NIST Special Publication 800-131A

Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths

Elaine Barker and Allen Roginsky

Computer Security Division
Information Technology Laboratory

C O M P U T E R S E C U R I T Y

January 2011

U.S. Department of Commerce
Gary Locke, Secretary
National Institute of Standards and Technology
Patrick Gallagher, Director

Abstract

At the start of the 21st century, the National Institute of Standards and Technology (NIST) began the task of providing cryptographic key management guidance, which includes defining and implementing appropriate key management procedures, using algorithms that adequately protect sensitive information, and planning ahead for possible changes in the use of cryptography because of algorithm breaks or the availability of more powerful computing techniques. NIST Special Publication (SP) 800-57, Part 1 was the first document produced in this effort, and includes a general approach for transitioning from one algorithm or key length to another. This Recommendation (SP 800-131A) provides more specific guidance for transitions to the use of stronger cryptographic keys and more robust algorithms.

Key Words: cryptographic algorithm, digital signatures, encryption, hash function, key agreement, key derivation, key management, key transport, key wrapping, message authentication codes, random number generation, security strength, transition.

Authority

This publication has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems.

This Recommendation has been prepared for use by federal agencies. It may be used by non-governmental organizations on a voluntary basis and is not subject to copyright. (Attribution would be appreciated by NIST.)

Nothing in this document should be taken to contradict 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.

Conformance testing for implementations of this Recommendation will be conducted within the framework of the Cryptographic Module Validation Program (CMVP) and the Cryptographic Algorithm Validation Program). The requirements of this Recommendation are indicated by the word "shall." Some of these requirements may be out-of-scope for CMVP or CAVP validation testing, and thus are the responsibility of entities using, implementing, installing or configuring applications that incorporate this Recommendation.

Table of Contents

Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths

1 Introduction

1.1 Background and Purpose

At the beginning of the 21st century, the National Institute of Standards and Technology (NIST) began the task of providing cryptographic key management guidance. This included lessons learned over many years of dealing with key management issues, and is intended to encourage the definition and implementation of appropriate key management procedures, to use algorithms that adequately protect sensitive information, and to plan ahead for possible changes in the use of cryptography because of algorithm breaks or the availability of more powerful computing techniques. General key management guidance, including the general approach for transitioning from one algorithm or key length to another, is addressed in Part 1 of Special Publication (SP) 800-57 [SP 800-57].

This Recommendation (SP 800-131A) is intended to provide more detail about the transitions associated with the use of cryptography by Federal government agencies for the protection of sensitive, but unclassified information. The Recommendation addresses the use of algorithms and key lengths; the validation of cryptographic modules that utilize them is provided in [SP 800-131B].

The dates provided in SP 800-131A may differ from the dates originally provided in the 2005 version of [SP 800-57]. The revised dates provided herein attempt to deal with the realities associated with an orderly transition and are based on a better understanding of the risks associated with extending the dates in those cases where this was done. Note that an upper-date limit is not provided herein for many of the algorithms and key lengths discussed; that information is provided in [SP 800-57], and should be considered valid unless different guidance is provided in the future.

1.2 Useful Terms for Understanding this Recommendation

1.2.1 Security Strengths

Some of the guidance provided in [SP 800-57] includes the definition of security strengths, the association of the **approved** algorithms and key lengths with these security strengths, and a projection of the time frames during which the algorithms and key lengths could be expected to provide adequate security. Note that the length of the cryptographic keys is an integral part of these determinations.

The security strength of an algorithm with a particular key length[1] is measured in bits and is, basically, a measure of the difficulty of discovering the key. The understood security strength for each algorithm is provided in [SP 800-57].

The appropriate security strength to be used depends on the sensitivity of the data being protected, and needs to be determined by the owner of that data (e.g., a person or an organization). For the Federal government, a minimum security strength of 80 bits is recommended in 2010; a minimum security strength of 112 bits is strongly recommended, beginning in 2011 (see [SP 800-57]). However, with the acceptance of a certain amount of risk, the minimum of 80 bits of security strength may be used until the end of 2013. Based on the latest understanding of the state of the art for breaking the cryptographic algorithms, given particular key lengths, the transition to the 112-bit security strength **shall** be accomplished by 2014, except where specifically indicated. See Appendix A for an explanation.

Specific key lengths are provided in [FIPS 186-3] for DSA, ECDSA and RSA digital signatures, in [SP 800-56A] for Diffie Hellman and MQV key agreement, and in [SP 800-56B] for RSA key agreement and key transport. These key lengths are strongly recommended for interoperability, and their security strengths are provided in [SP 800-57]. However, other key lengths are commonly used. The security strengths associated with these key lengths may be determined using the formula provided in the [FIPS 140-2] Implementation Guideline [IG 7.5]. In this Recommendation (SP 800-131A), security strengths of 80 bits and 112 bits are specifically addressed. The reference to 80 bits of security strength should be interpreted as a security strength of at least 80 bits, but less than 112 bits (i.e., 80 ≤ security strength < 112).

1.2.2 Definition of Terms

The terms "**approved**", "**acceptable**", "**deprecated**", "**restricted**" and "**legacy-use**" are used throughout this Recommendation.

