
1433 derived in this manner are not suitable to be used for most applications. However, [SP 800-  
1434 132](#)<sup>78</sup> specifies a family of functions that can be used to derive keying material from a  
1435 password<sup>79</sup> for electronic storage applications (e.g., when encrypting an entire disk drive).

## 1436 **5.4 Key Management Issues**

1437 A number of issues need to be addressed for selecting and using a CKMS.

### 1438 **5.4.1 Manual vs. Automated Key Establishment**

1439 As discussed in Sections [5.3](#) and [5.3.4](#), keys can be established between entities either  
1440 manually or using automated methods. In many cases, a hybrid approach is used in which  
1441 an entity generates and manually distributes one or more keys to other entities, and  
1442 thereafter these keys are used to establish other keys (see [SP 800-56A](#) and [SP 800-56B](#)).

1443 The number of keys to be manually distributed depends on the type of cryptography to be  
1444 used (i.e., symmetric or asymmetric methods) and must be considered when selecting the  
1445 capabilities required of a CKMS.

### 1446 **5.4.2 Selecting and Operating a CKMS**

1447 A CKMS could be designed, implemented and operated by the organization that will use  
1448 it. Or, the organization could operate a CKMS procured from a vendor. Or, an organization  
1449 could procure the services of a third party that procures a CKMS from a vendor. Whichever  
1450 choice is made, the organization needs to make sure that the CKMS that is used provides  
1451 the protections that are required for the organization's information. [SP 800-130](#) and [SP  
1452 800-152](#) discuss the considerations that need to be addressed by the federal organization,  
1453 including the scalability of the CKMS, and the metadata to be associated with the keys.

### 1454 **5.4.3 Storing and Protecting Keys**

1455 Keys can be stored in a number of places and protected in a variety of ways. They could  
1456 be stored in a safe. They could be present only in a validated cryptographic module where

---

<sup>77</sup> SP 800-38F: Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping.

<sup>78</sup> SP 800-132: Recommendation for Password-Based Key Derivation Part 1: Storage Applications.

<sup>79</sup> Note that this publication considers a passphrase to be a password.

1457 the module itself might adequately protect the keys, depending on its design. Keys could  
1458 also be stored on electronic media, such as a flash drive; in this case, a key may need to be  
1459 encrypted or split into key components so that no single person can determine what the key  
1460 is. These issues need to be addressed for operational keys.

1461 Certain keys may need to be backed up so that if an operational key is inadvertently lost or  
1462 modified, it can be recovered and operations resumed. Some keys may also need to be  
1463 archived for long-term storage (e.g., because of legal requirements or to decrypt archived  
1464 data). A key-recovery capability is needed whenever keys are backed up or archived. This  
1465 capability needs to be designed so that the keys can be recovered in an acceptable amount  
1466 of time and only by those entities authorized to do so; see [Part 1 of SP 800-57](#) for more  
1467 information about key backup, key archiving and key recovery.

#### 1468 **5.4.4 Cryptoperiods**

1469 A cryptoperiod is the time span during which a specific key is authorized for use. A  
1470 cryptoperiod for a key is assigned for a number of reasons, including limiting the amount  
1471 of exposure of encrypted data if a single key is compromised. Cryptoperiods are usually  
1472 assigned for a carefully considered period of time or by the maximum amount of data  
1473 protected by the key. Tradeoffs associated with the determination of a cryptoperiod involve  
1474 the risks and consequences of exposure. Section 5.3 of [SP 800-57, Part 1](#) provides a more  
1475 detailed discussion of the need for establishing cryptoperiods, the factors to be considered  
1476 when deciding on a suitable cryptoperiod and some suggestions for the length of  
1477 cryptoperiods.

#### 1478 **5.4.5 Use Validated Algorithms and Cryptographic Modules**

1479 Cryptographic algorithms must be validated and implemented in [FIPS 140](#)-validated  
1480 cryptographic modules. Every IT product available makes a claim as to functionality and/or  
1481 offered security. When protecting sensitive data, a minimum level of assurance is needed  
1482 that a product's stated security claim is valid. There are also legislative restrictions  
1483 regarding certain types of technology, such as cryptography, that require federal agencies  
1484 to use only tested and validated products.

1485 Federal agencies, private industry, and the public rely on cryptography for the protection  
1486 of information and communications used in electronic commerce, critical infrastructure,  
1487 and other application areas. At the core of all products offering cryptographic services is  
1488 the cryptographic module. Cryptographic modules, which contain cryptographic  
1489 algorithms, are used in products and systems to provide security services such as  
1490 confidentiality, integrity, and authentication. Although cryptography is used to provide  
1491 security, weaknesses such as poor design or weak algorithms can render the product  
1492 insecure and place highly sensitive information at risk. Adequate testing and validation of  
1493 the cryptographic module and its underlying cryptographic algorithms against established  
1494 standards is essential to provide security assurance.

