NIST Special Publication 800-131A
Revision 1

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

Elaine Barker
Allen Roginsky

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-131Ar1

COMPUTER SECURITY

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

NIST Special Publication 800-131A
Revision 1

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

Elaine Barker
Allen Roginsky
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-131Ar1

November 2015

U.S. Department of Commerce
Penny Pritzker, Secretary

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

Authority

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

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

National Institute of Standards and Technology Special Publication 800-131A Revision 1
Natl. Inst. Stand. Technol. Spec. Publ. 800-131A Rev. 1, 29 pages (November 2015)
CODEN: NSPUE2

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-131Ar1

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

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

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

Comments on this publication may be submitted to:

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

Reports on Computer Systems Technology

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

Abstract

The National Institute of Standards and Technology (NIST) provides cryptographic key management guidance for 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 includes a general approach for transitioning from one algorithm or key length to another. This Recommendation (SP 800-131A Revision 1) provides more specific guidance for transitions to the use of stronger cryptographic keys and more robust algorithms.

Keywords

cryptographic algorithm; digital signatures; encryption; hash function; key agreement; key derivation functions; key management; key transport; key wrapping; message authentication codes; random number generation; security strength; transition.

Acknowledgments

The authors would like to specifically acknowledge the assistance of the following NIST employees in developing this revision of SP 800-131A: Lily Chen, Sharon Keller, Kerry McKay and Apostol Vassilev.

Table of Contents

List of Tables

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 1) encourage the specification and implementation of appropriate key management procedures, 2) use algorithms that adequately protect sensitive information, and 3) 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

SP 800-131A was originally published in January 2011. This revision updates the transition guidance provided in the previous version; these changes are listed in Appendix C. The most significant difference is the deprecation of the non-approved key-agreement and key-transport schemes through December 31, 2017, and the intent to disallow them thereafter.

Although transition dates are provided in [SP 800-57], this document (i.e., SP 800-131A) is intended to provide more detailed information that deals with the realities associated with an orderly transition. 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.

In [SP 800-57], the security strength provided by an algorithm with a particular key length[1] is measured in bits and is a measure of the difficulty of subverting the cryptographic protection that is provided by the algorithm and key. A security strength for each algorithm is provided in [SP

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

800-57]. This is the estimated maximum security strength that an algorithm with a particular key length can provide, given that the key used with that algorithm has sufficient entropy[2].

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 112 bits is required for applying cryptographic protection (e.g., for encrypting or signing data). Note that prior to 2014, a security strength of 80 bits was **approved** for applying these protections, and the transitions in this document reflect this change to a strength of 112 bits. However, a large quantity of data was protected at the 80-bit security strength and may need to be processed (e.g., decrypted or have a digital signature verified). The processing of this already-protected data at the lower security strength is allowed, but a certain amount of risk must be accepted.

Specific key lengths are provided in [FIPS 186-4] 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] Implementation Guideline [IG 7.5].

1.2.2 Definition of Terms

The terms “**approved**”, “**acceptable**”, “**deprecated**”, “**restricted**”, “**legacy-use**” and **"disallowed"** 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 signature), but there may be a risk in doing so.
- **Disallowed** means that the algorithm or key length is no longer allowed for the indicated use.

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

[2] Entropy is a measure of the amount of disorder, randomness or variability in a closed system.

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 B.

1.2.3 Definition of Terms

The following symbol has been used in this Recommendation:

len(*x*) The length of an integer *x* in bits.

2 Encryption and Decryption Using Block Cipher Algorithms

Encryption is a cryptographic operation that is used to provide confidentiality for sensitive information, and decryption is the inverse operation. Several block cipher algorithms have been **approved** for use by the Federal government:

- TDEA (Triple Data Encryption Algorithm; often referred to as Triple DES) is specified in [SP 800-67], and has two variations, known as two-key TDEA and three-key TDEA. Three-key TDEA is the stronger of the two variations.
- SKIPJACK was **approved** in [FIPS 185]. However, approval for the use of SKIPJACK is being withdrawn, as its security strength is now considered inadequate.
- AES (Advanced Encryption Standard) is specified in [FIPS 197] and has three **approved** key lengths: 128, 192 and 256 bits.