- **Approved** is used to mean that an algorithm is specified in a FIPS or NIST Recommendation (published as a NIST Special Publication).
- **Acceptable** is used to mean that the algorithm and key length is safe to use; no security risk is currently known.
- **Deprecated** means that the use of the algorithm and key length is allowed, but the user must accept some risk. The term is used when discussing the key lengths or algorithms that may be used to apply cryptographic protection to data (e.g., encrypting or generating a digital signature).
- **Restricted** means that the use of the algorithm or key length is deprecated, and there are additional restrictions required to use the algorithm or key length for applying cryptographic protection to data (e.g., encrypting).
- **Legacy-use** means that the algorithm or key length may be used to process already protected information (e.g., to decrypt ciphertext data or to verify a digital

[1] The term "key size" is commonly used in other documents.

signature), but there may be risk in doing so. Methods for mitigating this risk should be considered (see Appendix B).

The use of algorithms and key lengths for which the terms deprecated, restricted and legacy-use are listed require that the user must accept some risk that increases over time. If a user determines that the risk is unacceptable, then the algorithm or key length is considered disallowed, from the perspective of that user. It is the responsibility of the user or the user's organization to determine the level of risk that can be tolerated for an application and its associated data and to define any methods for mitigating those risks.

Other cryptographic terms used in this Recommendation are defined in the documents listed in Appendix C.

2 Encryption

Encryption is a cryptographic operation that is used to provide confidentiality for sensitive information; decryption is the inverse operation. Several algorithms are currently **approved** for encryption by the Federal government:

- Triple DES is specified in [SP 800-67], and has two key lengths, known as two-key Triple DES and three-key Triple DES. Two-key Triple DES has been assessed at a security strength of 80 bits[2], whereas three-key Triple DES is assessed at a security strength of 112 bits.
- SKIPJACK was **approved** in [FIPS 185], and is assessed at a security strength of 80 bits.
- AES is specified in [FIPS 197]. It has three **approved** key lengths: 128, 192 and 256 bits. AES-128 is assessed at a security strength of 128 bits, AES 192 at a security strength of 192 bits, and AES-256 at a security strength of 256 bits.

The transition schedule for encryption algorithms is provided in Table 1.

Table 1: Encryption Transitions

Algorithm	Use
Two-key Triple DES Encryption	Acceptable through 2010 Restricted use from 2011 through 2015 Disallowed after 2015
Two-key Triple DES Decryption	Acceptable through 2010 Legacy-use after 2010
Three-key Triple DES Encryption and Decryption	Acceptable
SKIPJACK Encryption	Acceptable through 2010

[2] The assessment of at least 80-bits of security for two-key Triple DES is based on the assumption that an attacker has approximately 2^{40} matched plaintext and ciphertext blocks.

SKIPJACK Decryption	Acceptable through 2010 Legacy-use after 2010
AES-128 Encryption and Decryption	Acceptable
AES-192 Encryption and Decryption	Acceptable
AES-256 Encryption and Decryption	Acceptable

Two-key Triple DES encryption:

The use of two-key Triple DES is **acceptable** for encryption through December 31, 2010.

From January 1, 2011 through December 31, 2015, the use of two-key Triple DES for encryption is **restricted**: the total number of blocks of data encrypted with the same cryptographic key **shall not** be greater than 2^{20} (note that for this algorithm, a block is the 64-bit block of a Triple DES encryption operation). This restriction also applies to those keys that were first used prior to 2011 and continue to be used beyond December 31, 2010 (i.e., those keys whose cryptoperiod begins prior to 2011 and extends into 2011). Rationale for this exception is provided in Appendix A.1.

After December 31, 2015, two-key Triple DES **shall not** be used for encryption.

Two-key Triple DES decryption:

Decryption using two-key Triple DES is **acceptable** through 2010.

Decryption using two-key Triple DES is allowed for **legacy-use** after 2010

SKIPJACK encryption and decryption:

The use of SKIPJACK for encryption is **acceptable** through December 31, 2010. SKIPJACK **shall not** be used for encryption thereafter.

The use of SKIPJACK for decryption is **acceptable** through 2010.

The use of SKIPJACK for decryption is allowed for **legacy-use** after 2010

AES and three-key Triple DES encryption and decryption:

The use of AES-128, AES-192, AES-256 and three-key Triple DES is **acceptable**.

3 Digital Signatures

Digital signatures are used to provide assurance of origin authentication and data integrity. These assurances are sometimes extended to provide assurance that a party in a dispute (the signatory) cannot repudiate (i.e., refute) the validity of the signed document; this is commonly known as non-repudiation. The digital signature algorithms **approved** in [FIPS 186-2] and [FIPS 186-3] are DSA, ECDSA and RSA.

The generation of a digital signature on data requires the use of 1) a cryptographic hash function that operates on the data to be signed, and 2) the use of a cryptographic key and a signing algorithm to generate a signature on the output of the hash function (and, by extension, the data that is intended to be signed). This section addresses the use of the

cryptographic keys used with the signing algorithm; discussions of the hash function to be used during the generation of digital signatures are provided in Section 9. The details of the security strengths of the algorithms and the key lengths used can be found in [SP 800-57].

Note that the security strength of digital signatures is determined by the security strength of both the cryptographic key with the signing algorithm, and the cryptographic hash function used.

Table 2 provides the schedule for transitioning from digital signatures providing at least 80 bits of security strength to those providing at least 112 bits of security strength.