1495 NIST has established programs to validate the implementation of the **approved**  
1496 cryptographic algorithms and the cryptographic modules in which they are used: the

1497 Cryptographic Algorithm Validation Program (CAVP) and the Cryptographic Module  
1498 Validation Program (CMVP). Information about the CAVP is available at  
1499 <http://csrc.nist.gov/groups/STM/cavp/index.html>, while information about the CMVP is  
1500 available at <http://csrc.nist.gov/groups/STM/cmvp/index.html>.

1501 Also, see [Section 5.1.2](#) in this document for a discussion of the security requirements for  
1502 cryptographic modules.

#### 1503 **5.4.6 Control of Keying Material**

1504 The access to keys needs to be controlled. A key should only be accessible by an authorized  
1505 entity, and only for the purpose for which it is authorized. For example, a key designated  
1506 for key transport must not be used for the generation or verification of digital signatures.

1507 The proliferation of keys also needs to be controlled. While it is often convenient to make  
1508 copies of keys, these extra copies need to be accounted for. If a key is compromised, that  
1509 key and all its copies may need to be destroyed to prevent subsequent unauthorized use.  
1510 For example, if a private key used for the generation of a digital signature is compromised,  
1511 and a copy of the key still exists after the original copy was destroyed, then there is a  
1512 possibility that the copy could be used to generate unauthorized digital signatures at a later  
1513 time.

1514 Users must be provided with a list of responsibilities and liabilities, and each user should  
1515 sign a statement acknowledging these concerns before receiving a key. Users must be made  
1516 aware of their unique responsibilities, especially regarding the significance of a key  
1517 compromise or loss. Users must be able to store their secret and private keys securely, so  
1518 that no intruder can access them, yet the keys must be readily accessible for legitimate use.

#### 1519 **5.4.7 Compromises**

1520 It is imperative to have a plan for handling the compromise or suspected compromise of  
1521 keys, particularly those used and managed at a central site (e.g., the keys used by a CA to  
1522 sign certificates); this should be established before the system becomes operational. A  
1523 compromise-recovery plan should address what actions will be taken with compromised  
1524 system software and hardware, CA keys, user keys, previously generated signatures,  
1525 encrypted data, etc. [SP 800-57, Part 1](#) includes discussions of the effects of a key  
1526 compromise, measures for minimizing the likelihood or consequences of a key  
1527 compromise, and what should be considered in developing a compromise-recovery plan.

1528 If someone's private or secret key is lost or compromised, other users must be made aware  
1529 of this, so that they will no longer initiate the protection of data using a compromised key,  
1530 or accept data protected with a compromised key without assessing and accepting the risk  
1531 of doing so. This notification is often accomplished using CRLs or Compromised Key Lists  
1532 (CKLs); see [SP 800-57, Part 1](#) for discussions.

1533 In some cases, a key and all copies of the key should be destroyed immediately upon the  
1534 detection of a key compromise. For example, a private key used for the generation of digital  
1535 signatures should be immediately destroyed. However, the corresponding public key may  
1536 need to remain available for verifying the signatures that were previously generated using

1537 the compromised private key. Note that there is a risk associated with accepting these  
1538 signatures.

#### 1539 **5.4.8 Accountability and Auditing**

1540 Accountability involves the identification of those entities that have access to or control of  
1541 cryptographic keys throughout their lifecycles. Accountability can be an effective tool to  
1542 help prevent key compromises and to reduce the impact of compromises when they are  
1543 detected. Accountability 1) aids in the determination of when a compromise could have  
1544 occurred and what individuals could have been involved, 2) discourages key compromise  
1545 because users know their access to the key is known, and 3) is useful in determining where  
1546 the key was used and what data or other keys were protected by a compromised key, and  
1547 therefore, may also be compromised.

1548 Auditing is another mechanism used for the detection and recovery from key compromises.  
1549 Auditing includes reviewing the actions of humans that use, operate and maintain systems,  
1550 looking for unusual events that may indicate inappropriate actions by the humans or  
1551 processes using a key management system.

1552

1553

1554

## SECTION 6: OTHER ISSUES

1555 The use of cryptography should not be undertaken without a thorough risk analysis, and a  
1556 determination of the sensitivity of the information to be protected and the security controls  
1557 to be used (see [SP 800-175A](#) and [SP 800-53](#)). After performing a risk assessment and  
1558 determining the sensitivity level of the information to be protected (Low, Moderate or  
1559 High) and the security controls to be used, a number of issues need to be addressed to  
1560 ensure that cryptography is used properly.

