Nuclear Regulatory Commission
Office of the Chief Information Officer
Computer Security Standard

Office Instruction:	**OCIO-CS-STD-2009**
Office Instruction Title:	**Cryptographic Control Standard**
Revision Number:	**2.0**
Issuance:	**Date of last signature below**
Effective Date:	**October 1, 2017**
Primary Contacts:	**Kathy Lyons-Burke, Senior Level Advisor for Information Security**
Responsible Organization:	**OCIO**
Summary of Changes:	OCIO-CS-STD-2009, "Cryptographic Control Standard," provides the minimum security requirements that must be applied to the Nuclear Regulatory Commission (NRC) systems which utilize cryptographic algorithms, protocols, and cryptographic modules to provide secure communication services. This update is based on the latest versions of the National Institute of Standards and Technology (NIST) Guidance and Federal Information Processing Standards (FIPS) publications, Committee on National Security System (CNSS) issuances, and National Security Agency (NSA) requirements.
Training:	Upon request
ADAMS Accession No.:	ML17024A095

Approvals			
Primary Office Owner	Office of the Chief Information Officer	**Signature**	**Date**
Enterprise Security Architecture Working Group Chair	Kathy Lyons-Burke		09/26/17
CIO	David Nelson **/RA/**		09/26/17
CISO	Jonathan Feibus		09/26/17

TABLE OF CONTENTS

List of Tables

Computer Security Standard
OCIO-CS-STD-2009

Cryptographic Control Standard

1 PURPOSE

OCIO-CS-STD-2009, "Cryptographic Control Standard," provides the minimum security requirements that must be applied to all Nuclear Regulatory Commission (NRC) systems processing information up to and including, the classified level, which use cryptographic algorithms, protocols, or cryptographic modules.

APPENDIX A provides information to help determine if cryptography is required.

This standard is intended for system administrators and Information System Security Officers (ISSOs).

2 INTRODUCTION

The security requirements specified in this standard relate to cryptographic modules, algorithms, and key management systems of the NRC system environments. The following are high-level concepts and terminology associated with cryptography and provide the foundation for the requirements specified in Section 3, General Requirements, and Section 4, Specific Requirements, of this standard.

2.1 Overview of Cryptography and Cryptographic Systems

Cryptography is an "art or science concerning the principles, means, and methods for rendering plain information unintelligible and for restoring encrypted information to intelligible form."[1] Encryption is "the cryptographic transformation of data to produce ciphertext."[2] In order to encrypt plaintext and produce ciphertext, encryption uses a defined algorithm (also known as a cipher) to make information unintelligible and generate the ciphertext.

A cryptographic system (cryptosystem) refers to the "associated information security (INFOSEC) items interacting to provide a single means of encryption or decryption"[3] used to produce ciphertext via encryption or to obtain plaintext from ciphertext via decryption. A cryptographic system requires three components:

1. Plaintext data/information to encrypt;
2. Method to encrypt the data using a cryptographic algorithm; and
3. Encryption (cryptographic) keys to use in conjunction with the data and the algorithm.

1 Committee for National Security Systems Instruction (CNSSI) 4009, "National Information Assurance (IA) Glossary."
2 Ibid.
3 Ibid.

Most modern programming languages provide libraries with a wide range of available cryptographic algorithms, such as the Advanced Encryption Standard (AES). Choosing the right algorithm involves evaluating security, performance, and compliance requirements specific to any particular application. Just as the selection of an encryption algorithm is important, protecting the keys from unauthorized access is critical. Often, a Key Management Infrastructure (KMI) is used to manage the security of encryption keys. A KMI is the "framework and services that provide the generation, production, storage, protection, distribution, control, tracking, and destruction for all cryptographic keying material, symmetric keys as well as public keys and public key certificates."[4]

A KMI is comprised of the following functional elements or nodes for the generation, distribution, and management of cryptographic keys:

1. A central oversight authority
2. Key processing facility(ies)
3. Service agents, and
4. Client nodes.

Additional information on KMIs is provided in National Institute of Standards and Technology (NIST) Special Publication (SP) 800-57, "Recommendation for Key Management – Part 2: Best Practices for Key Management Organization."

A common way to protect keys in a KMI is to use a cryptographic module. A cryptographic module is "the set of hardware, software, and/or firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary."[5] With respect to KMI, a cryptographic module typically provides tamper evidence or resistance to protect keys from unauthorized use.

Cryptography algorithms, modules, and key management systems are used to protect sensitive data.

2.2 Cryptographic Algorithms

A cryptographic algorithm is "a well-defined computational procedure that takes variable inputs, including a cryptographic key, and produces an output."[6] Federal Information Processing Standards (FIPS)-validated cryptographic algorithms are used to secure data and communication services.

The NIST has established a Cryptographic Algorithm Validation Program (CAVP), in which cryptographic algorithm testing is conducted to ensure that a specific algorithm is implemented and functions correctly. An algorithm implementation has to meet all the requirements of its associated FIPS Publication (PUB), such as FIPS PUB 197, "Advanced Encryption Standard," before it can be listed as "Approved" on the NIST validation certificate.

[4] Committee for National Security Systems Instruction (CNSSI) 4009, "National Information Assurance (IA) Glossary."
[5] FIPS PUB 140-2, "Security Requirements for Cryptographic Modules."
[6] CNSSI-4009.

2.2.1 Types of Cryptographic Algorithms

There are three types of cryptographic algorithms:

1. Secret Key Cryptography
2. Public Key Cryptography
3. Hash Functions

2.2.1.1 Secret Key Cryptography

Secret key cryptography uses symmetric-key algorithms where a single secret key is used for both encryption and decryption. Symmetric-key algorithms are used for:

- Confidentiality: Ensuring that only those that know the secret key can see the data.
- Integrity: Ensuring that any modification of the data is detected when decrypting the data.

One of the approved symmetric-key algorithms for encryption/decryption is the AES. AES is a block-cipher algorithm, which operates on blocks of data during encryption/decryption operations. The AES algorithm encrypts and decrypts information in 128-bit blocks while using 128, 196, or 256-bit keys, which are specified within FIPS PUB 197. The block-cipher algorithm is used in conjunction with block cipher modes-of-operation to safeguard against unauthorized data interception. The most commonly used block cipher mode is Cipher Block Chaining (CBC), with Electronic Codebook (ECB), Cipher Feedback (CFB), Counter (CTR), and Output Feedback (OFB) as other options. Refer to NIST SP 800-38A, “Recommendation for Block Cipher Modes of Operation,” for details of approved modes.

Symmetric keys are most often known by more than one entity; however, the key should not be disclosed to entities that are not authorized to access the data protected by that algorithm and key.