See [SP 800-57] for more information about the security strengths provided by these algorithms.

Note that encryption and decryption using these algorithms require the use of modes of operation. Some of these modes also provide authentication when performing encryption, and provide verification when performing decryption on the encrypted and authenticated information (see [SP 800-38C] and [SP 800-38D]).

The approval status of the block cipher encryption/decryption algorithms is provided in Table 1.

Table 1: Approval Status of Symmetric Algorithms Used for Encryption and Decryption

Algorithm	Use
Two-key TDEA Encryption	Restricted through 2015 Disallowed after 2015
Two-key TDEA Decryption	Legacy-use
Three-key TDEA Encryption and Decryption	Acceptable
SKIPJACK Encryption	Disallowed
SKIPJACK Decryption	Legacy-use
AES-128 Encryption and Decryption	Acceptable
AES-192 Encryption and Decryption	Acceptable
AES-256 Encryption and Decryption	Acceptable

Two-key TDEA encryption:

> Through December 31, 2015, the use of two-key TDEA 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 TDEA encryption operation).
>
> After December 31, 2015, the use of two-key TDEA for encryption is **disallowed.**

Two-key TDEA decryption:

Decryption using two-key TDEA is allowed for **legacy-use**.

SKIPJACK encryption and decryption:

The use of SKIPJACK for encryption is **disallowed**.

The use of SKIPJACK for decryption is allowed for **legacy-use**.

AES and three-key TDEA encryption and decryption:

The use of AES-128, AES-192, AES-256 and three-key TDEA 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 186-4] 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 a digital signature algorithm is no greater than the minimum of 1) the security strength that can be supported by the cryptographic keys used to generate signatures, and 2) the security strength (with respect to collision resistance) of the cryptographic hash function that operates on the data to be signed.

Table 2 provides the approval status of the algorithms and key lengths for the generation and verification of digital signatures by the Federal government.

Table 2: Approval Status of Algorithms Used for Digital Signature Generation and Verification

Digital Signature Process	Use	
Digital Signature Generation	< 112 bits of security strength: DSA: $\mathbf{len}(p) < 2048$ OR $\mathbf{len}(q) < 224$ RSA: $\mathbf{len}(n) < 2048$ ECDSA: $\mathbf{len}(n) < 224$	Disallowed
	≥ 112 bits of security strength: DSA: $\mathbf{len}(p) \geq 2048$ AND $\mathbf{len}(q) \geq 224$ RSA: $\mathbf{len}(n) \geq 2048$ ECDSA: $\mathbf{len}(n) \geq 224$	Acceptable

Digital Signature Verification	< 112 bits of security strength: DSA[3]: $((512 \le \mathbf{len}(p) < 2048)$ OR $(160 \le \mathbf{len}(q) < 224))$ RSA: $1024 \le \mathbf{len}(n) < 2048$ ECDSA: $160 \le \mathbf{len}(n) < 224$	Legacy-use
	≥ 112 bits of security strength: DSA: $\mathbf{len}(p) \ge 2048$ AND $\mathbf{len}(q) \ge 224$ RSA: $\mathbf{len}(n) \ge 2048$ ECDSA: $\mathbf{len}(n) \ge 224$	Acceptable

Digital signature generation:

Key lengths providing less than 112 bits of security **shall not** be used to generate digital signatures.

Key lengths providing at least 112 bits of security are **acceptable** for the generation of digital signatures using **approved** algorithms.

Digital signature verification:

Key lengths providing less than 112 bits of security using **approved** digital signature algorithms for verifying digital signatures are allowed for **legacy-use**.

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

[3] The lower bounds for **len**(*p*) and **len**(*q*) are those that were specified in [FIPS 186-2].