1561 This section identifies issues to be addressed after determining that cryptography is  
1562 required.

### 1563 6.1 Required Security Strength

1564 The minimum security strength is determined by the sensitivity level of the information  
1565 (see [SP 800-175A](#)). [SP 800-152](#) requires a security strength of at least 112 bits for the  
1566 protection of Low-impact information, 128 bits for Moderate-impact information, and 192  
1567 bits for High-impact information. The required security strength can then be used to  
1568 determine the algorithm and key size to be used. Section 6 of [SP 800-57, Part 1](#) provides  
1569 tables for selecting appropriate algorithms and key sizes.

### 1570 6.2 Interoperability

1571 Interoperability is the ability of one entity to communicate with another entity, whether the  
1572 entities are people, devices or processes. In order to communicate, the entities must have:

- 1573 • A communications channel (e.g., the Internet) and the same communications  
1574 protocol (e.g., TLS), and
- 1575 • Policies that allow the entities to communicate.

1576 In order to communicate securely, the entities must also have:

- 1577 • Trust that each entity will enforce its own policies.
- 1578 • Interoperable cryptographic capabilities as discussed in [Section 4](#), and
- 1579 • Share appropriate keying material that has been established securely (see [Section](#)  
1580 [5.3](#)).

1581 For example, if entities A and B are in two different organizations, and

- 1582 • Each organization has a policy that allows the entities to communicate,
- 1583 • Each entity trusts that the other entity will enforce its own policies,
- 1584 • There is a TLS capability that can be used for communication,
- 1585 • Each entity can encrypt and decrypt information using AES with a 128-bit key and  
1586 establish keys using 2048-bit RSA key transport (see [Section 5.3.4](#)), and
- 1587 • One of the entities can generate a 128-bit AES key and act as the sender in the key-  
1588 transport scheme, and the other entity has a 2048-bit RSA key pair and can act as  
1589 the receiver (see [Section 5.3.4.2](#) for a discussion on key transport),



1590 then the two entities have a secure and interoperable communication channel that can be  
1591 used to establish a 128-bit key for encrypting information using AES.

### 1592 **6.3 When Algorithms are no Longer Approved**

1593 In the case that an algorithm is no longer **approved** for providing adequate protection (e.g.,  
1594 the algorithm may have been “broken”), any information protected by the algorithm could  
1595 be re-protected using an **approved** algorithm that is expected to protect the information for  
1596 the remainder of its security life. However, if the information protected using the no-  
1597 longer-approved algorithm was already collected by an adversary, the security of the re-  
1598 protected information may not be as desired (see Section 5.6.4 for [SP 800-57, Part 1](#) for  
1599 additional discussion).

### 1600 **6.4 Registration Authorities (RAs)**

1601 As discussed in [Section 5.2.3.1](#), an RA verifies the identity of users applying for a  
1602 certificate and authenticates other information to be included in a certificate generated by  
1603 a Certification Authority (CA). The correctness of this information is the linchpin on which  
1604 the security of using certificates is based. Once this information is verified, the appropriate  
1605 information is submitted to a CA for certificate generation using a signed certification  
1606 request. The CA must deem the RA as trustworthy, e.g.,

- 1607 • Appropriate identification is provided by an entity requesting a certificate and is  
1608 fully checked by the RA;
- 1609 • Information submitted for inclusion in the certificate is checked for validity (e.g.,  
1610 that the public key is valid, and the private key is in the possession of the claimed  
1611 owner); and
- 1612 • The RA provides adequate protection for the private key used to sign the  
1613 certification request.

### 1614 **6.5 Cross Certification**

1615 Cross certification is the establishment of a trust relationship between two [Certification](#)  
1616 [Authorities](#) (CAs) through the signing of each other's [public key](#) in a [certificate](#) referred  
1617 to as a "cross-certificate." Cross-certificates provide a means to create a chain of trust  
1618 from a single, trusted, root CA to multiple other CAs so that subscribers in one CA  
1619 domain can interact safely with subscribers in other CA domains (e.g., the subscriber in  
1620 one CA domain has assurance of the identity of the subscriber in the other domain and  
1621 assurance of the accurateness of the other information provided by his certificate).

1622 Cross certification should only be performed when each CA examines the other CA's  
1623 policies, finds them acceptable and trusts that CA to operate in accordance with those  
1624 policies.



1625

### Appendix A: References

1626 The following FIPS and NIST Special Publications (SP) apply to the use of cryptography  
1627 in the Federal Government.