2.2.1.2 Public Key Cryptography

Public key cryptography uses asymmetric-key algorithms, which use two related keys, a public key and a private key. The two keys are such that deriving the private key from the public key is computationally not feasible. The public key may be known by many, whereas, the private key is under the sole control of the key pair owner. Even though the public and private keys of a key pair are related, knowledge of the public key does not reveal the private key. The current FIPS-approved asymmetric algorithms are the Digital Signature Algorithm (DSA), RSA Algorithm, and Elliptic Curve Digital Signature Algorithm (ECDSA). Asymmetric-key algorithms are used for:

- Confidentiality: Ensuring that only the owner of the key pair can see the information when the owner’s public key is used to encrypt the data.
- Integrity: Addressing the unauthorized or accidental modification of information, which includes insertion, deletion, and modification. To ensure data integrity, the system must be able to detect unauthorized modification. The goal is to verify that the information has not been altered at the receiving end.

- Authentication: Establishing the validity of a transmission, message, or an originator. Therefore, this service applies to both individuals and the information itself. The goal is for the receiver of the information to determine its origin.
- Non-repudiation: Preventing an individual from denying that previous actions have been performed. The goal is to ensure that the recipient of the information is assured of the sender's identity.

2.2.1.3 Hash Functions

A hash function takes an input of arbitrary length and outputs a fixed-length value, often known as a hash value (aka message digest). Cryptographic hash functions do not require keys. Many algorithms and schemes that provide a security service use a hash function as a component of the algorithm. Hash functions can be found in digital signature algorithms, hash-based message authentication code (HMAC), and random number generators. These algorithms are used to determine the integrity of a message and protect against unauthorized modification. If there is any change in the message during data transmission, it is highly probable that the resulting hash value will be different. This fact provides reliable means of generation and verification of digital signatures and message authentication codes (MACs), and in the generation of random numbers or bits.

There are three FIPS-approved Secure Hash Algorithms (SHA) families: SHA-1, SHA-2, and SHA-3. The SHA-1, SHA-2, and SHA-3 families specify one-way hash functions that can process a message to produce a condensed representation called a hash value. SHA-1 is disallowed by NIST (except for generating digital signatures on ephemeral parameters) for digital signature generation and other applications that require collision resistance for federal applications due to inherent vulnerabilities.[7] FIPS PUB 180-4, "Secure Hash Standard," and FIPS PUB 202, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," specify the approved hash functions. Both SHA-2 and SHA-3 are considered stronger than SHA-1. SHA-2 and SHA-3 are considered to have similar security and are intended to be alternates where both can be used.

2.2.2 Message Authentication Code

A MAC is a "cryptographic checksum on data that uses a symmetric key to detect both accidental and intentional modifications of the data."[8] Use of a MAC enables detection of modification of the data and ensures the recipient that only someone that knows the secret key could have sent the message.

2.2.3 Hash-based Message Authentication Code

An HMAC is a "message authentication code that uses a cryptographic key in conjunction with a hash function."[9]

HMACs use an approved hash function, a secret key, and the information to be hashed. Keyed-hash functions are also used in challenge-response identification protocols for computing

[7] http://csrc.nist.gov/groups/ST/hash/policy.html
[8] CNSSI-4009.
[9] Ibid.

responses, which are a function of both a secret key and a challenge message. MAC and HMAC SHA-1 and HMAC SHA-2 use keyed-hashing algorithms.

The message sender uses an HMAC function to produce a value formed by condensing the secret key and the challenge message input. The MAC is sent to the message receiver along with the message. The receiver computes the MAC on the received message, using the same secret key and HMAC function. If the two values match, the message has been correctly received, and the receiver is assured that the sender is a member of the authorized users group that share the key. A variety of key sizes is allowed for HMAC, where the key size depends on the level of security and the choice of hash function. FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code (HMAC)," defines all HMAC implementations and key size requirements.

2.2.4 Digital Signatures

A digital signature is a type of electronic signature. A digital signature is a "result of a cryptographic transformation of data that, when properly implemented, provides the services of: 1. origin authentication, 2. data integrity, and 3. signer non-repudiation."[10] The digital signature is used to prove to the recipient or a third party that the originator signed the message that was received. Verification of a digital signature also verifies that the information was not modified since the signature was generated. Digital signatures may also be generated for data at rest so that the integrity of the data may be verified at a later time. Digital signatures authenticate the integrity of the signed data and the identity of the signatory. Signature generation uses a private key to generate a digital signature, and signature verification uses a public key that corresponds to the private key to verify the signature. The signatory owns both the public and private keys used in the process.

A hash function is used in the signature generation process to obtain a hash value. The resulting hash value is used with the digital signature algorithm to generate the digital signature. The digital signature is sent with the signed data to the recipient. The recipient verifies the message and signature by using the signatory's public key. The same hash function and digital signature algorithm is used in the verification process. Similar procedures generate and verify signatures for encrypted data in transit, as well as data at rest. FIPS PUB 186-4, "Digital Signature Standard (DSS)," provides detailed requirements for digital signature applications.

2.3 Cryptographic Modules

A cryptographic module is the set of hardware, software, and/or firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary. A cryptographic boundary is explicitly defined by the physical bounds of a cryptographic module that contains all the hardware, software, and/or firmware components of a cryptographic module.

FIPS PUB 140-2, "Security Requirements for Cryptographic Modules," specifies the security requirements for a cryptographic module used to protect sensitive information within information technology systems. A FIPS PUB 140-2 compliant cryptographic module is used to implement at least one approved security function in an approved mode of operation. A cryptographic module used to secure classified information has to be validated to comply with National Security Agency (NSA) specific requirements, and systems that process Sensitive

[10] CNSSI-4009.

Compartmented Information (SCI) must adhere to Director of National Intelligence (DNI) requirements.

Each of the security levels specified for a cryptographic module offers increased security over the preceding level. FIPS PUB 140-2 validations are determined at these security levels individually, with Level 1 being the lowest and Level 4 being the highest. The key difference in making this determination is the way physical and logical access to the cryptographic module is limited to ensure its integrity at each level. The Cryptographic Module Validation Program (CMVP) certificate indicates the security level; however, the strength and functionality of the cryptography is the same for each level, as per selection.

NIST established the CMVP that validates cryptographic modules to FIPS PUB 140-2 and other FIPS cryptography-based standards. Cryptographic module vendors use independent, accredited Cryptographic and Security Testing (CST) laboratories to test their modules and obtain validation certificates. The Computer Security Division at NIST serves as the Validation Authority for the program, validating the test results. Validated cryptographic module listings are placed at the CMVP website (http://www.nist.gov/cmvp).

2.4 Cryptographic Key Management

Cryptographic key management encompasses the entire lifecycle of cryptographic keys used by a cryptographic module. This includes random number generation, key generation, key establishment (including key transport), key entry/output, key storage, key retirement (retaining keys for decryption of data after retirement), and key destruction.

Cryptographic key management is an essential part of the effective use of cryptography for security. Cryptographic keys can be envisioned as the combination on a safe. If the combination is weak and can be guessed by the adversaries, then the strongest safe can provide no protection against penetration. Similarly, poor key management may easily compromise strong algorithms. All keys need to be protected against unauthorized substitution and modification. Key management provides the foundation for protecting cryptographic keys from unauthorized disclosure, modification, and substitution.