Table 2: Digital Signatures Security Strength Transitions

Digital Signature Process	**Use**	
Digital Signature Generation	80 bits of security strength: DSA: (($\|p\| \geq 1024$) and ($\|q\| \geq 160$)) and (($\|p\| < 2048$) OR ($\|q\| < 224$)) RSA: $1024 \leq \|n\| < 2048$ EC: $160 \leq \|n\| < 224$	Acceptable through 2010 Deprecated from 2011 through 2013 Disallowed after 2013
	≥ 112 bits of security strength: DSA: $\|p\| \geq 2048$ and $\|q\| \geq 224$ RSA: $\|n\| \geq 2048$ EC: $\|n\| \geq 224$	Acceptable
Digital Signature Verification	80 bits of security strength: DSA: (($\|p\| \geq 1024$) and ($\|q\| \geq 160$)) and (($\|p\| < 2048$) OR ($\|q\| < 224$)) RSA: $1024 \leq \|n\| < 2048$ EC: $160 \leq \|n\| < 224$	Acceptable through 2010 Legacy-use after 2010

	≥ 112 bits of security strength: DSA: $\|p\| \geq 2048$ and $\|q\| \geq 224$ RSA: $\|n\| \geq 2048$ EC: $\|n\| \geq 224$	Acceptable

Digital signature generation:

The use of key lengths providing 80 bits of security strength is **acceptable** for digital signature generation through December 31, 2010.

From January 1, 2011 through December 31, 2013, the use of key lengths providing 80 bits of security strength is **deprecated**. The user must accept risk when using these keys, particularly when approaching the December 31, 2013 upper-limit date. This is especially critical for digital signatures on data whose signature is required to be valid beyond this date. Appendix A.2 provides rationale for this modified guidance. See Section 5.6.2 of [SP 800-57] for further guidance.

After December 31, 2013, key lengths providing less than 112 bits of security strength **shall not** be used to generate digital signatures.

Key lengths providing at least 112 bits of security are **acceptable**.

Digital signature verification:

Key lengths providing 80 bits of security using **approved** digital signature algorithms are **acceptable** through 2010.

Key lengths providing 80 bits of security using **approved** digital signature algorithms are allowed for **legacy-use** after 2010.

Key lengths providing at least 112 bits of security using **approved** digital signature algorithms are **acceptable**.

4 Random Number Generation

Random numbers are used for various purposes, such as the generation of keys, nonces and authentication challenges. Several random number generators (RNGs) have been **approved** for use by the Federal government. Until relatively recently, [FIPS 186-2] was the approval vehicle for RNGs, specifying RNGs and approving the RNGs in American National Standard (ANS) X9.31-1998 [X9.31] and ANS X9.62-1998 [X9.62].

In 2007, a new set of RNGs were **approved** in SP 800-90 [SP 800-90] that provide higher levels of security than the previously-approved RNGs. In addition, [SP 800-90] contains more comprehensive guidance on RNG use. Note that in [SP 800-90], the term "random bit generator" (RBG) is used instead of "random number generator" (RNG); any

difference is not important for this Recommendation (i.e., SP 800-131A), but both terms will be used below.

Note that in 2005, a revision of [X9.62] was approved that includes the HMAC_DRBG specified in [SP 800-90], and does not include the RNGs in the 1998 version.

The transition schedule for RBGs and RNGs is provided in Table 3.

RBGs that are compliant with SP 800-90 are **acceptable**.

The use of the RNGs specified in FIPS 186-2, [X9.31] and [X9.62] is **acceptable** through December 31, 2010.

The use of the RNGs specified in FIPS 186-2, [X9.31] and ANS [X9.62] is **deprecated** from 2011 through December 31, 2015, and disallowed after 2015.

Table 3: Random Number Generation Transitions

Description	Use
RBGs specified in SP 800-90 (HASH, HMAC, CTR, DUAL_EC) and ANS X9.62-2005 (HMAC)	Acceptable
RNGs specified in FIPS 186-2, ANS X9.31-1998 and ANS X9.62-1998	Acceptable through 2010 Deprecated from 2011 through 2015 Disallowed after 2015

5 Key Agreement Using Diffie-Hellman and MQV

Key agreement is a technique that is used to establish symmetric keys between two entities that intend to communicate, whereby both parties contribute information to the key agreement process. Two families of key agreement schemes are defined and have been **approved** in [SP 800-56A]: Diffie-Hellman (DH) and MQV. Each has been defined over two different mathematical structures: finite fields and elliptic curves. Key agreement includes two steps: the use of an appropriate DH or MQV "primitive" to generate a shared secret, and the use of a key derivation function (KDF) to generate one or more keys from the shared secret. [SP 800-56A] contains **approved** DH and MQV primitives and **approved** KDFs for key agreement.

Several protocol standards specify one or more of the DH or MQV primitives specified in [SP 800-56A], but use different KDFs; the specifically-approved key agreement schemes to be used by these protocols are provided in [SP 800-135].

Other key agreement schemes that are not specified in either SP 800-56A or SP 800-135 are allowed by the FIPS 140-2 Implementation Guideline [IG D.2]; these will be discussed below as the non-compliant schemes.

Table 4 contains the transition schedule for DH and MQV key agreement schemes.