4 Random Bit Generation

Random numbers are used for various purposes, such as the generation of keys, nonces and authentication challenges. Several deterministic random bit generator (DRBG) algorithms have been **approved** for use by the Federal government. SP 800-90A [SP 800-90A] includes three **approved** DRBG algorithms: HASH_DRBG, HMAC_DRBG and CTR_DRBG.

A previous version of [SP 800-90A] included a fourth algorithm, DUAL_EC_DRBG, whose use is now disallowed for Federal applications. Several other algorithms that were previously approved for random number generation are now deprecated and will be disallowed after 2015: the random number generators specified in [FIPS 186-2], in American National Standard (ANS) X9.31-1998 [X9.31] and in ANS X9.62-1998 [X9.62].

The current approval status for DRBGs is provided in Table 3.

Table 3: Approval Status of Algorithms Used for Random Bit Generation

Description	Use
HASH_DRBG, HMAC_DRBG and CTR_DRBG	Acceptable
DUAL_EC_DRBG	Disallowed
RNGs in FIPS 186-2, ANS X9.31 and ANS X9.62-1998	Deprecated through 2015 Disallowed after 2015

RBGs that are compliant with the 2015 revision of SP 800-90A are **acceptable** for generating random bits.

The use of the Dual_EC_DRBG is **disallowed**.

Through December 31, 2015, the use of the RNGs specified in FIPS 186-2, [X9.31] and the 1998 version of [X9.62] are **deprecated**. After 2015, these RNGs are **disallowed**.

5 Key Agreement Using Diffie-Hellman and MQV

Key agreement is a technique that is used to establish keying material 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 Menezes-Qu-Vanstone (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 method (KDM) to generate one or more keys from the shared secret. [SP 800-56A] contains **approved** DH and MQV primitives and **approved** KDMs for key agreement.

Other key agreement schemes that are not specified in SP 800-56A are allowed by the FIPS 140 Implementation Guideline [IG D.8]; these will be discussed below as the **deprecated** schemes. They are **disallowed** after 2017.

Table 4 contains the approval status for DH and MQV key agreement schemes.

Table 4: Approval Status for SP 800-56A Key Agreement (DH and MQV) Schemes

Scheme	Use	
SP 800-56A DH and MQV schemes using finite fields	< 112 bits of security strength: $\mathbf{len}(p) < 2048$ OR $\mathbf{len}(q) < 224$	Disallowed
	≥ 112 bits of security strength: $\mathbf{len}(p) \geq 2048$ AND $\mathbf{len}(q) \geq 224$	Acceptable
SP 800-56A DH and MQV schemes using elliptic curves	< 112 bits of security strength: $160 \leq \mathbf{len}(n) < 224$	Disallowed
	≥ 112 bits of security strength: $\mathbf{len}(n) \geq 224$ AND $\mathbf{len}(h)$ as specified in Table 5	Acceptable
Non-compliant DH and MQV schemes using finite fields	< 112 bits of security strength: $\mathbf{len}(p) < 2048$ OR $\mathbf{len}(q) < 224$	Disallowed
	≥ 112 bits of security strength: $\mathbf{len}(p) \geq 2048$ AND $\mathbf{len}(q) \geq 224$	Deprecated through 2017 Disallowed after 2017

Non-compliant DH and MQV schemes using elliptic curves	< 112 bits of security strength: **len**(n) < 224	Disallowed
	≥ 112 bits of security strength: **len**(n) ≥ 224	Deprecated through 2017 Disallowed after 2017

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

The use of the finite field schemes in SP 800-56A is **acceptable** if **len**(p) ≥ 2048 and **len**(q) ≥ 224. Otherwise, their use is **disallowed**.

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

In [SP 800-56A], five parameter sets are defined: EA – EE. Except for the EA parameter set, all of them define acceptable ECC parameter sizes. The acceptable values for **len**(n) and **len**(h) are provided 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:

The use of these schemes is **disallowed** if **len**(p) < 2048 or **len**(q) < 224.