2.4.1 Cryptographic Key Establishment

Cryptographic key establishment is a "stage in the lifecycle of keying material; the process by which cryptographic keys are securely distributed among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key transport and/or key agreement protocols), or a combination of automated and manual methods (consists of key transport plus key agreement)."[11]

Automated cryptographic key-establishment schemes are used to set up keys for two communicating entities:

- Key Transport: "Secure transport of cryptographic keys from one cryptographic module to another module. When used in conjunction with a public key (asymmetric) algorithm, keying material is encrypted using a public key and subsequently decrypted using a

[11] NIST SP 800-57, Part 2.

private key. When used in conjunction with a symmetric algorithm, key transport is known as key wrapping."[12]

- Key Agreement: "A key-establishment procedure where resultant keying material is a function of information contributed by two or more participants, so that no party can predetermine the value of the keying material independently of the other party's contribution." [13]

  If an asymmetric-key algorithm is used, each entity has either a static key pair or an ephemeral key pair, or both. If a symmetric-key algorithm is used, each entity shares the same key-wrapping key.

Approved key-establishment schemes are defined in NIST SP 800-56A Rev 2, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography," and 56B, Rev 1, "Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography." NIST SP 800-56A specifies key-establishment schemes that use discrete-logarithm-based public key algorithms and NIST SP 800-56B provide key-establishment schemes that use integer-factorization-based public key algorithms.

Key-establishment protocols use key-establishment schemes to specify the processing necessary to establish a key. However, key-establishment protocols also specify message flow and format. The key-establishment protocols ensure:

- Protocols do not provide for an early exit from the protocol upon detection of a single error;
- Protocols trigger an alarm after a certain reasonable number of detected error conditions; and
- Key-dependent computations are obscured from the observer in order to prevent or minimize the detection of key-dependent characteristics.

2.4.2 Cryptographic Key Usage

Cryptographic keys are used in many processes, such as encryption, authentication, random number generation, or digital signatures. If the same key is used for more than one purpose, it may result in an undesirable outcome if the key is compromised. The following should be kept in mind when selecting cryptographic key usage in multiple applications:

- The use of the same key for two different cryptographic processes may weaken the security provided by one or both of the processes.
- Limiting the use of a key limits the damage that could be done if the key is compromised.
- Some uses of keys interfere with each other. For example, the longevity requirements for the private key-transport key and for the private digital-signature key will contradict each other, due to the fact that the first has to be kept for encryption/decryption purposes beyond its cryptoperiod and the second has to be destroyed at the expiration of its cryptoperiod to prevent its compromise. Another example is the need to escrow encryption keys to enable decryption of organization data if the owner of the key is not

[12] Ibid.
[13] CNSSI-4009.

available. If the same key is used for authentication, then the escrow key could be used to impersonate the owner.

2.4.3 Cryptoperiod

Per NIST SP 800-57, Part 2: "A cryptoperiod is the time span during which a specific key is authorized for use by legitimate entities, or the keys for a given system will remain in effect." A suitably defined cryptoperiod:

1. Limits the amount of information protected by a given key that is available for cryptanalysis,
2. Limits the amount of exposure if a single key is compromised,
3. Limits the use of a particular algorithm (e.g., to its estimated effective lifetime),
4. Limits the time available for attempts to penetrate physical, procedural, and logical access mechanisms that protect a key from unauthorized disclosure,
5. Limits the period within which information may be compromised by inadvertent disclosure of keying material to unauthorized entities, and
6. Limits the time available for computationally intensive cryptanalytic attacks (in applications where long-term key protection is not required)."

NIST SP 800-57, Part 1 specified risk factors that should be taken into account when defining a cryptoperiod include:

1. The strength of the cryptographic mechanisms (e.g., the algorithm, key length, block size, and mode of operation),
2. The embodiment of the mechanisms (e.g., a [FIPS140] Level 4 implementation or a software implementation on a personal computer),
3. The operating environment (e.g., a secure limited-access facility, open office environment, or publicly accessible terminal),
4. The volume of information flow or the number of transactions,
5. The security life of the data,
6. The security function (e.g., data encryption, digital signature, key derivation, or key protection),
7. The re-keying method (e.g., keyboard entry, re-keying using a key loading device where humans have no direct access to key information, or remote re-keying within a PKI),
8. The key update or key-derivation process,
9. The number of nodes in a network that share a common key,
10. The number of copies of a key and the distribution of those copies,
11. Personnel turnover (e.g., CA system personnel),
12. The threat to the information from adversaries (e.g., whom the information is protected from, and what are their perceived technical capabilities and financial resources to mount an attack), and

13. The threat to the information from new and disruptive technologies (e.g., quantum computers).

A key uses an algorithm to create ciphertext from plaintext and to decipher it on the receiving end. Once the cryptoperiod ends, the key is no longer available for either encryption or decryption. A properly defined cryptoperiod, as an example, limits the amount of exposure if a single key is compromised or an attempt is made for cryptanalysis. If specific data is sensitive for a very long time, the strength of the key used to protect that information must be much greater than if the data is only sensitive for a short period of time.

2.5 Cryptographic Algorithm and Key Size Selection

Key-size factors are likely to be less important when very strong cryptography is used since attackers go after the weakest link. When using strong cryptography, physical, procedural, and logical access protections will often be easier to break than the cryptography.

Strong cryptographic algorithms mitigate security issues other than just brute force cryptographic attacks. However, some unintentional implementations of these algorithms may leak small amounts of information about the key. In this case, the larger key may reduce the likelihood that this leaked information will eventually compromise the key. When selecting a block-cipher cryptographic algorithm (e.g., AES), the block size is also an important factor as the amount of security is dependent on the block size.

Table 2.5-1 identifies the approved cryptographic algorithms with key sizes of 128 bits of security strength or above for federal system applications.

The following defines the information contained within the columns of Table 2.5-1:

- Bits of Security: Indicates the number of bits of security provided by the algorithm and key sizes given in that row.
- Symmetric-Key Algorithm: Identifies the symmetric-key algorithms that provide the indicated level of security.
- FFC: Indicates the minimum size of the parameters associated with the standards that use Finite-Field Cryptography (FFC) (e.g., algorithms such as DSA and Diffie-Hellman [DH]), where the public key size is denoted by L and private key by N.
- IFC: Indicates the value for K (the size of the modulus n) for algorithms (e.g., RSA) based on Integer-Factorization Cryptography (IFC). The value of K is taken as the key size.
- ECC: Indicates the range for algorithms (e.g., ECDSA) based on elliptic-curve cryptography (ECC) that is specified for digital signatures. The ECC value is commonly taken as key size.

Table 2.5-1: Cryptographic Algorithms and Key Size Strength

Bits of Security	Symmetric-Key Algorithm	FFC (for DSA, DH – L=Public key, N=Private key)	IFC (for RSA)	ECC (for ECDSA)
128	AES-128	L=3072 N=256	K=3072	f=256-383
192	AES-192	L=7680 N=384	K=7680	f=384-511

Bits of Security	Symmetric-Key Algorithm	FFC (for DSA, DH – L=Public key, N=Private key)	IFC (for RSA)	ECC (for ECDSA)
256	AES-256	L=15360 N=512	K=15360	f=512+

Table 2.5-2 identifies the hash functions for security strengths of 128-bit and above. These hash functions are also used for the generation of digital signatures and HMAC values, and for deriving keys using key-derivation functions and random number generation.

