Check for updates

NIST Special Publication 800
NIST SP 800-131Ar3 ipd

Transitioning the Use of Cryptographic Algorithms and Key Lengths

Initial Public Draft

Elaine Barker
Allen Roginsky

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-131Ar3.ipd

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication 800
NIST SP 800-131Ar3 ipd

Transitioning the Use of Cryptographic Algorithms and Key Lengths

Initial Public Draft

Elaine Barker
Allen Roginsky
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-131Ar3.ipd

October 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Publication History

Approved by the NIST Editorial Review Board on YYYY-MM-DD [will be added in final published version]

How to Cite this NIST Technical Series Publication:
Barker EB, Roginsky AL (2024) Transitioning the Use of Cryptographic Algorithms and Key Lengths. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-131Ar3 ipd. https://doi.org/10.6028/NIST.SP.800-131Ar3.ipd

Author ORCID iDs
Elaine Barker: 0000-0003-0454-0461
Allen Roginsky: 0000-0003-2684-6736

Public Comment Period
October 21, 2024 – December 4, 2024

Submit Comments
sp800-131a_comments@nist.gov

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

Additional Information
Additional information about this publication is available at https://csrc.nist.gov/pubs/sp/800/131/a/r3/ipd, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

NIST provides cryptographic key management guidance for defining and implementing appropriate key-management procedures, using algorithms that adequately protect sensitive information, and planning for possible changes in the use of cryptography because of algorithm breaks or the availability of more powerful computing techniques. This publication provides guidance for transitions to the use of stronger cryptographic keys and more robust algorithms.

Keywords

cryptographic algorithm; digital signature; elliptic curves; encryption; entropy; extendable output functions; hash function; key agreement; key-derivation functions; key encapsulation; key transport; key wrapping; message authentication codes; quantum-resistant algorithms; random bit generation; security strength; transition.

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.

Note to Reviewers

This revision proposes a) the retirement of ECB as a confidentiality mode of operation (see Sec. 2.2) and the use of DSA for digital signature generation (Sec. 3) and b) a schedule for the retirement of SHA-1 and the 224-bit hash functions (Sec. 3 and 10). New sections are included for block cipher modes, (Sec. 2.2), key generation (Sec. 4), and extendable-output functions (Sec. 12). These and other changes are listed in Appendix B.3.

Question: Does the retirement date of December 31, 2030, for the 224-bit hash functions pose an unacceptable burden on implementers or users?

This draft revision also discusses the transition from a security strength of 112 bits to a 128-bit security strength and to quantum-resistant algorithms for digital signatures and key establishment. Since NIST is simultaneously working on multiple FIPS and SPs related to the quantum-resistant algorithms, this draft may include references to documents that have not yet been finalized, are in the process of being revised to address the availability of the quantum-resistant algorithms (e.g., SP 800-57 Part 1 and SP 800-175B), or are being developed as guidance for using them (e.g., a proposed transition schedule to the PQC algorithms). It is anticipated that these documents will either be finalized and/or publicly available as drafts by the end of 2024.

NIST is in the process of developing a schedule for a transition to the quantum-resistant algorithms. SP 800-131A will then be revised to be consistent with that guidance.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to sp800-131a_comments@nist.gov

Table of Contents

List of Tables

Acknowledgments

The authors would like to specifically acknowledge the assistance of the following colleagues in developing this revision of SP 800-131A: Alex Calis, Chris Celi, Lily Chen, David Cooper, Morris Dworkin, Kerry McKay, Nicky Mouha, Ray Perlman, and Andrew Regenscheid.

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 guidance 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 for possible changes in the use of cryptographic algorithms, including any migration to different algorithms and key lengths. The third item addresses not only the possibility of new cryptanalysis but also the increasing power of classical computing technology and the emergence of quantum computers.