Table 4: SP 800-56A Key Agreement (DH and MQV)

Scheme	Use[a]

Scheme	Use[a]	
SP 800-56A and SP 800-135 DH and MQV schemes using finite fields	$\|p\|$ = 1024 bits, and $\|q\|$ = 160 bits	Acceptable through 2010 Deprecated from 2011 through 2013 Disallowed after 2013
	$\|p\|$ = 2048 bits, and $\|q\|$ = 224 or 256 bits	Acceptable
SP 800-56A and SP 800-135 DH and MQV schemes using elliptic curves	$160 \leq \|n\| \leq 223$ bits and $\|h\| \leq 10$	Acceptable through 2010 Deprecated from 2011 through 2013 Disallowed after 2013
	$\|n\| \geq 224$ bits and h as specified in Table 5	Acceptable
Non-compliant DH and MQV schemes using finite fields	$\|p\| \geq 1024$ bits, and $\|q\| \geq 160$ bits	Acceptable through 2010 Deprecated from 2011 through 2013
	$\|p\| \geq 2048$ bits, and $\|q\| \geq 224$ bits	Deprecated after 2013. All other values of p and q are disallowed after 2013
Non-compliant DH and MQV schemes using elliptic curves	$\|n\| \geq 160$	Acceptable through 2010 Deprecated from 2011 through 2013
	$\|n\| \geq 224$	Deprecated after 2013. All other values of n are disallowed after 2013

a $|p|$, $|q|$, $|n|$ and $|h|$ are used to denote the bit length of p, q, n and h, respectively.

SP 800-56A and SP 800-135 DH and MQV schemes using finite fields:

Through December 31, 2010, the use of $|p|$ = 1024 bits, and $|q|$ = 160 bits is **acceptable**, where p is the field order (i.e., the modulus), and q is the subgroup order. Note that the length of p is also the length of the public key, and the length of q is the length of the private key**.** In [SP 800-56A], these parameters are denoted as parameter set FA.

From January 1, 2011 through December 31, 2013, the use of $|p|$ = 1024 bits, and $|q|$ = 160 bits is **deprecated**. The rationale for extending the transition is similar to that for digital signatures, which is discussed in Appendix A.2.

After December 31, 2013, $|p| = 1024$ bits, and $|q| = 160$ bits **shall not** be used in a key agreement scheme.

The use of $|p| = 2048$ bits, and $|q| = 224$ or 256 bits is **acceptable**. In [SP 800-56A], $|p| = 2048$ bits, and $|q| = 224$ are denoted as parameter set FB; and $|p| = 2048$ bits, and $|q| = 256$ are denoted as parameter set FC.

SP 800-56A and SP 800-135 DH and MQV schemes using elliptic curves:

Through December 31, 2010, the use of $160 \leq |n| \leq 223$ bits and $|h| \leq 10$ is **acceptable**, where n is the number of elements in the subgroup, and h is the cofactor of n for the order of the elliptic curve.

From January 1, 2011 through December 31, 2013, the use of $160 \leq |n| \leq 223$ bits and $|h| \leq 10$ is **deprecated**. The rationale for extending the transition is similar to that for digital signatures, which is discussed in Appendix A.2.

After December 31, 2013, $|n| \leq 223$ bits **shall not** be used in a key agreement scheme.

The use of $|n| \geq 224$ bits and h as specified in Table 5 is **acceptable**. In [SP 800-56A], four parameter sets are defined: EB – EE. The acceptable values for n and h are provided in [SP 800-56A] and in the following table.

Table 5: EC Parameter sets

	EB	**EC**	**ED**	**EE**
Length of n	224-255	256-383	384-511	512+
Maximum bit length of cofactor h	14	16	24	32

Non-compliant DH and MQV schemes using finite fields:

Through December 31, 2010, the use of $|p| \geq 1024$ bits, and $|q| \geq 160$ bits is **acceptable**.

From January 1, 2011 through December 31, 2013, the use of $|p| \geq 1024$ bits, and $|q| \geq 160$ bits is **deprecated**. The rationale for extending the transition is similar to that for digital signatures, which is discussed in Appendix A.2.

After December 31, 2013, the use of $|p| \geq 2048$ and $|q| \geq 224$ is **deprecated**. Values of p or q that do not meet this condition **shall not** be used.

Non-compliant DH and MQV schemes using elliptic curves:

Through December 31, 2010, the use of $|n| \geq 160$ bits is **acceptable**.

From January 1, 2011 through December 31, 2013, the use of $|n| \geq 160$ bits is **deprecated**. The rationale for extending the transition is similar to that for digital signatures, which is discussed in Appendix A.2.

After December 31, 2013, the use of $|n| \geq 224$ is **deprecated**. Values of $|n| < 224$ **shall not** be used.

6 Key Agreement and Key Transport Using RSA

[SP 800-56B] specifies the use of RSA for both key agreement and key transport. Key agreement is a technique in which both parties contribute information to the key agreement process. Key transport is a key establishment technique in which only one party determines the key. Some protocols that include key transport schemes are provided in [IG D.2]; these will be discussed below as the non-56B-compliant schemes. Note that in [IG D.2], key transport is often referred to as key wrapping. Note also that while there are implementations of RSA-based Key Transport schemes that are not compliant with SP 800-56B, there are no RSA-based Key Agreement schemes that are not compliant with SP 800-56B.

Guidance on **approved** key lengths for RSA is provided in [SP 800-56B]. Table 6 provides the transition schedule.

In the case of key transport keys (i.e., the keys used to encrypt other keys for transport), this Recommendation (SP 800-131A) applies to both the encryption and decryption of the transported keys.