Table 2.5-2: Hash Functions for Required Security Strength

Bits of Security	Digital Signatures and Hash-Only Applications	HMAC	Key Derivation Functions	Random Number Generation
128	SHA-256 SHA-512/256 SHA3-256	SHA-1	SHA-1	SHA-1
192	SHA-384 SHA3-384	SHA-224 SHA-512/224	SHA-224 SHA-512/224	SHA-224 SHA-512/224
256	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

2.6 Cryptographic Key Management System

The Cryptographic Key Management System (CKMS) provides administrators with the ability to centrally manage the lifecycle of all cryptographic keys across a range of encryption platforms.

Cryptography is used to protect sensitive data. This protection is provided by using specifically designed algorithms and cryptographic keys, which are managed by the CKMS. The cryptographic algorithm and various size keys provide the level of protection or security strength. The keys are either static or ephemeral, where static keys are regarded as long-term, multi-use keys and ephemeral keys are short-term, and single use keys are generated when needed. NIST SP 800-152, “A Profile for U.S. Federal Cryptographic Key Management System,” provides recommendations for such system designs and implementation. NIST SP 800-57 (all parts), provides key management policy and practice guidance. Physical protection of a cryptographic key management system is equally important to ensure authorized user access and use only by the designated system operators.

2.7 Public Key Infrastructure

A Public Key Infrastructure (PKI) enables users to securely and privately exchange information via a public and private cryptographic key pair that is obtained and shared through a trusted authority. The PKI binds public keys to an entity via a digital certificate digitally signed by a trusted entity and provides the capability for others to verify those findings. A PKI also provides services that can store, access, add, and revoke certificates. A PKI is an enabler of trust that provides strong user and other entity identification, confidential communication, data integrity,

and evidence for non-repudiation among individuals and entities that may or may not have had prior knowledge of each other.

The trust that PKI facilitates is enterprise-wide through distinct, yet integrated policies and technology components. These policies and components explicitly identify and determine the roles, responsibilities, constraints, range of use, and services available.

Certification Authorities (CAs) represent the people, processes, and tools to create digital certificates that securely bind a user's identity to the user's public keys. The following parties rely on the appropriate level of trust with respect to the creation and use of public key certificates:

- The individual subscriber or entity identified by the certificate,
- The CA who issues the certificate,
- The Registration or Validation Authority that provides identity verification and validation services in certain implementations, and
- The Relying Party (company, agency, or individual) relying on the certificate.

As long as users trust a CA and the CA's policies for issuing and managing certificates, the users can trust certificates issued by the CA. The CA investigates individuals and verifies their identity, binding that identity to the public key and verifying that the individual has the private key. The CA maintains and provides certificate status information for the life of the binding. Levels of trust are placed on the level of identification and verification required by the certificate level (i.e., low, moderate, high).

2.8 Transport Layer Security

Transport Layer Security (TLS) is a protocol that provides for authentication, confidentiality, and data integrity between two communicating applications. Secure Sockets Layer (SSL), a precursor to TLS, is not approved for use in the protection of Federal information. TLS versions 1.1 and 1.2 when properly configured are approved for the protection of Federal information. NIST SP 800-52 Rev 1, "Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations," provides requirements for configuration and use of TLS.

TLS establishes an encrypted connection to an authenticated peer and uses a handshake protocol to negotiate session parameters. The TLS handshake protocol initializes both the client and server to use optional cryptographic capabilities by negotiating a cipher suite of algorithms and functions, including key establishment, digital signature, confidentiality, and integrity algorithms. The handshake protocol is also used for configuration of security services, such as:

- Confidentiality: Provides assurance that data is kept secret.
- Message Integrity: Provides detection of unauthorized data modification, to ensure that undetected deletion, addition, or modification of data did not take place.
- Authentication: Provides assurance of the sender or receiver's identity.
- Replay Protection: Provides assurance that an unauthorized user does not capture and successfully replay previous data.

TLS is an application independent protocol, which provides security to any two communicating applications that transmit data over a network using an application protocol, such as the Hypertext Transfer Protocol (HTTP) and the Internet Message Access Protocol (IMAP). Hypertext Transfer Protocol Secure (HTTPS) is a combination of HTTP and TLS. The unencrypted HTTP protocol does not protect data from interception or alteration, which can subject users to eavesdropping, tracking, and the modification of data. Unencrypted HTTP connections create a privacy vulnerability and expose sensitive information about users of unencrypted federal websites and services. Data sent over HTTP is susceptible to interception, manipulation, and impersonation. This data can include browser identity, website content, search terms, and other user-submitted information.

The HTTPS/TLS security model uses “certificates” to guarantee authenticity. These certificates are cryptographically “signed” by a trusted CA. A part of the signature process is computing a “hash” of the data included in the certificate, which is accomplished by using a standard hashing algorithm.

When a TLS connection is established between a client and server, the handshake protocol is responsible for establishing the session parameters. The client and server negotiate algorithms for authentication, confidentiality, and integrity, as well as derive secret keys and establish other session parameters. The server selects a cipher suite from the list sent by the client to start the session. Cipher suites have the following format:

TLS_KeyExchangeAlgorithm_WITH_EncryptionAlgorithm_MessageAuthenticationAlgorithm

The following is an example of a cipher suite supported by TLS servers for interoperability purposes:

TLS_RSA_WITH_AES_128_CBC_SHA

The following explains each component in the cipher suite example above:

- TLS – Specifies whether SSL or TLS is used for the cipher suite.
- RSA – Algorithm used for the key exchange when this cipher suit is used.
- AES_128_CBC – Defines that AES encryption algorithm is used with a 128-bit AES key in CBC mode.
- SHA – Defines that SHA-1 is used as the message authentication algorithm.

NIST SP 800-52 Rev 1, Section 3.3.1, Cipher Suites, provides a list of approved cipher suites for use in federal system applications.

2.9 Cryptographic Controls for Data in Transit and at Rest

Encryption can provide both confidentiality and integrity of data in transit and at rest. Cryptography can provide data integrity either by using a hash of the data or by encrypting the data. Confidentiality is provided when the data is encrypted in such a way that only authorized users have access to the key to decrypt the data. All encryption is required to meet the minimum cryptographic key size requirements specified in Section 4.1, Cryptographic Key Requirements.

2.9.1 Data in Transit

Data in transit refers to data transmitted over an internal or external network, where it could be intercepted by an unauthorized person with access to the network. The internal or external networks could be either wired or wireless, or both, where an adversary could access the data by intercepting the data traffic by means of wire-tapping, or using sniffing tools. When clear text protocols (e.g. HTTP or File Transfer Protocol [FTP]) are used for data transmission, the data traffic is "in clear text" and can be easily intercepted by someone using tools to access user emails, copy personal credentials, or copy sensitive files. Hence, to safeguard against unauthorized interception, data in transit is encrypted (e.g., using HTTPS).