This third revision of Special Publication (SP) 800-131A is intended to provide details about the transitions associated with the use of cryptography by federal agencies to protect sensitive but unclassified information. The document addresses the use of algorithms and key lengths specified in Federal Information Processing Standards (FIPS) and SPs. Unless otherwise specified (e.g., by a revision number), the latest versions of specific FIPS and SPs are referenced in the discussions.

SP 800-131A was originally published in January 2011 and revised in 2015 and 2019. This revision updates the transition guidance provided in the 2019 version and includes 1) the retirement of the ECB mode when used for confidentiality (Sec. 2.2) and DSA for digital signature generation (Sec. 3), 2) a schedule for the retirement of SHA-1 and the 224-bit hash functions (Sec. 3 and 10), and 3), and discussions about the planned transition from 112-bit security strength to 128-bit security strength and/or the use of quantum-resistant algorithms. New sections have been included for block cipher modes (Sec. 2.2), key generation (Sec. 4), and extendable-output functions (XOFs) (Sec. 12). These and other changes are listed in Appendix B.

1.2. Useful Terms for Understanding this Recommendation

1.2.1. Security Strengths

NIST Special Publication (SP) 800-57 Part 1 [SP 800-57] includes the definition of an estimated maximum security strength (hereafter shortened to "security strength") and the association of the algorithms and key lengths with these security strengths. The length of the cryptographic keys is an integral part of these determinations.

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

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

provided in [SP 800-57] and the FIPS 140 Implementation Guide [FIPS 140 IG] Annex D.B. This is the security strength that an algorithm with a particular key length can provide, given that the key used with that algorithm is correctly generated.[2]

The appropriate (classical) security strength to protect data depends on its sensitivity and needs to be determined by the data owner (e.g., a person or organization). For the Federal Government, a security strength of at least 112 bits is currently required for applying cryptographic protection (e.g., for encrypting or signing data). Section 1.2.3 discusses the proposed strategy used in this document for a transition from the 112-bit security strength.

1.2.2. Definition of Status Approval Terms

The terms "**acceptable**," "**deprecated**, "**disallowed**," and "**legacy use**" are used throughout this recommendation to indicate the approval status of an algorithm. Often, the approval status for an algorithm will also depend on the length and/or strength of its key, any domain parameters, and the mode or manner in which it is used.

- **Acceptable** means that the algorithm and key length/strength in a FIPS or SP are **approved** for use in accordance with any associated guidance. The FIPS 140 Implementation Guide [FIPS_140_IG] may indicate additional **acceptable** algorithms that are allowed for use but are not specified in a FIPS or NIST recommendation.
- **Deprecated** means that the algorithm and key length/strength may be used, but there is some security risk. The data owner must examine this risk potential and decide whether to use a **deprecated** algorithm or key length.
- **Disallowed** means that the algorithm, key length/strength, parameter set, or scheme is no longer allowed for the stated purpose.
- **Legacy use** means that the algorithm, key length/strength, parameter set, or scheme may only be used to process already protected information (e.g., to decrypt ciphertext data or to verify a digital signature). By default, applications **should** treat data processed in this way as having no more assurance of integrity and/or confidentiality than unprotected data. User interfaces **should** clearly distinguish between data processed via legacy-use cryptography and data processed using cryptography that remains acceptable.

    The relevant risks associated with **legacy use** differ depending on the type of cryptography. The risk of a loss of confidentiality due to the use of weak encryption exists whether an authorized user has decrypted the data or not. Therefore, restricting the application of legacy-use decryption is not an effective risk management strategy. Instead, risk management **should** rely on informing the user that a loss of

[2] The term "security strength" refers to the classical security strength — a measure of the difficulty of subverting the cryptographic protection (e.g., discovering the key) using classical computers. For a discussion of quantum security strength (i.e., the difficulty of subverting the protection using quantum computers), see [NIST IR 8413].

confidentiality has occurred and either revoking or replacing any secrets that may have suffered a loss of confidentiality.