Through December 31, 2017, the use of these schemes is **deprecated** if **len**(p) ≥ 2048 and **len**(q) ≥ 224. All of these schemes are **disallowed** after 2017.

Non-compliant DH and MQV schemes using elliptic curves:

The use of these schemes is **disallowed** if **len**(n) < 224.

Through December 31, 2017, the use of these schemes is **deprecated** if **len**(n) ≥ 224. All of these schemes are **disallowed** after 2017.

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 listed in [IG D.9]; these will be discussed below as the non-56B-compliant schemes. Note that in [IG D.9], key transport is often referred to as key wrapping. Note also that, while there are allowed implementations of RSA-based Key Transport schemes that are not compliant with [SP 800-56B], there are no approved or allowed 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 approval status.

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: Approval Status for the RSA-based Key Agreement and Key Transport Schemes

Scheme	Use	
SP 800-56B Key Agreement and Key Transport schemes	**len**(*n*) < 2048	Disallowed
	len(*n*) ≥ 2048	Acceptable
Non-56B-compliant Key Transport schemes	**len**(*n*) < 2048	Disallowed
	len(*n*) ≥ 2048	Deprecated through 2017 Disallowed after 2017

SP 800-56B RSA Key Agreement and Key Transport schemes:

The use of these schemes is **disallowed** if **len**(*n*) < 2048.

The use of these schemes is **acceptable** if **len**(*n*) ≥ 2048.

Non-56B-compliant RSA Key Transport schemes:

The use of these schemes is **disallowed** if **len**(*n*) < 2048.

Through December 31, 2017, the use of these schemes is **deprecated** if **len**(*n*)≥ 2048.

The use of these schemes is **disallowed** after December 31, 2017.

7 Key Wrapping

Key wrapping is the encryption of keying material by a symmetric key with integrity protection. [SP 800-38F] specifies three algorithms for key wrapping that use block ciphers: KW (AES Key Wrap) and KWP (AES Key Wrap with Padding), which use AES, and TKW (Triple DEA Key Wrap), which uses TDEA.

[SP 800-38F] also approves the CCM (Counter with Cipher Block Chaining-Message Authentication Code) and GCM (Galois Counter Mode) authenticated-encryption modes specified in [SP 800-38C] and [SP 800-38D] for key wrapping, as well as combinations of an approved encryption mode with an approved authentication method.

Table 7 provides the approval status of the block cipher algorithms used for key wrapping.

Table 7: Approval Status of Block Cipher Algorithms Used for Key Wrapping

Algorithm	Use
Key wrap using two-key TDEA	Restricted through 2015 Disallowed after 2015
Key unwrap using two-key TDEA	Legacy-use
Key wrap and unwrap using AES and three-key TDEA using any approved key-wrapping method	Acceptable
Block cipher key-wrapping methods not approved by [SP 800-38F]	Disallowed after 2017

Two-key TDEA:

> Through December 31, 2015, the use of two-key TDEA for key wrapping is **restricted**: the total number of blocks of data wrapped 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 TDEA encryption operation).
>
> Two-key TDEA **shall not** be used to wrap keying material after December 31, 2015.
>
> The use of two-key TDEA for unwrapping keying material using **approved** methods is allowed for **legacy-use**.

AES and three-key TDEA:

> AES and three-key TDEA are **acceptable** for both the wrapping and unwrapping of keying material using **approved** methods.

Symmetric-key wrapping methods not approved by [SP 800-38F]:

> Symmetric-key-wrapping methods that are not compliant with [SP 800-38F] are **disallowed** after December 31, 2017.

8 Deriving Additional Keys from a Cryptographic Key

[SP 800-108] specifies key derivation functions that use a pre-shared cryptographic key (called a key derivation key) to generate additional keys. Note that key derivation methods used within key-agreement schemes are not relevant to this section.

Table 8 provides the approval status of the key lengths used for key derivation.