Table 6: RSA-based Key Agreement and Key Transport Key Length Transitions

Scheme	Use	
SP 800-56B Key Agreement schemes	$\|n\|$ = 1024 bits	Acceptable through 2010 Deprecated from 2011 through 2013 Disallowed after 2013
	$\|n\|$ = 2048 bits	Acceptable
SP 800-56B Key Transport schemes	$\|n\|$ = 1024 bits	Acceptable through 2010 Deprecated from 2011 through 2013 Disallowed after 2013
	$\|n\|$ = 2048 bits	Acceptable
Non-56B-compliant Key Transport schemes	$\|n\| \geq$ 1024 bits	Acceptable through 2010 Deprecated from 2011 through 2013
	$\|n\| \geq$ 2048 bits	Deprecated after 2013. All other values of $n <$ 2048 bits are disallowed after 2013

SP 800-56B RSA Key Agreement schemes:

The use of RSA key agreement schemes with $|n|$ = 1024 is **acceptable** through December 31, 2010.

From January 1, 2011 through December 31, 2013, the use of RSA key agreement schemes with $|n| = 1024$ is **deprecated**. The rationale for extending the transition is similar to that for digital signatures, which is discussed in Appendix A.2.

After December 31, 2013, RSA key agreement schemes with $|n| = 1024$ **shall not** be used.

The use of RSA key agreement schemes with $|n| = 2048$ is **acceptable**.

SP 800-56B RSA Key Transport schemes:

The use of RSA key transport schemes with $|n| = 1024$ is **acceptable** through December 31, 2010.

From January 1, 2011 through December 31, 2013, the use of RSA key transport schemes with $|n| = 1024$ is **deprecated**. The rationale for extending the transition is similar to that for digital signatures, which is discussed in Appendix A.2.

After December 31, 2013, RSA key transport schemes with $|n| = 1024$ **shall not** be used.

The use of RSA key transport schemes with $|n| = 2048$ is **acceptable**.

Non-56B-compliant RSA Key Transport schemes:

The use of RSA key transport schemes with $|n| \geq 1024$ is **acceptable** through December 31, 2010.

From January 1, 2011 through December 31, 2013, the use of RSA key transport schemes with $|n| \geq 1024$ is **deprecated**. The rationale for extending the transition is similar to that for digital signatures, which is discussed in Appendix A.2.

After December 31, 2013, the use of $|n| \geq 2048$ is **deprecated**. Values of $|n| < 2048$ **shall not** be used.

7 Key Wrapping

Key wrapping is the encryption of a symmetric key by another symmetric key with integrity protection. Symmetric keys are used with algorithms such as Triple-DES and AES. See [SP 800-57] for further information. As of 2010, neither a FIPS nor a NIST Recommendation specify key wrapping algorithms, although an informal specification for key wrapping using AES is available at http://csrc.nist.gov/groups/ST/toolkit/documents/kms/AES_key_wrap.pdf. However, [IG D.2] addresses key wrapping as defined in the AES key wrapping specification.

Table 7 provides the transition schedule.

Table 7: Symmetric Key Wrapping Key Length Transitions

Algorithm	Use
Two-key Triple DES Key Wrap	Acceptable through 2010 Restricted use from 2011 through 2015 Disallowed after 2015

Two-key Triple DES Key Unwrap	Acceptable through 2010 Legacy-use after 2010
AES and Three-key Triple DES Key Wrap and Unwrap	Acceptable

Two-key Triple DES key wrapping:

Two-key Triple DES is **acceptable** for wrapping and unwrapping keying material through December 31, 2010.

From January 1, 2011 through December 31, 2015, the use of two-key Triple DES for wrapping keying material is **restricted**: the total number of blocks of keying material wrapped with the same cryptographic key **shall** be no more than 2^{20}.

After December 31, 2015, two-key Triple DES **shall not** be used to wrap keying material.

Two-key Triple DES key unwrapping:

The use of two-key Triple DES is **acceptable** through December 31, 2010.

The use of two-key Triple DES is allowed for **legacy-use** after December 31, 2010.

AES and three-key Triple DES key wrapping and unwrapping:

AES and three-key Triple DES are **acceptable** for wrapping and unwrapping keying material.

8 Deriving Additional Keys from a Cryptographic Key

[SP 800-108] specifies key derivation functions that use a cryptographic key (called a key derivation key) to generate additional keys. The key derivation key could be:

- Generated using an **approved** RNG (see [IG 7.8] and [SP 800-133]),
- Obtained using a key agreement or key transport scheme (see Sections 5 and 6 of this Recommendation, i.e., SP 800-131A), or
- Obtained using a key wrapping algorithm (see Section 7).

Table 8 provides the transition of key lengths for key derivation.

Table 8: Key Length Transitions for a Key Derivation Function (KDF)

Algorithm	Use	
HMAC-based KDF	Acceptable	
CMAC-based KDF	Two-key TDES-based KDF	Acceptable through 2010 Deprecated from 2011 through 2015 Disallowed after 2015
	AES- and Three-key Triple	Acceptable

	DES-based KDFs	

HMAC-based KDF:

The use of HMAC-based KDFs is **acceptable** using an **approved** hash function, including SHA-1. See Section 10 for discussions of the key lengths used with HMAC.

CMAC-based KDF:

The use of two-key TDES as the block cipher algorithm in a CMAC-based KDF is **acceptable** through December 31, 2010.

The use of two-key TDES as the block cipher algorithm in a CMAC-based KDF is **deprecated** from 2011 through December 31, 2015.

Two-key Triple DES **shall not** be used to derive keying material after December 31, 2015.

The use of AES and three-key Triple DES as the block cipher algorithm in a CMAC-based KDF is **acceptable**.