1628 All publications are available at <http://csrc.nist.gov/publications>.  
1629

FIPS 140	<p>Federal Information Processing Standard 140-2, <i>Security Requirements for Cryptographic Modules</i>, May 2001.</p> <p>FIPS 140-2 specifies the requirements that must be met by cryptographic modules protecting U.S. Government information. The standard provides four increasing, qualitative levels of security. The security requirements cover areas related to the secure design and implementation of a cryptographic module.</p>
FIPS 180	<p>Federal Information Processing Standard 180-4, <i>Secure Hash Standard (SHS)</i>, August 2015.</p> <p>FIPS 180-4 specifies seven cryptographic hash algorithms: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256.</p>
FIPS 185	<p>Federal Information Processing Standard 185, <i>Escrowed Encryption Standard</i>, February 1994, Withdrawn in October 2015.</p> <p>FIPS 185 specified the use of an encryption/decryption algorithm and a Law Enforcement Access Field (LEAF) creation method that could be implemented in electronic devices and used for protecting government telecommunications when such protection was desired. The algorithm and the LEAF creation method were classified. The LEAF was intended for use in a key escrow system that provided for the decryption of telecommunications when access to the telecommunications was lawfully authorized.</p>
FIPS 186	<p>Federal Information Processing Standard 186-4, <i>Digital Signature Standard (DSS)</i>, July 2013.</p> <p>FIPS 186-4 specifies a suite of algorithms that can be used to generate a digital signature: DSA, ECDSA and RSA. This Standard includes methods for the generation of digital signatures, methods for the generation of domain parameters (for DSA and ECDSA), and methods for the generation of key pairs, and requires certain assurances for using digital signatures: assurance of domain-parameter validity (DSA and ECDSA), and assurance of public-key validity and assurance of private-key possession for all three algorithms.</p>

FIPS 197	<p>Federal Information Processing Standard 197, <i>Advanced Encryption Standard (AES)</i>, November 2001.</p> <p>FIPS 197 specifies a symmetric key block cipher algorithm. The Standard supports key sizes of 128, 192, and 256 bits and a block size of 128 bits.</p>
FIPS 198	<p>Federal Information Processing Standard 198-1, <i>Keyed-Hash Message Authentication Code (HMAC)</i>, published in July 2008.</p> <p>FIPS 198-1 defines a message authentication code (MAC) that uses a cryptographic hash function in conjunction with a secret key for the calculation and verification of the MACs.</p>
FIPS 199	<p>Federal Information Processing Standard 199, <i>Standards for Security Categorization of Federal Information and Information Systems</i>, February 2004.</p> <p>FIPS 199 establishes security categories for both information and information systems. The security categories are based on the potential impact on an organization if certain events occur that jeopardize the information and information systems needed by the organization to accomplish its assigned mission, protect its assets, fulfill its legal responsibilities, maintain its day-to-day functions, and protect individuals.</p>
FIPS 202	<p>Federal Information Processing Standard 202, <i>SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions</i>, August 2015.</p> <p>FIPS 202 specifies SHA3-224, SHA3-256, SHA3-384 and SHA3-512. This FIPS also specifies two extendable-output functions (SHAKE128 and SHAKE256), which are not, in themselves, considered to be hash functions.</p>
SP 800-22	<p>Special Publication 800-22, <i>A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications</i>, April 2010.</p> <p>SP 800-22 discusses some aspects of selecting and testing random and pseudorandom number generators for providing random numbers that are indistinguishable from truly random output.</p>
SP 800-32	<p>Special Publication 800-32, <i>Federal Agency Use of Public Key Technology for Digital Signatures and Authentication</i>, February 2001.</p>

	<p>SP 800-32 was developed to assist agency decision-makers in determining if a PKI is appropriate for their agency, and how PKI services can be deployed most effectively within a Federal agency. It is intended to provide an overview of PKI functions and their applications.</p>
SP 800-38	<p>A series of publications specifying modes of operation for block cipher algorithms.</p>
SP 800-38A	<p>Special Publication 800-38A, <i>Recommendation for Block Cipher Modes of Operation - Methods and Techniques</i>, December 2001.</p> <p>SP 800-38A defines five confidentiality modes of operation for use with an underlying symmetric key block cipher algorithm: Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). Used with an <b>approved</b> underlying block cipher algorithm (i.e., AES and TDEA), these modes can provide cryptographic protection for sensitive, but unclassified, computer data.</p>
SP 800-38B	<p>Special Publication 800-38B, <i>Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication</i>, May 2005.</p> <p>SP 800-38B specifies a message authentication code (MAC) algorithm based on a symmetric key block cipher (i.e., AES or TDEA). This block cipher-based MAC algorithm, called CMAC, may be used to provide assurance of the source and integrity of binary data.</p>
SP 800-38C	<p>Special Publication 800-38C, <i>Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality</i>, May 2004.</p> <p>SP 800-38C defines a mode of operation, called CCM, for a symmetric-key block cipher algorithm with a 128-bit block size (i.e., AES). CCM may be used to provide assurance of the confidentiality and the authenticity of computer data by combining the techniques of the Counter (CTR) mode specified in <a href="#">SP 800-38A</a>, and the Cipher Block Chaining-Message Authentication Code (CBC-MAC) algorithm (specified in <a href="#">SP 800-90B</a>, but not currently approved for general use).</p>
SP 800-38D	<p>Special Publication 800-38D, <i>Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC</i>, November 2007.</p> <p>SP 800-38D specifies the Galois/Counter Mode (GCM), an algorithm for authenticated encryption with associated data, and its</p>