Table 8: Approval Status of the Algorithms Used for a Key Derivation Function (KDF)

Algorithm	Use	
HMAC-based KDF	Acceptable	
CMAC-based KDF	Two-key TDEA-based KDF	Deprecated through 2015 Disallowed after 2015
	AES and Three-key TDEA	Acceptable

HMAC-based KDF (HMAC is the Keyed-Hash Message Authentication Code [FIPS 198-1]):

> 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 TDEA as the block cipher algorithm in a CMAC-based KDF is **deprecated** through December 31, 2015.
>
> Two-key TDEA **shall not** be used to derive keying material after December 31, 2015.
>
> The use of AES and three-key TDEA as the block cipher algorithm in a CMAC-based KDF is **acceptable**.

9 Hash Functions

Seven **approved** hash functions are specified in [FIPS 180-4], and four additional **approved** hash functions are specified in [FIPS 202]. The security strengths for hash functions are dependent on their use, and this information is provided in [SP 800-57]. Additional discussions about the different uses of the SHA-1 and SHA-2 hash functions specified in [FIPS 180-4] are provided in [SP 800-107], while discussions about the SHA-3 hash functions specified in [FIPS 202] are provided in that FIPS. Note that [FIPS 202] also specifies extendable output functions (XOFs); however, these are not approved as hash functions, and their use is not included in this section[4].

Table 9 provides the approval status of the **approved** hash functions.

Table 9: Approval Status of Hash Functions

Hash Function	**Use**	
SHA-1	Digital signature generation	Disallowed, except where specifically allowed by NIST protocol-specific guidance.
	Digital signature verification	Legacy-use
	Non-digital signature applications	Acceptable
SHA-2 family (SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256)	Acceptable for all hash function applications	
SHA-3 family (SHA3-224, SHA3-256, SHA3-384, and SHA3-512)	Acceptable for all hash function applications	

SHA-1 for digital signature generation:

> SHA-1 may only be used for digital signature generation where specifically allowed by NIST protocol-specific guidance. For all other applications, SHA-1 **shall not** be used for digital signature generation.

SHA-1 for digital signature verification:

> For digital signature verification, SHA-1 is allowed for **legacy-use**.

[4] The approved uses of XOFs will be addressed in future publications.

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 Bit Generation, 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]).

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

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

SHA3-224, SHA3-256, SHA3-384, and SHA3-512:

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, whether using SHA-1, SHA-2 or SHA-3. Block cipher modes for generating MACs are specified in [SP 800-38B] and [SP 800-38D][5]. The CMAC (cipher-based MAC) mode specified in [SP 800-38B] uses either AES or TDEA; the GMAC mode specified in [SP 800-38D] uses AES.

Figure 10 provides the approval status for the **approved** MAC algorithms.

Table 10: Approval Status of MAC Algorithms

MAC Algorithm	Use	
HMAC Generation	Key lengths < 112 bits	Disallowed
	Key lengths ≥ 112 bits	Acceptable
HMAC Verification	Key lengths < 112 bits	Legacy-use
	Key lengths ≥ 112 bits	Acceptable
CMAC Generation	Two-key TDEA	Restricted through 2015 Disallowed after 2015
	AES and Three-key TDEA	Acceptable
CMAC Verification	Two-key TDEA	Legacy-use
	AES and TDEA	Acceptable
GMAC Generation	AES	Acceptable
GMAC Verification	AES	Acceptable

[5] Note that the CCM authenticated encryption mode specified in [SP 800-38C] also generates a MAC. However, the CCM mode cannot be used to only generate a MAC without also performing encryption. The modes listed in this section are used only to generate a MAC.

HMAC Generation:

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

Keys less than 112 bits in length are **disallowed** for HMAC generation.

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

HMAC Verification:

The use of key lengths < 112 bits is allowed for **legacy-use**.

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

CMAC Generation:

Through December 31, 2015, the use of two-key TDEA for CMAC generation is **restricted**: the total number of blocks of data using 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 TDEA encryption operation).

The use of two-key TDEA for CMAC generation is **disallowed** after December 31, 2015.