9 Hash Functions

Five **approved** hash functions are specified in [FIPS 180-3]. The security strengths for hash functions are dependent on their use, and are provided in [SP 800-57]. Additional discussions about the different uses of hash functions are provided below and in [SP 800-107].

Note that, while there have been attacks reported on SHA-1, this Recommendation (i.e., SP 800-131A) will consider its strength to be 80 bits for digital signature generation for the purpose of discussion.

The transition schedule for hash functions is provided in Table 9.

Table 9: Hash Function Transitions

Hash Function	Use	
SHA-1	Digital signature generation	Acceptable through 2010 Deprecated from 2011 through 2013 Disallowed after 2013
	Digital signature verification	Acceptable through 2010 Legacy-use after 2010
	Non-digital signature generation applications	Acceptable
SHA-224		

SHA-256	Acceptable for all hash function applications
SHA-384	
SHA-512	

SHA-1 for digital signature generation:

SHA-1 is **acceptable** for digital signature generation through December 31, 2010.

From January 1, 2011 through December 31, 2013, the use of SHA-1 is **deprecated** for digital signature generation. The user must accept risk when SHA-1 is used, particularly when approaching the December 31, 2013 upper limit. This is especially critical for digital signatures on data for which the signature is required to be valid beyond this date. See Section 5.6.2 of [SP 800-57] for further guidance.

SHA-1 **shall not** be used for digital signature generation after December 31, 2013.

SHA-1 for digital signature verification:

For digital signature verification, the use of SHA-1 is **acceptable** through December 31, 2010.

For digital signature verification, SHA-1 is allowed for **legacy-use** after December 31, 2010.

SHA-1 for non-digital signature applications:

For all other hash function applications, the use of SHA-1 is **acceptable**. The other applications include HMAC, Key Derivation Functions (KDFs), Random Number Generation (RNGs and RBGs), and hash-only applications (e.g., hashing passwords and using SHA-1 to compute a checksum, such as the approved integrity technique specified in Section 4.6.1 of [FIPS 140-2]).

SHA-224, SHA-256, SHA-384, SHA-512:

The use of these hash functions is **acceptable** for all hash function applications.

10 Message Authentication Codes (MACs)

Two types of message authentication code mechanisms using symmetric keys have been **approved** for use: those based on hash functions, and those based on block-cipher algorithms. [FIPS 198-1] specifies a keyed-hash message authentication code (HMAC) that uses a hash function; [SP 800-107] provides additional guidance on the uses of HMAC. Block cipher modes for generating MACs are specified in [SP 800-38B], [SP 800-38C] and [SP 800-38D]. The CMAC mode specified in [SP 800-38B] uses either AES or Triple DES; the CCM and GCM/GMAC modes specified in [SP 800-38C] and [SP 800-38D], respectively, use AES.

Table 10: Message Authentication Code Transitions

MAC Algorithm	Use	
HMAC Generation	Key lengths ≥ 80 bits and < 112 bits	Acceptable through 2010 Deprecated from 2011 through

		2013 Disallowed after 2013
	Key lengths ≥ 112 bits	Acceptable
HMAC Verification	Key lengths ≥ 80 bits and < 112 bits	Acceptable through 2010 Legacy-use after 2010
	Key lengths ≥ 112 bits	Acceptable
CMAC Generation	Two-key Triple DES	Acceptable through 2010 Deprecated from 2011 through 2015 Disallowed after 2015
	AES and Three-key Triple DES	Acceptable
CMAC Verification	Two-key Triple DES	Acceptable through 2010 Legacy-use after 2010
	AES and Three-key Triple DES	Acceptable
CCM and GCM/GMAC Generation	AES	Acceptable
CCM and GCM/GMAC Verification	AES	Acceptable

HMAC Generation:

Any **approved** hash function may be used.

The use of key lengths ≥ 80 bits, but < 112 bits is **acceptable** through December 31, 2010.

From January 1, 2011 through December 31, 2013, the use of key lengths ≥ 80 bits, but < 112 bits is **deprecated**.

After December 31, 2013, key lengths < 112 bits **shall not** be used.

The use of key lengths ≥ 112 bits is **acceptable**.

HMAC Verification:

The use of key lengths ≥ 80 bits, but < 112 bits is **acceptable** through December 31, 2010.

The use of key lengths ≥ 80 bits, but < 112 bits is allowed for **legacy-use** after December 31, 2010.

The use of key lengths ≥ 112 bits is **acceptable.**

CMAC Generation:

The use of two-key Triple DES is **acceptable** through December 31, 2010.

From January 1, 2011 through December 31, 2015, the use of two-key Triple DES is **deprecated**.

After December 31, 2015, two-key Triple DES **shall not** be used.

The use of AES and three-key Triple DES is **acceptable**.

CMAC Verification:

The use of two-key Triple DES is **acceptable** through December 31, 2010.

The use of two-key Triple DES is allowed for **legacy-use** after December 31, 2010.

The use of AES and three-key Triple DES is **acceptable**.

CCM and GCM/GMAC Generation and Verification:

The use of CCM and GCM/GMAC is **acceptable**.

Appendix A: Decision Rationale

A.1 Security Strength of the Two-Key Triple DES Encryption Algorithm

The security strength of two-key Triple DES is discussed in Section 5.6.1 of [SP 800-57]. The estimate given there is 80 bits of security, assuming that the attacker has access to approximately 2^{40} (plaintext, ciphertext) pairs, where the plaintext is encrypted with the same secret key that the attacker is trying to discover. The more general formula given in [OorWie91] says that the security strength of two-key Triple DES against the best known attack is 120-*n* bits, where *n* is such that the attacker has access to 2^n (plaintext, ciphertext) pairs.