	<p>specialization, GMAC, for generating a message authentication code (MAC) on data that is not encrypted. GCM and GMAC are modes of operation for an underlying, <b>approved</b> symmetric-key block cipher with a 128-bit block size (i.e., AES).</p>
SP 800-38E	<p>Special Publication 800-38E, <i>Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices</i>, January 2010.</p> <p>SP 800-38E approves the XTS-AES mode of the AES algorithm by reference to <a href="#">IEEE 1619</a>, subject to one additional requirement, as an option for protecting the confidentiality of data on storage devices. The mode does not provide authentication of the data or its source.</p>
SP 800-38F	<p>Special Publication 800-38F, <i>Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping</i>, December 2012.</p> <p>SP 800-38F describes cryptographic methods that are approved for key wrapping. In addition to approving existing methods, this publication specifies two new, deterministic authenticated-encryption modes of operation of the Advanced Encryption Standard (AES) algorithm: the AES Key Wrap (KW) mode and the AES Key Wrap with Padding (KWP) mode. An analogous mode with the Triple Data Encryption Algorithm (TDEA) as the underlying block cipher, called TKW, is also specified to support legacy applications.</p>
SP 800-38G	<p>Special Publication 800-38G, DRAFT <i>Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption</i>, July 2013.</p> <p>SP 800-38G specifies methods for format-preserving encryption, called FF1 and FF3. Each of these methods is a mode of operation of the AES algorithm, which is used to construct a round function within the Feistel structure for encryption.</p>
SP 800-52	<p>Special Publication 800-52, <i>Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations</i>, April 2014.</p> <p>Transport Layer Security (TLS) provides mechanisms to protect sensitive data during electronic dissemination across the Internet. SP 800-52 provides guidance about the selection and configuration of TLS protocol implementations, while making effective use of Federal Information Processing Standards (FIPS) and NIST-recommended cryptographic algorithms (specified in SPs), and requires that TLS 1.1 be configured with FIPS-based cipher suites as the minimum appropriate secure transport protocol. This</p>

	<p>publication also identifies TLS extensions for which mandatory support must be provided and identifies other recommended extensions.</p>
SP 800-53	<p>Special Publication 800-53, Rev. 4, <i>Security and Privacy Controls for Federal Information Systems and Organizations</i>, April 2013.</p> <p>SP 800-53 provides a catalog of security and privacy controls for federal information systems and organizations, and a process for selecting controls to protect organizational operations (including mission, functions, image, and reputation), organizational assets, individuals, other organizations, and the Nation from a diverse set of threats, including hostile cyber attacks, natural disasters, structural failures, and human errors.</p>
SP 800-56A	<p>Special Publication 800-56A, <i>Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography</i>, May 2013.</p> <p>SP 800-56A specifies key-establishment schemes based on the discrete logarithm problem over finite fields and elliptic curves, including several variations of Diffie-Hellman and Menezes-Qu-Vanstone (MQV) key establishment schemes.</p>
SP 800-56B	<p>Special Publication 800-56B, <i>Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography</i>, September 2014.</p> <p>SP 800-56B specifies key-establishment schemes using integer-factorization cryptography (RSA). Both key transport and key agreement schemes are specified.</p>
SP 800-56C	<p>Special Publication 800-56C, <i>Recommendation for Key Derivation through Extraction-then-Expansion</i>, November 2011.</p> <p>SP 800-56C specifies techniques for the derivation of keying material from a shared secret established during a key-establishment scheme defined in SP 800-56A or SP 800-56B through an extraction-then-expansion procedure.</p>
SP 800-57, Part 1	<p>Special Publication 800-57, Part 1, <i>Recommendation for Key Management: Part 1: General (Revision 3)</i>, January 2016.</p> <p>Part 1 of SP 800-57 provides general guidance and best practices for the management of cryptographic keying material. It focuses on issues involving the management of cryptographic keys: their generation, use, and eventual destruction. Related topics, such as algorithm selection and appropriate key size, cryptographic policy, and cryptographic module selection, are also included.</p>