The use of AES or three-key TDEA for CMAC generation is **acceptable**.

CMAC Verification:

The use of two-key TDEA for CMAC verification is allowed for **legacy-use**.

The use of AES or three-key TDEA for CMAC verification is **acceptable**.

GMAC Generation and Verification:

The use of GMAC for MAC generation and verification is **acceptable** when using AES.

Appendix A: 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 strong as desired.

A.1 Decryption and Key Unwrapping Using Block Cipher Key Algorithms (e.g., Two-key TDEA)

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.

Block cipher 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 was previously protected from attack (see item 2 above), but 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[6] using a more secure algorithm and key length.

A.2 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.

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

As in the case of block cipher algorithms used for encryption, 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).

A.3 Digital Signature Verification Using Asymmetric (Public) Keys

The rules specified in this publication require that digital signatures are generated using keys that provide at least 112 bits of security strength. However, before the end of 2013, the use of keys that provided only 80 bits of security strength was approved.

While it is possible to disallow the use of the low-strength keys and hash functions when generating new signatures, it is also necessary to deal with the existence of a large set of already-generated signatures that need to be verified. Hence, this publication specifies different feasible strengths of keys and an unlimited use of the SHA-1 hash function when it is used for digital signature verification of previously approved key lengths.

There are, however, risks involved that must be understood by the verifying user. The signature verification procedure might work, but since the document was signed by a key that is now considered weak, its integrity and the authenticity of the signatory could be compromised. For example, an attacker might be able to use the signatory’s public key and the publicly known set of domain parameters and determine the private key used for signature generation, due to its low security strength. The attacker could then alter the original document and sign it using the discovered private key. Other related attacks are also possible.

Therefore, while it is necessary to allow the use of weaker keys to verify the existing signatures, it is also important to remind the user to remember the risk of verifying the wrong one.

Appendix B: References

Current versions of FIPS and SP documents are available at http://csrc.nist.gov/publications/.

Previous versions of FIPS are archived at http://csrc.nist.gov/publications/PubsFIPSArch.html.

Previous versions of SPs are available at http://csrc.nist.gov/publications/PubsSPArch.html.

NIST References:

[FIPS 140] Federal Information Processing Standard (FIPS) 140-2, *Security Requirements for Cryptographic Modules*, May 2001 (with Change Notices through December 3, 2002). http://csrc.nist.gov/publications/PubsFIPS.html#140-2.

[FIPS 180-4] Federal Information Processing Standard (FIPS) 180-4, *Secure Hash Standard (SHS)*, August 2015. http://dx.doi.org/10.6028/NIST.FIPS.180-4.

[FIPS 185] Federal Information Processing Standard (FIPS) 185, *Escrowed Encryption Standard*, February 1994. http://csrc.nist.gov/publications/fips/fips185/fips185.pdf.

[FIPS 186-2] Federal Information Processing Standard (FIPS) 186-2, *Digital Signature Standard (DSS)*, January 27, 2000. http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2-change1.pdf.

[FIPS 186-4] Federal Information Processing Standard (FIPS) 186-4, *Digital Signature Standard (DSS)*, July 2013. http://dx.doi.org/10.6028/NIST.FIPS.186-4.

[FIPS 197] Federal Information Processing Standard (FIPS) 197, *Advanced Encryption Standard (AES)*, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[FIPS 198-1] Federal Information Processing Standard (FIPS) 198-1, *The Keyed-Hash Message Authentication Code (HMAC)*, July 2008. http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf.

[FIPS 202] Federal Information Processing Standard (FIPS) 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, August 2015. http://dx.doi.org/10.6028/NIST.FIPS.202.

[IG X.Y] *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program* [where X.Y is the section number being referenced]. http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf.

[SP 800-38B] NIST Special Publication (SP) 800-38B, *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*, May 2005. http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf.

[SP 800-38C] NIST Special Publication (SP) 800-38C, *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality*, May 2004 (including updates as of July 20, 2007). http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf.