For the transition period from January 1, 2011 through December 31, 2015, the effective 100-bit security strength provided when no more than 2^{20} blocks are encrypted with the same key is allowed.

A.2 Rationale for Extending the Deadline for a Transition to Digital Signature Keys Providing 112 Bits of Security Strength

With the publication of SP 800-57, Part 1 in 2005, NIST announced the intent to transition from a minimum cryptographic security strength of 80 bits to a security strength of 112 bits by the end of 2010.
The schedule was based on academic research available at that time indicating that the digital signature algorithms using the 1024-bit RSA or DSA keys, would either be broken or be in serious danger of being broken by this date. In addition, the expected increase in the speed of computers and the anticipated improvement of the factoring techniques for RSA justified this point of view.

However, in the five years since the proposal of the 2010 transition date, factoring techniques have not progressed as quickly as anticipated. The first successful factorization of a 768-bit RSA modulus was not reported until December 2009 (see [Factoring]). This effort required six months of computations using highly sophisticated equipment. According to the paper's authors, factoring a 1024-bit modulus would be "one thousand times harder" than a 768-bit one. This means that another 6-7 years are likely to pass before 1024-bit numbers could realistically be factored.

In view of this research, and because some widely-used protocols still require the use of the 1024-bit RSA keys, NIST decided to extend the transition date for keys of this length for three years longer than originally anticipated, that is, through the end of 2013. Similarly, the 1024-bit DSA keys and the 160-223 bit ECDSA keys can be used during this time period. However, since such keys are more and more likely to be broken as the 2013 date approaches, the data owner must understand and accept the risk of continuing to use these keys to generate digital signatures. Hence, the use of such keys during the 2011-2013 transition period for the generation of digital signatures is marked as

deprecated. As stated in Section 3, "After December 31, 2013, key lengths providing less than 112 bits of security strength **shall not** be used to generate digital signatures."

Signature verification will continue to be permissible with any key that provides at least 80 bits of security.

Appendix B: Mitigating Risk When Using Algorithms and Keys for Legacy-Use

Certain algorithms and key sizes are allowed for legacy-use when removing or verifying the cryptographic protection already applied to sensitive information (e.g., decrypting ciphertext or verifying a digital signature or message authentication code). However, a user must accept that the protection of the information may no longer be as good as desired.

B.1 Decryption and Key Unwrapping Using Symmetric Key Algorithms (e.g., Two-key Triple DES)

Sensitive information may continue to need confidentiality protection beyond the date when the algorithm and key length used to protect that information are no longer considered adequate.

Symmetric key algorithms use the same key for encryption to produce ciphertext data as must be used to decrypt the ciphertext data back to the original plaintext data. However, since the algorithm and key length used to encrypt the information are no longer considered secure, those entities using the algorithm to decrypt the ciphertext data should consider that an adversary may be capable of determining the key that was used for encryption. If the adversary has access to the ciphertext data and can determine the key, then the data no longer has reliable confidentiality protection. That is, the owner of the sensitive information should consider the information to no longer be protected (i.e., the information should be considered as being in plaintext form).

Several scenarios need to be considered when evaluating whether or not the information is or will remain secure.

1. If the ciphertext information was made available to an adversary (e.g., the ciphertext was transmitted over the Internet), the ciphertext may have been recorded by the adversary. In such a case, there is a possibility that the adversary can determine the key for decrypting the ciphertext, thus exposing the sensitive information. The remaining items assume that this situation is not the case or that the probability is sufficiently low that other measures to further protect the information are warranted.
2. If the ciphertext data is protected from exposure to potential attack (e.g., the ciphertext data is saved in secure storage), then the confidentiality of the information as encrypted using the now-insecure algorithm or key length may remain valid.
3. If the ciphertext data is re-encrypted or rewrapped[3] using a stronger algorithm or key length, then the confidentiality of the sensitive information will remain valid as long as the stronger algorithm remains secure.

[3] Decrypted or unwrapped using the original algorithm and key to produce the original plaintext, and then encrypting or wrapping the plaintext using another algorithm and key.

4. If the ciphertext data needs to be made publicly available (e.g., transmitted) during the period in which the algorithm and key length are only allowed for legacy-use, then the information must be re-encrypted or super-encrypted[4] using a more secure algorithm and key length.

B.2 Digital Signature Generation Using Asymmetric (Public) Keys and SHA-1

The purpose of a digital signature is to bind information to an entity. A party uses their secret (private) key to electronically sign a document. Everyone else can use this party's public key, which is not secret, to check that only the owner of the secret key could sign the document.

Suppose, however, that the digital values of the signature were the same when two completely different messages were signed with the same private key. While the signing entity signed one message only (this could be a purchase contract or any other financial or legal document), the attacker could claim that a different message was signed. That other message would probably be more to the attacker's liking and, presumably, the signer would have never signed it. As the attacker does not know the signing party's private key, the only way for the attacker to force this collision of signatures is to generate a hash function collision, since in the **approved** signature schemes, the signer signs not the message itself, but the hash value of the message.