The computational effort required to encrypt using symmetric keys is far less than that used to encrypt using asymmetric keys. As a result, asymmetric keys are usually used to securely exchange symmetric keys between the communicating parties, and the symmetric keys are used to encrypt the following data transmissions.

The TLS protocol is commonly used to encrypt data in transit. TLS uses certificates to exchange public keys, and then the public keys are used to securely exchange private keys, making it very difficult for an adversary to intercept. Most encryption protocols include a hashing algorithm to ensure that data was not altered while in transit. This can also safeguard against "man-in-the-middle" attacks, because by decrypting and re-encrypting data, the attackers will alter the signature even if they do not change the data. When data in transit is encrypted, it can only be compromised if the session key is compromised.

2.9.2 Data at Rest

Data at rest refers to data stored on media. Encryption of data at rest can protect sensitive data against unauthorized access. Refer to CSO-STD-2004, "Electronic Media and Device Handling Standard," for further details in defining electronic media and storage devices.

There are two types of encryption:

- Hardware-based encryption: Typically provided by a special internal or external (e.g., Universal Serial Bus [USB]) hard drive with built-in hardware encryption, which is most efficient and provides the least adverse performance impact, and
- Software-based encryption: Provided on computer systems where performance is less of a concern; may be at the file system, file/folder, or other object (e.g., database) level.

If any adversary has physical access to a server or workstation, then file system permissions may not be effective in preventing unauthorized access to data. However, if data is encrypted at rest, the adversaries cannot access the data unless they obtain the decryption key. To quickly encrypt and decrypt data with minimal impact to system performance, most implementations of encryption of data at rest use symmetric-key algorithms. For example, the AES encryption algorithm can be used to encrypt data at rest.

Hashing algorithms protect the integrity of data at rest, by means of calculating the hash value and comparing it later to quickly and easily detect any changes made to the data. AES-encrypted portable media is an example of encryption of data at rest.

3 GENERAL REQUIREMENTS

This section provides general requirements that all system administrators and ISSOs authorized to administer and configure the cryptographic systems must comply with as the minimum set of controls.

ISSOs must ensure that all national security systems meet Committee on National Security Systems (CNSS) and NSA specified requirements for cryptographic controls. Systems that process Sensitive Compartmented Information (SCI) must adhere to Director of National Intelligence (DNI) policy, standards, and guidance. The DNI website is located at http://www.dni.gov/.

All externally-facing NRC websites and services must only be accessible through a securely configured HTTPS connection in accordance with Office of Management and Budget (OMB) Memorandum (M) M-15-13, "Policy to Require Secure Connections across Federal Websites and Web Services."

All cryptography must be implemented using a FIPS-validated cryptographic module operated in FIPS mode.

NRC shall employ a KMI to manage cryptographic modules and keys.

3.1 Cryptographic Algorithms and Modules

The following requirements apply to all unclassified systems for the use of cryptographic algorithms and modules:

- NRC systems must use FIPS-validated cryptographic algorithms with FIPS validated modules.
- Only cryptographic modules that meet the following requirements may be used:
  - Maintain a current NIST CMVP validation certificate,
  - Meet all CMVP configuration and policy requirements, and
  - Configured to support minimum cryptographic strength requirements for the specific application, as specified in Section 4.1, Cryptographic Key Requirements.

Cryptographic algorithms and modules used to protect classified information must comply with the requirements specified by CNSS and NSA for such information.

3.1.1 NSA Algorithm Suites

NSA has developed/approved various levels of cryptographic algorithms for secure federal applications that are categorized based upon the level of security provided by the algorithm. National Security Systems (NSSs) are required to use algorithms approved by NSA for the specific level of classification of the system. Detailed information regarding NSA approved algorithms and required key strengths can be found on the NSA website.

NSA is increasing reliance on commercial cryptographic technologies for securing NSS and is moving toward more transparency. Cryptographic algorithms are specified by the NIST and are used by NSA in solutions approved for protecting NSS. They include cryptographic algorithms for encryption, key exchange, digital signature, and hashing.

3.1.2 NSA Commercial Solutions for Classified Program

The Commercial Solutions for Classified (CSfC) Program was developed by NSA to enable commercial products to be used in layered solutions protecting classified NSS data. NSA has developed and published solution-level specifications called Capability Packages (CPs) and works with industry leaders, governments, and academia to develop product-level requirements in U.S. Government Protection Profiles (PPs). CPs for mobile access, virtual private networks (VPNs), wireless local area networks (WLANs), and data at rest solutions are published on the NSA website.

3.2 Cryptographic Key Management

The following requirements apply to all unclassified systems for cryptographic key management:

- Cryptographic key management must comply with NIST SP 800-57 (all parts).
- Public key certificates must be issued and validated by an NRC-approved CA with an NRC Authorizing Official (AO) authorization to operate (ATO).
- Equipment used to generate, store, and archive cryptographic keys must be physically protected in accordance with NRC MD 12.5, “NRC Cybersecurity Program,” Section V, “Physical and Environmental Security.”

Cryptographic key management systems used to protect classified information must comply with the requirements specified by CNSS and NSA for such information.

3.3 Encryption for Data in Transit and at Rest

Classified systems must comply with the requirements specified by CNSS and NSA for encryption of data in transit and data at rest.

OMB Circular A-130 Appendix III, “Managing Information as a Strategic Resource,” provides the following overreaching requirement for encryption of unclassified information in transit and at rest:

> *Encrypt all FIPS 199 moderate-impact and high-impact information at rest and in transit, unless encrypting such information: is technically infeasible or would demonstrably affect the ability of agencies to carry out their respective missions, functions, or operations; and the risk of not encrypting is accepted by the authorizing official and approved by the agency CIO.*

Cryptographic algorithms and modules used to encrypt data in transit and data at rest to meet the OMB Circular A-130 requirement must be configured in accordance with Section 4.1 of this standard.

3.4 Transport Layer Security

The following requirements apply to all unclassified NRC information systems employing TLS:

- All implementations and uses of the TLS protocol must meet the minimum requirements specified in NIST SP 800-52 Rev 1.
- Existing NRC systems must use TLS v1.1 at a minimum and should make migrating to TLS v1.2 a priority. New systems must use TLS 1.2.
- Only FIPS-approved cipher suites are permitted for NRC TLS applications. FIPS-approved cipher suites are listed in NIST SP 800-52 Rev 1.
- NRC systems must be configured to use the strongest TLS cryptographic cipher suites first when negotiating a connection between the server and a client. If the client does not support stronger ciphers, then less strong FIPS-approved cipher suites can be used.

In addition to FIPS requirements, cryptographic algorithms and cipher strengths used for TLS implementations must meet the minimum requirements specified in Section 4.1, Cryptographic Key Requirements.

4 SPECIFIC REQUIREMENTS

This section provides specific requirements for NRC unclassified (not Controlled Unclassified Information [CUI]), CUI Basic and CUI Specified (non-SGI), and CUI Specified (SGI) systems that utilize cryptographic algorithms, protocols, and cryptographic modules.

All classified information systems must comply with CNSS and NSA specific requirements for cryptographic controls.