[SP 800-38D] NIST Special Publication (SP) 800-38D, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, November 2007. http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf.

[SP 800-38F] NIST Special Publication (SP) 800-38F, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*, December 2012. http://dx.doi.org/10.6028/NIST.SP.800-38F.

[SP 800-56A] NIST Special Publication (SP) 800-56A Revision 2, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, May 2013. http://dx.doi.org/10.6028/NIST.SP.800-56Ar2.

[SP 800-56B] NIST Special Publication (SP) 800-56B Revision 1, *Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography*, September 2014. http://dx.doi.org/10.6028/NIST.SP.800-56Br1.

[SP 800-57] NIST Special Publication (SP) 800-57 Part 1, *Recommendation for Key Management—Part 1: General (Revision 3)*, July 2012. http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf.

[SP 800-67] NISTSpecial Publication (SP) 800-67 Revision 1, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, January 2012. http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf.

[SP 800-90A] NIST Special Publication (SP) 800-90A Revision 1, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators,* June 2015. http://dx.doi.org/10.6028/NIST.SP.800-90Ar1.

[SP 800-107] NIST Special Publication (SP) 800-107 Revision 1, *Recommendation for Applications Using Approved Hash Algorithms*, August 2012. http://csrc.nist.gov/publications/nistpubs/800-107-rev1/sp800-107-rev1.pdf.

[SP 800-108] NIST Special Publication (SP) 800-108, *Recommendation for Key Derivation Using Pseudorandom Functions*, October 2009. http://csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf.

Non-NIST References:

[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 (ANS) X9.62-1998, *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*.

Appendix C: Summary of Changes Between this Version of SP 800-131A and the Previous Version

The following is a list of non-editorial changes from the 2011 version of this document. Changes indicated by a yellow highlight are entirely new requirements (the changes were not reflected in the previous version of SP 800-131A.

1. The use of two-key TDEA for applying cryptographic protection (e.g., encryption, key wrapping or CMAC generation in KDFs) is restricted through December 31, 2015. Its use for processing already-protected information (e.g., decryption, key unwrapping and MAC verification) is allowed for **legacy use**.
2. The use of SKIPJACK is **disallowed** for encryption, but allowed for **legacy use** (e.g., decryption of already encrypted information).
3. Section 1.2.3 was added to define the single symbol used in this Recommendation: **len**(x); this has been used to replace $|p|$, $|q|$, $|n|$ and $|h|$, rather than defining them in footnotes.
4. The use of keys that provide less than 112 bits of security strength for digital signature generation are no longer allowed; however, their use for digital signature verification is allowed for **legacy use** (i.e., the verification of already-generated digital signatures). For digital signature verification using DSA, the legacy-use row has been specified to reflect the lower bound that was specified in FIPS 186-2 (i.e., 512 bits).
5. The use of the DUAL_EC_DRBG, formerly specified in [SP 800-90A], is no longer allowed.
6. The use of the RNGs specified in [FIPS 186-2], [X9.31] and [X9.62] is **deprecated** until December 31, 2015, and **disallowed** thereafter.
7. The use of keys that provide less than 112 bits of security strength for key agreement is now **disallowed**.
8. The use of non-approved key-agreement schemes is **deprecated** through December 31, 2017, and **disallowed** thereafter.
9. The use of non-approved key-transport schemes is **deprecated** through December 31, 2017, and is **disallowed** thereafter.
10. Non-approved key-wrapping methods are disallowed after December 31, 2017.
11. The use of SHA-1 for digital signature generation is **disallowed** (except where specifically allowed in NIST protocol-specific guidance); however, its use for digital signature verification is allowed for **legacy use** (i.e., the verification of already-generated digital signatures).
12. The SHA-3 family of hash functions specified in [FIPS 202] has been included in Section 9 as **acceptable**.
13. The use of HMAC keys less than 112 bits in length is no longer allowed for the generation of a MAC; however, they may be used for **legacy use** (i.e., the verification of already-generated MACs).