The SHA-1 hash function has at most 80 bits of security against collision attacks. Therefore, it is quite likely that due to the advancement in computing technology and the discovery of new attacks against the hash functions, the attackers will be able to generate, in the foreseeable future, the scenario described in the previous paragraph. Hence, it is important to find the mitigating factors that would not allow this to happen. NIST recommends that during the 2011-2013 transition period, the users of the cryptographic modules that perform digital signature generation carefully assess their risk and decide if the risk of allowing SHA-1 in digital signature generation can be accepted. Some applications, such as signing a public key certificate, are very high risk and the use of SHA-1 in those applications should be avoided as much as possible. In NIST's view, after 2013, the risk is unacceptable in all applications, and the use of SHA-1 when generating a digital signature is not allowed after that date.

The verification of digital signatures generated prior to the end of 2013 may need to be performed at a later date. Appendix B.3 addresses ways to mitigate the risk of validating a signature that was generated using SHA-1.

B.3 Digital Signature Verification Using Asymmetric (Public) Keys and SHA-1

Digital signatures are generated on information (e.g., a message) using a hash function, an asymmetric private key and a signing algorithm. Such signatures are verified using the information that was signed (i.e., the message), the same hash function that was used

[4] The ciphertext is encrypted or wrapped using an additional algorithm and key.

during the generation of the digital signature, the public key associated with the private key, the digital signature itself, and a verification algorithm. The digital signature on the message may need to be verified and considered valid beyond the period when the signing algorithm, hash function and private key are considered secure.

Asymmetric digital signature algorithms use different, but associated, keys for generating and verifying the digital signatures. The only secret associated with these processes is the private key, which should only be known by the party that generates the signature; that is, the hash function, the public key, the generating and verification algorithms, and the message can be assumed to be known by an adversary. The adversary's goal is to either determine the private key using this known information, including other messages that were signed using the same private key, or to determine alternative information (i.e., alternate messages) that produce the same digital signature. If the private key or other messages can be so determined, the information that has already been signed is subject to possible attack (e.g., by substituting messages and signatures that are beneficial to the adversary). During the "legacy-use" period, the entity that is verifying the digital signature should accept that an adversary may have at least one of these capabilities.

In order for the signed information to continue to be verifiable as valid, both the signed information and the digital signature need to be protected against possible modification (e.g., placed in secure storage) or against modification without detection (e.g., time-stamped and signed with an additional signature).

B.4 Verification of Message Authentication Codes (MACs) Using CMAC

A message authentication code (MAC) may need to remain verifiable and valid beyond the date when the algorithm and key length used to generate the MAC are no longer considered adequate.

As in the case of symmetric algorithms used for encryption (see Appendix B.1), the same key is used to generate the MAC as must be used for verification of that MAC. Since the algorithm and key length used to generate that MAC are no longer considered secure, an entity that verifies a MAC using a no-longer-secure algorithm and key length should assume that an adversary may be capable of determining the key that was used for MAC generation. During the "legacy-use" period, the adversary may be assumed to be capable of determining the MAC key and generating MACs on new messages or substituting more beneficial messages (beneficial to the adversary) that produce the same MAC.

In order for the MACed data to continue to be verifiable as valid during the "legacy-use" period, both the MACed data and the MAC need to be protected against possible modification or substitution (e.g., placed in secure storage).

Appendix C: References

FIPS and SP documents are available at http://csrc.nist.gov/publications/, except for the FIPS 140-2 Implementation Guidance, which is available at http://csrc.nist.gov/groups/STM/cmvp/standards.html#02.

[FIPS 140-2] Security Requirements for Cryptographic Modules, with Change Notices, December 2002.

[FIPS 180-3] Secure Hash Standard (SHS), October 2008.

[FIPS 185] Escrowed Encryption Standard, Feb 1994.

[FIPS 186-2] Digital Signature Standard, January 2000.

[FIPS 186-3] Digital Signature Standard, June 2009.

[FIPS 197] Advanced Encryption Standard, November 2001.

[FIPS 198-1] Keyed-Hash Message Authentication Code (HMAC), July 2008.

[IG X.Y] Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program, where X.Y is the section number.

[SP 800-38B] CMAC Mode of Authentication, May 2005.

[SP 800-38C] Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, May 2004.

SP 800-38D] Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.

[SP 800-56A] Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, March 2007.

[SP 800-56B] Recommendation for Pair-Wise Key Establishment Using Integer Factorization, DRAFT, December 2008.

[SP 800-57] Part 1, Recommendation for Key Management: General, March 2007.

[SP 800-67] Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, May 2008.

[SP 800-90] Recommendation for Random Number Generation Using Deterministic Random Bit Generators, March 2007.

[SP 800-107] Recommendation for Applications Using Approved Hash Algorithms, February 2009.

[SP 800-108] Recommendation for Key Derivation Using Pseudorandom Functions, November 2008.

[SP 800-131B] Transitions: Validation of Transitioning Cryptographic Algorithms and Key Lengths, under development.

[SP 800-133] Recommendation for Cryptographic Key Generation, under development.

[SP 800-135] Recommendation for Existing Application-Specific Key Derivation Functions, Draft.

Non-NIST References:

[Factoring] T. Kleinjung et al, "Factorization of a 768-bit RSA Modulus), February 18, 2010.

[OorWie91] P. van Oorschot and M. Wiener, "A known plaintext attack on two-key triple encryption", *Advances in Cryptology*, EUROCRYPT '90 (LNCS 473), 318-325, 1991.

[X9.31] American National Standard (ANS) X9.31-1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), Withdrawn, but available from X9.org.

[X9.62] American National Standard X9.62-2005, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).