4.1 Cryptographic Key Requirements

Key strength requirements are determined based on the sensitivity of the information paired with the length of time the information will remain sensitive. Table 4.1-1, Minimum Cryptographic Strength Requirements, identifies the minimum cryptographic key strength requirements for NRC systems based upon information sensitivity and Table 4.1-2, Security Strength Time Frames, specifies key strength time frames.

All cryptographic key sizes and hash functions must comply with the minimum cryptographic strength requirements for the respective NRC systems information type as specified in Table 4.1-1.

The following defines the information contained within the columns of Table 4.1-1:

- Information Category: Identifies the sensitivity level of information to encrypt for processing within NRC networks (i.e., unclassified [FIPS 199 low-impact non-CUI], CUI Basic and CUI Specified [non-SGI], or CUI Specified [SGI]).
- Confidentiality and Integrity (High Watermark): Identifies the high watermark impact level of confidentiality or integrity associated with the information type being encrypted (i.e., low, moderate, or high). If the impact level differs between confidentiality and integrity, the higher of the two (high watermark) applies.

- Minimum HMAC Hash: Identifies the minimum HMAC function to use to secure the messaging between two entities processing the select information type.

  For example: To exchange CUI Basic and CUI Specified (non-SGI) – Moderate information between a client and server, it has to use the HMAC-SHA1 hash function to attain the required minimum level of security. However, a higher level of HMAC hash can be used for added security, if so desired.

- Minimum Hash (Non-HMAC): Identifies hash functions to use for each information type to secure messaging between two entities.

  For example: To exchange CUI Basic and CUI Specified (non-SGI) – High information between two entities, any one of the listed hash functions can be used (i.e., SHA-256, SHA-512/256, or SHA3-256) to secure the messages.

- Minimum Symmetric-Key: Identifies the minimum key size for each information type when symmetric-key algorithm is used for encryption.
- Minimum Asymmetric-Key: Identifies the key type and the associated length/size to use for encrypting each information type.

  For example: An RSA-3072 key coupled with hash function SHA-384 provides computation of a digital signature that can be used to process a CUI Specified (SGI) information type.

Table 4.1-1: Minimum Cryptographic Strength Requirements

Information Category	Confidentiality and Integrity (High Watermark)	Minimum HMAC Hash	Minimum Hash (Non-HMAC)	Minimum Symmetric-Key	Minimum Asymmetric-Key
Unclassified (non-CUI)					
Unclassified (non-CUI)	Low	HMAC-SHA1	SHA-224 SHA-512/224 SHA3-224	128-bit AES	DSA – 2048 bits RSA – 2048 bits DH – 2048 bits EC – 224 bits
CUI Basic and CUI Specified (non-SGI)					
CUI Basic & CUI Specified (non-SGI)	Moderate	HMAC-SHA1	SHA-224 SHA-512/224 SHA3-224	128-bit AES	DSA – 2048 bits RSA – 2048 bits DH – 2048 bits EC – 256 bits
CUI Specified (non-SGI)	High	HMAC-SHA1	SHA-256 SHA-512/256 SHA3-256	128-bit AES	DSA – 2048 bits RSA – 2048 bits DH – 2048 bits EC – 256 bits
CUI Specified (SGI)					
CUI Specified (SGI)	High	HMAC-SHA-256	SHA-384 SHA-512/256 SHA3-256	256-bit AES	DSA – 3072 bits RSA – 3072 bits DH – 3072 bits EC – 384 bits

Supplemental Information: Increased key size or hash functions, other than the minimum specified in Table 4.1-1, can be selected where more stringent data security is required. However, it must be kept in mind that when a key establishment scheme is used with one or more algorithms (e.g., HMAC or AES), the security strength is determined by the weakest algorithm and key size used. For example, if a 224-bit ECC key is used to establish a 128-bit AES key, only 112 bits of security can be provided by the AES key, as a 224-bit ECC key can only provide a maximum of 112 bits of security. If increased level of security is required, then the ECC key size can be increased in size (e.g., 256 bits for 128 bits of security).

Similarly, when a hash function and digital signature algorithms are used to compute a digital signature, the weaker of the two will determine the security strength. For example, when SHA-256 and a 2048-bit RSA key is used to compute a digital signature, only 112 bits of security is achieved due to the fact that the 2048-bit RSA key cannot provide more than 112 bits of security strength. In order to increase the security strength in this case, use a 3072-bit RSA key.

However, increased key size and hash functions inversely affect the system performance. Therefore, it is necessary that the appropriate key size and hash functions are selected to achieve required security strength for the information type being protected.

The following defines the information contained within the columns of Table 4.1-2:

1. Column 1 is divided into two sub-columns. The first sub-column indicates the security strength to be provided; the second sub-column indicates whether cryptographic protection is being applied to data (i.e., encrypted), or whether cryptographically protected data is being processed (i.e., decrypted).
2. Columns 2 and 3 indicate the time frames during which the security strength is either acceptable, OK for legacy use or disallowed.
   - "Acceptable" indicates that the algorithm or key length is not known to be insecure.
   - "Legacy-use" means that an algorithm or key length may be used because of its use in legacy applications (i.e., the algorithm or key length can be used to process cryptographically protected data).
   - "Disallowed" means that an algorithm or key length shall not be used for applying cryptographic protection.

Table 4.1-2: Security Strength Time Frames

<table>
<tr><th colspan="2">Security Strength</th><th>Through 2030</th><th>2031 and Beyond</th></tr>
<tr><td rowspan="2">< 112</td><td>Applying</td><td colspan="2">Disallowed</td></tr>
<tr><td>Processing</td><td colspan="2">Legacy-use</td></tr>
<tr><td rowspan="2">112</td><td>Applying</td><td rowspan="2">Acceptable</td><td>Disallowed</td></tr>
<tr><td>Processing</td><td>Legacy use</td></tr>
<tr><td>128</td><td rowspan="3">Applying/Processing</td><td>Acceptable</td><td>Acceptable</td></tr>
<tr><td>192</td><td>Acceptable</td><td>Acceptable</td></tr>
<tr><td>256</td><td>Acceptable</td><td>Acceptable</td></tr>
</table>

4.2 Public Keys

NRC users must have separate public certificates and key pairs for authentication and for encryption. The private key for authentication should never be known to anyone other than the owner of that key pair. This ensures that no other person can impersonate the authentication key pair owner. The private key for encryption should be placed in a key escrow to enable decryption of information should the key pair owner not be available. Escrowed encryption private keys should be under a two-person rule (requires the presence of two authorized individuals at the same time to access the key) with established procedures to ensure the keys are made available only for NRC authorized purposes.

While creating digital certificates and key pairs for protection of data:

- The digital signature certificate and key pairs must only be used for identification and authentication.
- The encryption certificate and key pairs must only be used for purposes of encryption.
- The certificates and associated key pairs used to access or encrypt CUI Specified (SGI) applications must be kept separate and distinct from certificates and associated key pairs used in CUI Basic and CUI Specified (non-SGI) applications or applications/systems using non-public/non-sensitive and public information.
- All encryption certificates must be issued by an NRC CA that has received an ATO by the NRC AO.
- Self-signed certificates must not be used in any NRC system.