In contrast, the primary risk for signatures is that an authorization decision may be made based on trusting the result of verifying a weak signature. This can be mitigated by treating the signature as invalid, regardless of the result of any verification. In most cases, applications **should not** trust a signature verified via legacy-use cryptography without displaying a warning message.

A possible case where the result of legacy-use verification may be trusted is when obtaining assurance (e.g., via local log files) that the signature was not altered before the status of the verification algorithm changed to **legacy use**. Even in this case, the level of risk incurred depends significantly on the level of assurance provided by cryptographic and non-cryptographic protections on the relevant log files.

The use of algorithms and key lengths/strengths for which the terms **deprecated** and **legacy use** are listed involve some risk that increases over time.[3] If it is determined that the risk is unacceptable for a given application, then the algorithm or key length/strength **shall** be considered **disallowed** for that application. The level of risk that can be tolerated for an application and its associated data must be determined, and methods for mitigating those risks must be defined.

This document uses the terms **acceptable**, **deprecated**, and **disallowed** as the approval status for applying cryptographic protection (e.g., encrypting data or generating a MAC or digital signature). The terms **acceptable** and **legacy use** are used as the approval status for processing already protected information (e.g., decryption or MAC or digital signature verification). When **acceptable** or **deprecated** is used as the status for applying protection, **acceptable** is used for processing already protected information. When **disallowed** is used for the status of applying protection, the **legacy use** status applies to the processing of already protected information.

1.2.3. Transition Strategy from the 112-Bit Security Strength

NIST recognizes that large-scale quantum computers, when available, will threaten the security of several NIST-approved public-key algorithms. In particular, NIST-approved digital signature schemes, key agreements using Diffie-Hellman and MQV, and key agreements and key transport using RSA will need to be replaced with secure quantum-resistant counterparts. NIST has finalized the initial quantum-resistant standards: [FIPS 203], [FIPS 204], and [FIPS 205]. Additional standards are expected in the future. NIST encourages implementers to plan for cryptographic agility to facilitate transitions to quantum-resistant algorithms where needed. Information on the post-quantum project is available at https://csrc.nist.gov/projects/post-quantum-cryptography.

[3] For example, a signature that was purportedly created when the algorithm was deemed **acceptable** is verified after the algorithm is declared to only be allowed for **legacy use**, and the actual time when the signature was generated cannot be verified.

For several years, the plan has been to transition from the 112-bit security strength to a 128-bit security strength on January 1, 2031. However, since quantum-resistant digital signature and key-encapsulation mechanisms are now standardized, this revision of SP 800-131A is modifying the transition schedule as follows:

- Transition to the 128-bit security strength for block ciphers and hash functions (for collision resistance) on January 1, 2031, as planned. TDEA is **disallowed** as of 2024 (see Sec. 2). This revision of SP 800-131A also **deprecates** SHA-1 and the 224-bit hash functions through December 31, 2030, and **disallows** them thereafter for applying cryptographic protection (see Sec. 11).
- Deprecate the use of the 112-bit security strength for the classical digital signature and key-establishment mechanisms after December 31, 2030 (rather than requiring a transition to the 128-bit security strength). Instead of a two-step transition from a 112-bit security strength to a 128-bit security strength and ultimately to the **approved** quantum-resistant algorithms, this revision is proposing a one-step approach whereby the quantum-resistant algorithms are implemented and available as soon as feasible. Currently, a 112-bit security strength for the classical digital signature and key-establishment algorithms does not appear to be in imminent danger of becoming insecure in the near future, so this approach should allow an orderly transition to quantum-resistant algorithms without unnecessary effort for the cryptographic community.
- NIST is developing a schedule for transitioning to the quantum-resistant algorithms discussed in Sec. 3, 6, and 7.

If attacks against 112-bit security strength for digital signature and key establishment become viable, a transition to the 128-bit security strength will be required. Prudent implementers and users should consider transitioning to the 128-bit security strength as originally planned.