SP 800-57, Part 2	<p>Special Publication 800-57, Part 2, <i>Recommendation for Key Management: Part 2: Best Practices for Key Management Organization</i>, August 2005.</p> <p>Part 2 of SP 800-57 provides guidance on policy and security planning requirements for U.S. government agencies. This part of SP 800-57 contains a generic key-management infrastructure, guidance for the development of organizational key-management policy statements and key-management practices statements, an identification of key-management information that needs to be incorporated into security plans for general support systems and major applications that employ cryptography, and an identification of key-management information that needs to be documented for all Federal applications of cryptography.</p>
SP 800-57, Part 3	<p>Special Publication 800-57, Part 3, <i>Implementation-Specific Key Management Guidance</i>, June 2015.</p> <p>Part 3 of SP 800-57 addresses the key management issues associated with currently available cryptographic mechanisms, such as the Public Key infrastructure (PKI), Internet Protocol Security (IPsec), the Transport Layer Security protocol (TLS), Secure/Multipart Internet Mail Extensions (S/MIME), Kerberos, Over-the-Air Rekeying (OTAR), Domain Name System Security Extensions (DNSSEC), Encrypted File Systems and the Secure Shell (SSH) protocol.</p>
SP 800-67	<p>Special Publication 800-67, <i>Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher</i>, January 2012.</p> <p>SP 800-67 specifies the Triple Data Encryption Algorithm (TDEA), including its primary component cryptographic engine, the Data Encryption Algorithm (DEA).</p>
SP 800-89	<p>Special Publication 800-89, <i>Recommendation for Obtaining Assurances for Digital Signature Applications</i>, November 2006.</p> <p>Entities participating in the generation or verification of digital signatures depend on the authenticity of the process. SP 800-89 specifies methods for obtaining the assurances necessary for valid digital signatures: assurance of domain parameter validity, assurance of public key validity, assurance that the key-pair owner actually possesses the private key, and assurance of the identity of the key pair owner.</p>
SP 800-90A	<p>Special Publication 800-90A, <i>Recommendation for Random Number Generation Using Deterministic Random Bit Generators</i>, June 2015.</p>

	<p>SP 800-90A specifies DRBG mechanisms for the generation of random bits using deterministic methods. The methods provided are based on either hash functions or block cipher algorithms and are designed to support selected security strengths. DRBGs must be initialized from a randomness source that provides sufficient entropy for the security strength to be supported by the DRBG.</p>
SP 800-90B	<p>Special Publication 800-90B, (DRAFT) <i>Recommendation for the Entropy Sources Used for Random Bit Generation</i>, January 2016.</p> <p>SP 800-90B specifies the design principles and requirements for the entropy sources used by Random Bit Generators, including health tests to determine that the entropy source has not failed and tests for the validation of entropy sources.</p>
SP 800-90C	<p>Special Publication 800-90C, (DRAFT) <i>Recommendation for Random Bit Generator (RBG) Constructions</i>, September 2013.</p> <p>SP 800-90C specifies constructions for the implementation of random bit generators (RBGs). An RBG may be a deterministic random bit generator (DRBG) or a non-deterministic random bit generator (NRBG). The constructed RBGs consist of DRBG mechanisms as specified <a href="#">SP 800-90A</a> and entropy sources as specified in <a href="#">SP 800-90B</a>.</p>
SP 800-102	<p>Special Publication 800-102, <i>Recommendation for Digital Signature Timeliness</i>, September 2009.</p> <p>Establishing the time when a digital signature was generated is often a critical consideration. A signed message that includes the (purported) signing time provides no assurance that the private key was used to sign the message at that time unless the accuracy of the time can be trusted. With the appropriate use of digital signature-based timestamps from a Trusted Timestamp Authority and/or verifier-supplied data that is included in the signed message, the signer can provide some level of assurance about the time that the message was signed.</p>
SP 800-106	<p>Special Publication 800-106, <i>Randomized Hashing for Digital Signatures</i>, February 2009.</p> <p>NIST-approved digital signature algorithms require the use of an <b>approved</b> cryptographic hash function in the generation and verification of signatures. SP 800-106 specifies a method to enhance the security of the cryptographic hash functions used in digital signature applications by randomizing the messages that are signed.</p>

SP 800-107	<p>Special Publication 800-107, <i>Recommendation for Applications Using Approved Hash Algorithms</i>, August 2012.</p> <p>Hash functions that compute a fixed-length message digest from arbitrary length messages are widely used for many purposes in information security. SP 800-107 provides security guidelines for achieving the required or desired security strengths when using cryptographic applications that employ the approved hash functions specified in <a href="#">FIPS 180</a>. These include functions such as digital signatures, Keyed-hash Message Authentication Codes (HMACs) and Hashed-based Key Derivation Functions (hash-based KDFs).</p>
SP 800-108	<p>Special Publication 800-108, <i>Recommendation for Key Derivation Using Pseudorandom Functions</i>, October 2009.</p> <p>SP 800-108 specifies techniques for the derivation of additional keying material from a secret key (i.e., a key-derivation key) using pseudorandom functions. The key-derivation key may have been either established through a key-establishment scheme or shared through some other manner (e.g., a manual key distribution).</p>
SP 800-130	<p>Special Publication 800-130, <i>A Framework for Designing Cryptographic Key Management Systems</i>, August 2013.</p> <p>SP 800-130 contains topics to be considered by a CKMS designer when developing a CKMS design specification. Topics include security policies, cryptographic keys and metadata, interoperability and transitioning, security controls, testing and system assurances, disaster recovery, and security assessments.</p>
SP 800-131A	<p>Special Publication 800-131A, <i>Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths</i>, November 2015.</p> <p>Section 5.6.4 of <a href="#">SP 800-57, Part 1</a> provides recommendations for transitioning to new cryptographic algorithms and key lengths because of algorithm breaks or the availability of more powerful computers that could be used to efficiently search for cryptographic keys. SP 800-131A offers more specific guidance for such transitions. Each algorithm and service is addressed in SP 800-131A, indicating whether its use is acceptable, deprecated, restricted, allowed only for legacy applications<sup>80</sup>, or disallowed.</p>