4.3 Cryptographic Key Management

The following requirements apply to cryptographic key management:

- An AO authorized Key Management solution must be used to generate cryptographic keys (e.g., for digital signatures) within NRC systems.
- A cryptoperiod (key lifetime designation) must:
  - Be assigned to a key upon key issuance.
  - Be determined based on the categorization of data being encrypted, key strength, and risk factors.
  - Meet the guidelines provided in NIST SP 800-57 (all parts).
  - Be used to either replace or destroy the keys, as applicable.

APPENDIX A. HOW DO I DECIDE IF I NEED CRYPTOGRAPHY?

If cryptography is used within the system, the cryptographic requirements must be met. The following table identifies situations where cryptography is required.

Situation	Required?	Cryptography Purpose
Information with a confidentiality sensitivity of moderate or above	Yes	Confidentiality of data at rest and data in transit
Information with an integrity sensitivity of moderate or above	Yes	Integrity of data at rest and data in transit
Cryptographic key management	Yes	Confidentiality and integrity protection of cryptographic keys
Remote system/data access	Yes	Protection of remote user, system, and device access to a system. This includes use of digital certificates to ensure user access can validate that the connection is with a valid NRC computer (e.g., NRC web sites).
Wireless access to system/data	Yes	Protection of authentication information and information transmission
Authenticator management	Yes	Protection of individual and device authentication information
Electronic signatures	Yes	Creation of all types of electronic signatures and verification of signature authenticity
Secure name and address resolution	Yes	Secure DNS

APPENDIX B. ACRONYMS

AES	Advanced Encryption Standard
AO	Authorizing Official
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CFR	Code of Federal Regulations
CKMS	Cryptographic Key Management System
CMVP	Cryptographic Module Validation Program
CNSS	Committee on National Security Systems
CNSSI	Committee for National Security Systems Instruction
CP	Capability Package
CS	Cybersecurity
CSfC	Commercial Solutions for Classified
CST	Cryptographic and Security Testing
CTR	Counter
CUI	Controlled Unclassified Information
DH	Diffie-Hellman key exchange
DNI	Director of National Intelligence
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EA	Executive Agent
ECB	Electronic Codebook
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography

ECDSA	Elliptic Curve Digital Signature Algorithm
FTP	File Transfer Protocol
FIPS	Federal Information Processing Standard
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IA	Information Assurance
IFC	Integer Factorization Cryptography
IMAP	Internet Message Access Protocol
ISSO	Information System Security Officer
KMI	Key Management Infrastructure
M	Memorandum
MAC	Message Authentication Code
MD	Management Directive
NARA	National Archives and Records Administration
NIST	National Institute of Standards and Technology
NRC	Nuclear Regulatory Commission
NSA	National Security Agency
NSS	National Security Systems
OCIO	Office of the Chief Information Officer
OFB	Output Feedback
OMB	Office of Management & Budget
PKI	Public Key Infrastructure
PP	Protection Profile
PUB	Publication
RSA	Public-key algorithm developed by Rivest, Shamir, and Adleman

SCI	Sensitive Compartmented Information
SGI	Safeguards Information
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	Special Publication
SSL	Secure Sockets Layer
STD	Standard
SUNSI	Sensitive Unclassified Non-Safeguards Information
TCP	Transmission Control Protocol
TLS	Transport Layer Security
USB	Universal Serial Bus
VPN	Virtual Private Network
WLAN	Wireless Local Area Network

APPENDIX C. GLOSSARY

Approved	FIPS approved. An algorithm or technique that is either 1) specified in a FIPS PUB, or 2) adopted in a FIPS PUB and specified in either the FIPS PUB, or in a document referenced by the FIPS PUB.
Bits of Security	A number associated with the amount of work (that is, the number of operations) that is required to break a cryptographic algorithm or system. The security strength is specified in bits and is a specific value from the set {80, 112, 128, 192, and 256}. Note that a security strength of 80 bits is no longer considered sufficiently secure.
Certificate	A set of data that uniquely identifies a key pair and an owner that is authorized to use the key pair. The certificate contains the owner's public key and possibly other information, and is digitally signed by a CA (i.e., a trusted party), thereby binding the public key to the owner.
Certification Authority	The entity in a PKI that is responsible for issuing certificates and exacting compliance with a PKI policy.
Ciphertext	Data in its encrypted form.
Cryptanalysis	1. Operations performed in defeating cryptographic protection without an initial knowledge of the key employed in providing the protection. 2. The study of mathematical techniques for attempting to defeat cryptographic techniques and information system security. This includes the process of looking for errors or weaknesses in the implementation of an algorithm or in the algorithm itself.
Cryptographic Algorithm	A well-defined computational procedure that takes variable inputs, including a cryptographic key, and produces an output.
Cryptographic Boundary	An explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all hardware, software, and/or firmware components of a cryptographic module.
Cryptographic Key	A parameter used in conjunction with a cryptographic algorithm that determines the specific operation of that algorithm.
Cryptographic Module	The set of hardware, software, and/or firmware that implements at least one approved security function (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.
Cryptographic System	Associated information security (INFOSEC) items interacting to provide a single means of encryption or decryption.
Cryptoperiod	The time span during which a specific key is authorized for use or remains in effect for a given system or application.

CUI	Federal Register 32 Code of Federal Regulations (CFR) Part 2002, "Controlled Unclassified Information; Final Rule," established policy for a federal government-wide Controlled Unclassified Information (CUI) program and a CUI Registry, maintained by the CUI Executive Agent (EA), to provide authorized categories, subcategories, associated markings, as well as applicable safeguarding, dissemination, and decontrol procedures for CUI. The CUI program provides a standardized and simplified way to manage unclassified information that requires protections and dissemination controls, pursuant to and consistent with applicable laws, regulations, and government-wide policies. This excludes all information classified under EO 13526, "Classified National Security Information," dated December 29, 2009, and the Atomic Energy Act of 1954, as amended. All federal government-wide unclassified information that requires any protection or dissemination control is declared CUI and mandates that authorized holders protect CUI using CUI Basic or CUI Specified controls.
CUI Basic	CUI Basic is the subset of CUI for which the authorizing law, regulation, or Government-wide policy does not set out specific handling or dissemination controls. Agencies handle CUI Basic per the uniform set of controls in the 32 CFR Part 2002 and the CUI Registry. All CUI Basic categories are controlled at 'moderate' confidentiality level at a minimum. However, some CUI may have higher, or different level of requirements if a law, regulation, or government-wide policy requires or permits other controls for protecting or disseminating that information. The final rule mandates that authorized holders of CUI use at least the CUI Basic default set of standards to protect information.
CUI Specified	CUI Specified is the subset of CUI in which the authorizing law, regulation, or government-wide policy contains specific handling controls that it requires or permits agencies to use that are in addition to those for CUI Basic. The CUI Registry indicates which laws, regulations, and government-wide policies include such specific requirements. CUI Specified information may be handled at higher confidentiality levels if the authorities establishing and governing the CUI Specified allow or require a more specific or stringent controls. Safeguards Information (SGI) is considered to be CUI specified and requires more stringent controls.
Decryption	The process of transforming ciphertext into plaintext using a cryptographic algorithm and key.
Digital Signature	The result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity and signatory non-repudiation.
Encryption	The cryptographic transformation of data to produce ciphertext.