2. Data 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. Encryption and decryption using block cipher algorithms employ a *cryptographic primitive* algorithm with a *mode of operation*. This section addresses the encryption and decryption of data using block cipher algorithms (Sec. 2.1) and the modes of operation that may be used to provide confidentiality for that data (Sec. 2.2). Some of the modes may also provide data authentication.

2.1. Block Cipher Cryptographic Primitive Algorithms

Since 2004, two block cipher primitive algorithms have been **approved** for use by the Federal Government for unclassified applications:

- AES is specified in FIPS 197, *Advanced Encryption Standard (AES)* [FIPS 197], and has three key lengths/strengths: 128, 192, and 256 bits.
- The Triple Data Encryption Algorithm (TDEA) (often referred to as "Triple DES") is specified in SP 800-67r2, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher* [SP 800-67], and has two variants known as two-key TDEA and three-key TDEA. Three-key TDEA is the stronger of the two variants.

Table 1 provides the approval status of the block cipher primitive algorithms. These algorithms are also used for purposes other than data encryption and decryption (see Sec. 5, 8, 9, and 12).

Table 1. Approval status of block cipher algorithms for encryption and decryption

Algorithm	Status
TDEA Encryption	Disallowed
TDEA Decryption	Legacy use
AES-128 Encryption and Decryption	Acceptable
AES-192 Encryption and Decryption	Acceptable
AES-256 Encryption and Decryption	Acceptable

TDEA:

- Encryption using TDEA is **disallowed**.
- Decryption using TDEA is allowed for **legacy use**.

AES:

Encryption and decryption using AES with 128, 192, and 256-bit keys are **acceptable**.[4]

[4] Even with the impact of quantum computers, AES-128, AES-192, and AES-256 will remain secure for the foreseeable future. See Appendix A for further discussion.

2.2. Block Cipher Modes of Operation for Encryption and Decryption

Encryption and decryption using block cipher algorithms require the use of modes of operation to perform successive encryption or decryption processes on data, which in turn require multiple calls to the primitive algorithm. [SP 800-38A] and a separately published addendum [SP 800-38A_addendum] specify modes that are only used to perform encryption and decryption on the input data. [SP 800-38C] and [SP 800-38D] specify authenticated encryption modes and are used to both encrypt/decrypt data and provide a method for determining the authenticity of data processed by the mode. [SP 800-38E] specifies the encryption and decryption of data for storage devices with fixed-length data units. [SP 800-38G] specifies the use of encryption and decryption that preserve the format of the original unencrypted data.

The approval status of these modes for block-cipher encryption and decryption is provided in Table 2.

Table 2. Approval status of the block cipher modes of operation for AES encryption and decryption

Publication	Mode	Status
SP 800-38A	ECB	Disallowed for data encryption Legacy use for decryption
	CBC	Acceptable
	CFB	Acceptable
	CTR	Acceptable
	OFB	Acceptable
SP 800-38A Addendum	CBC-CS1	Acceptable (will be incorporated into SP 800-38A)
	CBC-CS2	
	CBC-CS3	
SP 800-38C	CCM	Acceptable
SP 800-38D	GCM	Acceptable
SP 800-38E	XTS-AES	Acceptable
SP 800-38G	FF1	Acceptable (domain size of at least one million)
		Disallowed (domain size of less than one million)
	FF3	Disallowed

[SP 800-38A] includes modes of encryption/decryption for use with AES and TDEA (see Sec. 2.1 for the approval status of these algorithms):

The ECB mode is **disallowed** for encrypting secret data but is allowed for **legacy use** (i.e., to decrypt data that has been encrypted prior to the publication of this revision of SP 800-131A). However, NIST Internal Report (IR) 8459 discusses applications for which the ECB mode remains **acceptable** for non-confidentiality purposes (i.e., challenge-response protocols and initialization vector [IV] generation) [NIST IR 8459].