<sup>80</sup> The algorithm and key length may be used to process already-protected information, but there may be a risk in doing so.



<p>SP 800-132</p>	<p>Special Publication 800-132, <i>Recommendation for Password-Based Key Derivation Part 1: Storage Applications</i>, December 2010.</p> <p>SP 800-132 specifies techniques for the derivation of master keys from passwords or passphrases to protect stored electronic data or data protection keys.</p>
<p>SP 800-133</p>	<p>Special Publication 800-133, <i>Recommendation for Cryptographic Key Generation</i>, December 2012.</p> <p>SP 800-133 discusses the generation of the keys to be managed and used by the <b>approved</b> cryptographic algorithms.</p>
<p>SP 800-135</p>	<p>Special Publication 800-135, <i>Recommendation for Existing Application-Specific Key Derivation Functions</i>, December 2011.</p> <p>Many widely-used internet security protocols have their own application-specific Key Derivation Functions (KDFs) that are used to generate the cryptographic keys required for their cryptographic functions. SP 800-135 provides security requirements for those KDFs.</p>
<p>SP 800-152</p>	<p>Special Publication 800-152, <i>A Profile for U. S. Federal Cryptographic Key Management Systems (CKMS)</i>, October 2015.</p> <p>SP 800-152 contains requirements for the design, implementation, procurement, installation, configuration, management, operation and use of a CKMS by and for U.S. federal organizations and their contractors. The Profile is based on NIST Special Publication SP 800-130.</p>
<p>SP 800-175A</p>	<p>Special Publication 800-175A, <i>Guideline for Using Cryptographic Standards in the Federal Government: Directives, Mandates and Policies</i>, NOT YET AVAILABLE.</p>
<p>NISTIR 7924</p>	<p>NIST Internal Report, DRAFT <i>Reference Security Policy</i>, May 2014.</p> <p>NIST 7924 is intended to identify a set of security controls and practices to support the secure issuance of certificates. It was written in the form of a Certificate Policy (CP), a standard format for defining the expectations and requirements of the relying party community that will trust the certificates issued by its Certificate Authorities (CAs).</p>

1630  
1631

Non-NIST Publications:

<p>IEEE 802.11</p>	<p>Wireless Local Area Networks.</p>
--------------------	--------------------------------------

IEEE P1363	IEEE P1363: Standard Specifications for Public-Key Cryptography, 2000.
IEEE P1363a	IEEE P1363a: Standard Specifications For Public Key Cryptography- Amendment 1: Additional Techniques, 2004.
IEEE P1363.1	Public-Key Cryptographic Techniques Based on Hard Problems over Lattices, 2008.
IEEE P1363.2	Password-Based Public-Key Cryptography, 2008.
IEEE P1619	Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices, 2008.
ISO/IEC 9594-8	ITU-T Recommendation X.509 (2005)   ISO/IEC 9594-8:2005, Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks.
ISO/IEC 9797-1	ISO/IEC 9797-1:2011, Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher, 2011.  This standard includes CMAC, as specified in <a href="#">SP 800-38B</a> .
ISO/IEC 9797-2	Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function, 2011.  This standard includes HMAC, as specified in <a href="#">FIPS 198</a> .
ISO/IEC 10116	Information technology -- Security techniques -- Modes of operation for an n-bit block cipher, 2006.  This standard includes all the modes specified in <a href="#">SP 800-38A</a> .
ISO/IEC 10118-3	Information technology -- Security techniques -- Hash-functions -- Part 3: Dedicated hash-functions, 2004.  This standard includes SHA-1 and the SHA-2 family of hash functions specified in <a href="#">FIPS 180</a> . A revision of ISO/IEC 10118-3 will include the SHA-3 functions specified in <a href="#">FIPS 202</a> .
ISO/IEC 11770-3	Information technology -- Security techniques -- Key management -- Part 3: Mechanisms using asymmetric techniques, 2008.  This standard specifies key establishment mechanisms, some of which can be instantiated with key establishment schemes specified in <a href="#">SP 800-56A</a> and <a href="#">SP 800-56B</a> .