Entity	A person, organization, device, or process.
Ephemeral Key	A cryptographic key that is generated for each execution of a key establishment process and meets other requirements of the key type.
Handshake Protocol	An automated process that sets parameters for communication between two devices before normal communication takes place between the devices.
Hash Function	A mathematical function that maps a string of arbitrary length (up to a predetermined maximum size) to a fixed length string.
Hash Value	The fixed-length bit string produced by a hash function.
Hash-based message authentication code (HMAC)	A message authentication code that utilizes a keyed hash.
Key	A parameter used in conjunction with a cryptographic algorithm that determines its operation.
Key Agreement	A key-establishment procedure where resultant keying material is a function of information contributed by two or more participants, so that no party can predetermine the value of the keying material independently of the other party's contribution.
Key Escrow	A deposit of the private key of a subscriber and other pertinent information pursuant to an escrow agreement or similar contract binding upon the subscriber, the terms of which require one or more agents to hold the subscriber's private key for the benefit of the subscriber, an employer, or other party, upon provisions set forth in the agreement.
Key Management	The activities involving the handling of cryptographic keys and other related security parameters (e.g. passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.
Key Management Infrastructure (KMI)	The framework and services that provide the generation, production, storage, protection, distribution, control, tracking, and destruction for all cryptographic keying material, symmetric keys as well as public keys and public key certificates.
Key Pair	A public key and its corresponding private key.
Key-Derivation Function	A function that generates a binary string, called keying material, with the input of a cryptographic key or shared secret and possibly other data.

Keyed-Hash Message Authentication Code	A message authentication code that uses a cryptographic key in conjunction with a hash function.
Keying Material	The data necessary to establish and maintain cryptographic keying relationships.
Key-Wrapping	A method of encrypting keys (along with associated integrity information) that provides both confidentiality and integrity protection using a symmetric key.
KMI Central Oversight Authority	The Key Management Infrastructure (KMI) entity that provides overall KMI data synchronization and system security oversight for an organization or set of organizations.
KMI Client Nodes	Client nodes are interfaces for managers, devices, and applications to access KMI functions, including the requesting of certificates and other keying material. They may include cryptographic modules, software, and procedures necessary to provide user access to the KMI.
KMI Key Processing Facility(ies)	The Key Processing Facility is a KMI component that performs one or more of the following functions: • Acquisition or generation of public key certificates, • Initial generation and distribution of keying material, • Maintenance of a database that maps user entities to an organization's certificate/key structure, • Maintenance and distribution of nodal key compromise lists and/or certificate revocation lists, and • Generation of audit requests and the processing audit responses as necessary for the prevention of undetected compromises.
KMI Service agents	Entities that support organizations' KMIs as single points of access for other KMI nodes.
Message	The data that is signed. Also known as "signed data" during the signature verification and validation process.
Message Authentication Code	A cryptographic checksum that results from passing data through a message authentication algorithm. In this standard, the message authentication algorithm is called HMAC, while the result of applying HMAC is called the MAC.
Message Digest	The result of applying a hash function to a message. Also known as "hash value."
Non-Repudiation	A service that is used to provide assurance of the integrity and origin of data in such a way that the integrity and origin can be verified and

	validated by a third party as having originated from a specific entity in possession of the private key (i.e., the signatory).
Private Key	A cryptographic key that is used with an asymmetric cryptographic algorithm. For digital signatures, the private key is uniquely associated with the owner and is not made public. The private key is used to compute a digital signature that may be verified using the corresponding public key.
Public Key	A cryptographic key that is used with an asymmetric cryptographic algorithm and is associated with a private key. The public key is associated with an owner and may be made public. In the case of digital signatures, the public key is used to verify a digital signature that was signed using the corresponding private key.
Public Key Infrastructure	A framework that is established to issue, maintain, and revoke public key certificates.
Secret Key	A cryptographic key that is uniquely associated with one or more entities. The use of the term "secret" in this context does not imply a classification level; rather the term implies the need to protect the key from disclosure or substitution.
Security Strength	A number associated with the amount of work (that is, the number of operations) that is required to break a cryptographic algorithm or system. Sometimes referred to as a security level or bits of security.
Self-Signed Certificate	A public-key certificate whose digital signature may be verified by the public key contained within the certificate. The signature on a self-signed certificate protects the integrity of the data, but does not guarantee the authenticity of the information. The trust of self-signed certificates is based on the secure procedures used to distribute them.
Signatory	The entity that generates a digital signature on data using a private key.
Signed Data	The data or message upon which a digital signature has been computed.
Static Key	A key that is intended for use for a long period of time and is for use in many instances of cryptographic key-establishment schemes.
Symmetric-Key	A cryptographic algorithm that uses the same secret key for an operation; such as encryption and decryption.

APPENDIX D. REFERENCES

CNSS Advisory Memorandum IA 02-15	Use of Public Standards for the Secure Sharing of Information Among National Security Systems
CNSSI 4009	National Information Assurance (IA) Glossary
FIPS PUB 140-2	Security Requirements for Cryptographic Modules
FIPS PUB 180-4	Secure Hash Standard (SHS)
FIPS PUB 186-4	Digital Signature Standard (DSS)
FIPS PUB 198-1	The Keyed-Hash Message Authentication Code (HMAC)
FIPS PUB 202	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions
NIST Policy on Hash Functions	http://csrc.nist.gov/groups/ST/hash/policy.html
NIST SP 800-38A	Recommendation for Block Cipher Modes of Operation
NIST SP 800-52	Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementation
NIST SP 800-56A	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography
NIST SP 800-56B	Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography
NIST SP 800-57	Recommendation for Key Management Part 1: General Part 2: Best Practices for Key Management Organization Part 3: Application-Specific Key Management Guidance
NIST SP 800-73-4	Interfaces for Personal Identity Verification
NIST SP 800-76-2	Biometric Specifications for Personal Identity Verification
NIST SP 800-78-4	Cryptographic Algorithms and Key Sizes for Personal Identity Verification
NIST SP 800-131A	Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths
NIST SP 800-152	A Profile for U.S. Federal Cryptographic Key Management Systems (CKMS)

NIST SP 800-157	Guidelines for Derived Personal Identity Verification Credentials
OMB Circular A-130	Managing Information as a Strategic Resource
OMB M-15-13	Policy to Require Secure Connections across Federal Websites and Web Services

OCIO-CS-STD-2009 Change History

Date	Version	Description of Changes	Method Used to Announce & Distribute	Training
04-Feb-10	1.0	Initial issuance	Distribution at ISSO forum and posting on ISD web page	Upon request
05-Sep-17	2.0	Revised based on the latest versions of NIST PUBs and FIPS PUBs addressing cryptographic controls based upon the current threat environment.	Distribution at ISSO forum and posting on OCIO web page	Upon request