CBC (including CBC-CS1, CBC-CS2, and CBC-CS3, defined in [SP 800-38A_addendum]), CFB, CTR, and OFB are **acceptable**.

[SP 800-38C], [SP 800-38D], and [SP 800-38E] specify, correspondingly, the CCM, GCM, and XTS-AES modes for block ciphers with a block size of 128 bits (i.e., using AES):

> The use of the CCM, GCM, and XTS-AES modes is **acceptable** when used as specified in SP 800-38C, SP 800-38D, and SP 800-38E, respectively. In addition, an implementation of the AES GCM mode **shall** comply with one of the scenarios defined in the FIPS 140 Implementation Guide, Annex C.H [FIPS 140_IG].

[SP 800-38G] specifies two modes for format-preserving encryption and decryption:

1. **FF1:** The FF1 mode is acceptable when used as specified in SP 800-38G with the additional restriction that the domain size be at least one million. The use of FF1 with a domain size of less than one million is **disallowed**.
2. **FF3:** The use of FF3, as currently specified in SP 800-38G, is **disallowed**.[5]

[5] SP 800-38G will be revised to remove FF3.

3. Digital Signatures

Digital signatures provide assurance of origin authentication and data integrity. These assurances can be extended to provide assurance that the signatory cannot effectively deny signing a document, which is commonly known as non-repudiation. The digital signature algorithms are specified in [FIPS 186], [FIPS 204], [FIPS 205], and [SP 800-208].

The security strength estimated for a digital signature algorithm depends on the hash function used, the key length/strength and method used for key generation, and any other parameters used during the digital signature process.

- **DSA:** DSA keys are generated and used with domain parameters *p*, *q*, and *g.* The security strength that the algorithm can provide depends on the bit lengths of *p* (*L*) and *q* (*N*) and the proper generation of the domain parameters used. The specification for DSA is not included in the current version of FIPS 186 (i.e., [FIPS 186-5]). However, DSA is specified in the previous version (i.e., [FIPS 186-4]).
- **Elliptic Curve-based Digital Signatures (ECDSA and EdDSA):** Keys are generated and used with respect to domain parameters that define elliptic curves. The length of *n* (i.e., the domain parameter that specifies the order of the base point *G*) is used to determine the security strength that can be provided by a properly generated key. ECDSA and EdDSA are specified in FIPS 186.
- **RSA:** RSA keys are generated with respect to a modulus *n*, which is used to determine the security strength that can be provided by a digital signature. The RSA algorithm for digital signatures is specified in [RFC 8017], and guidance for use is provided in FIPS 186.
- **ML-DSA:** ML-DSA is a lattice-based quantum-resistant digital signature algorithm that is specified in [FIPS 204].
- **SLH-DSA:** SLH-DSA is a quantum-resistant stateless hash-based digital signature algorithm that is specified in [FIPS 205].
- **Stateful hash-based signatures:** The LMS, HSS, XMSS, and $XMSS^{MT}$ quantum-resistant digital signature algorithms are specified in [SP 800-208].

The security strength provided by a digital signature generation process is no greater than the minimum of 1) the security strength that the digital signature algorithm can support with a given parameter set (including the length of the key) and 2) the security strength supported by the cryptographic hash method[6] that is used. See [SP 800-57] for the estimated security strength for a given algorithm and parameter set.

Sections 11 and 12 discuss the hash methods used during the digital signature generation and verification processes: hash functions and extendable-output functions (XOFs).

Table 3 provides the approval status of the algorithms and key lengths used for the generation and verification of digital signatures in accordance with [FIPS 186], [FIPS 204], [FIPS 205], and [SP

[6] A hash method is 1) a hash function specified in either FIPS 180 or FIPS 202 or 2) an XOF specified in FIPS 202.

800-208]. The approval status of DSA, ECDSA, EdDSA, and RSA will be impacted by the transition to quantum-resistant digital signature algorithms. This is indicated in the table using an asterisk (*).