ISO/IEC DIS 11770-6	<p>Information technology -- Security techniques -- Key management -- Part 6: Key derivation, 2015.</p> <p>This draft standard will include all key derivation functions specified in <a href="#">SP 800-108</a>, as well as the two-step key derivation methods specified in <a href="#">SP 800-56C</a>.</p>
ISO/IEC 11889	<p>Information technology -- Trusted Platform Module Library -- Part 1: Architecture, 2015.</p> <p>Information technology -- Trusted Platform Module -- Part 2: Design principles, 2009.</p> <p>Information technology -- Trusted Platform Module -- Part 3: Structures, 2009.</p> <p>Information technology -- Trusted Platform Module Library -- Part 4: Supporting Routines, 2015.</p>
ISO/IEC 14888-2	<p>Information technology -- Security techniques -- Digital signatures with appendix -- Part 2: Integer factorization based mechanisms, 2008.</p> <p>This standard includes RSA signatures, as specified in <a href="#">FIPS 186</a>.</p>
ISO/IEC DIS 14888-3	<p>Information technology -- Security techniques -- Digital signatures with appendix -- Part 3: Discrete logarithm based mechanisms, 2006.</p> <p>This draft standard will include DSA, as specified for finite fields and elliptic curves in <a href="#">FIPS 186</a>.</p>
ISO/IEC 18033-3	<p>Information technology -- Security techniques -- Encryption algorithms -- Part 3: Block ciphers, 2010.</p> <p>This standard includes 64-bit block ciphers: TDEA, MISTY1, CAST-128, HIGHT and 128-bit block ciphers: AES, Camellia, and SEED. TDEA is specified in <a href="#">SP 800-67</a> and AES is specified in <a href="#">FIPS 197</a>.</p>
ISO/IEC 19772	<p>Information technology -- Security techniques -- Authenticated encryption, 2009.</p> <p>This standard includes CCM (as specified in <a href="#">SP 800-38C</a>), GCM (as specified in <a href="#">SP 800-38D</a>), and Key wrapping (as specified in <a href="#">SP 800-38E</a>).</p>
PKCS 1	<p>Public Key Cryptography System 1, version 2.2, RSA Cryptography Standard, June 2002; available at <a href="http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm">http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-rsa-cryptography-standard.htm</a>.</p>

	<p>PKCS 1 provides recommendations for the implementation of public-key cryptography based on the RSA algorithm, covering cryptographic primitives, encryption schemes, signature schemes with appendix and the ASN.1 syntax for representing keys and for identifying the schemes.</p>
ISO/IEC 18033-3:2010	<p>Information technology -- Security techniques -- Encryption algorithms -- Part 3: Block ciphers, 2005.</p>
X9.31	<p>American National Standard for Financial Services X9.31, <i>Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)</i>, 1998; WITHDRAWN.</p> <p>ANS X9.31 defined a method for digital signature (signature) generation and verification for the protection of financial messages and data using reversible public key cryptography systems without message recovery. In addition, 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 were provided.</p>
X9.42	<p>American National Standard for Financial Services X9.42, <i>Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography</i>, 2001; WITHDRAWN.</p> <p>ANS X9.42, partially adapted from <a href="#">ISO 11770-3</a>, specifies schemes for the agreement of symmetric keys using Diffie-Hellman and MQV algorithms. It covers methods for domain parameter generation, domain parameter validation, key pair generation, public key validation, shared secret value calculation, key derivation, and test message authentication code computation for discrete logarithm problem based key agreement schemes.</p>
X9.44	<p>American National Standard for Financial Services X9.44, <i>Key Establishment Using Integer Factorization Cryptography</i>, 2007.</p> <p>ANS X9.44 specifies key-establishment schemes using public-key cryptography, based on the integer factorization problem. Two types of key-establishment schemes are specified: key transport and key agreement.</p>
X9.62	<p>American National Standard X9.62, <i>The Elliptic Curve Digital Signature Algorithm (ECDSA)</i>, 2005; available at <a href="http://x9.org">http://x9.org</a>.</p> <p>ANS X9.62 defines methods for digital signature (signature) generation and verification for the protection of messages and data using the Elliptic Curve Digital Signature Algorithm (ECDSA). This Standard provides methods and criteria for the generation of public and private keys that are required by ECDSA and the</p>

	<p>procedural controls required for the secure use of the algorithm with these keys. This ECDSA Standard also provides methods and criteria for the generation of elliptic-curve domain parameters that are required by ECDSA and the procedural controls required for the secure use of the algorithm with these domain parameters.</p>
<p>X9.63</p>	<p>American National Standard X9.63, <i>Key Agreement and Key Transport Using Elliptic Curve Cryptography</i>, 2005.</p> <p>ANS X9.63 defines key-establishment schemes that employ asymmetric cryptographic techniques. The arithmetic operations involved in the operation of the schemes take place in the algebraic structure of an elliptic curve over a finite field. Both key-agreement and key-transport schemes are specified.</p>

1632