Table 3. Approval status of algorithms used for digital signature generation

Digital Signature Algorithm	Criteria	Status
DSA generation	All security strengths	Disallowed
DSA verification		Legacy use
ECDSA generation	< 112 bits of security strength	Disallowed
	≥ 112 but < 128 bits of security strength	Acceptable through 2030 Deprecated after 2030*
	≥ 128 bits of security strength	Acceptable*
ECDSA verification	< 112 bits of security strength	Legacy use
	≥ 112 bits of security strength	Acceptable*
EdDSA generation and verification	≥ 128 bits of security strength	Acceptable*
RSA generation (PKCS #1 v1.5 & PSS)	< 112 bits of security strength	Disallowed
	≥ 112 but < 128 bits of security strength	Acceptable through 2030 Deprecated after 2030*
	≥ 128 bits of security strength	Acceptable*
RSA verification (PKCS #1 v1.5 & PSS)	< 112 bits of security strength	Legacy use
	≥ 112 bits of security strength	Acceptable*
RSA generation (ASC X9.31)	All security strengths	Disallowed
RSA verification (ASC X9.31)		Legacy use
ML-DSA generation and verification	Parameter sets in FIPS 204	Acceptable
SLH-DSA	Parameter sets in FIPS 205	Acceptable

generation and verification		
LMS, HSS, XMSS, XMSS^{MT} generation and verification	Parameter sets in SP 800-208	Acceptable

DSA:

- Signature generation:
    - Signature generation using DSA is **disallowed**.
- Signature verification:
    - Verification of DSA digital signatures is allowed for **legacy use** when using previously **approved** domain parameters and private keys.

ECDSA and EdDSA:

- Signature generation: The security strength provided by an elliptic curve signature is 1/2 of the length of the domain parameter *n*. Recommended and deprecated elliptic curves for digital signature generation are provided in [SP 800-186]. Elliptic curves that meet the security strength requirements are also allowed when they satisfy the requirements of FIPS 140 Implementation Guide [FIPS 140 IG], Annex C.A.
    - ECDSA signature generation providing less than 112 bits of security is **disallowed**.
    - ECDSA signature generation providing at least 112 bits of security (but less than 128 bits of security) is **acceptable** through December 31, 2030. For these curves, $224 \le \mathbf{len}(n) < 256$.

      After December 31, 2030, the use of these curves and keys is **deprecated** for digital signature generation.
    - ECDSA and EdDSA signature generation providing at least 128 bits of security is **acceptable**. These signatures **shall** be generated using elliptic curves and private keys such that $\mathbf{len}(n) \ge 256$.
- Signature verification:
    - Signature verification of ECDSA digital signatures that were generated to provide less than 112 bits of security is allowed for **legacy use** when using curves and public keys such that $160 \le \mathbf{len}(n) < 224$.
    - Signature verification of ECDSA digital signatures that were generated to provide at least 112 bits of security is **acceptable** using the recommended elliptic curves included in [SP 800-186]. In this case, $\mathbf{len}(n) \ge 224$.
    - Signature verification of EdDSA digital signatures is **acceptable** using the recommended elliptic curves included in [SP 800-186] where $\mathbf{len}(n) \ge 256$. The use of EdDSA was never approved for a security strength less than 128 bits.

  - Signature verification using curves that comply with FIPS 140 Implementation Guide [FIPS140_IG], Annex C.A is allowed for **legacy use**.

RSA (PKCS #1 v1.5 & PSS):

- Signature generation: The security strength provided by an RSA signature depends on the bit length of the modulus *n*. The security strength associated with several values of **len**(*n*) is provided in [SP 800-57]. The security strength associated with other values of **len**(*n*) may be estimated using the formula in FIPS 140 Implementation Guide [FIPS140_IG], Annex D.B.
  - Signature generation providing less than 112 bits of security is **disallowed**.
  - Signature generation providing at least 112 bits of security (but less than 128 bits of security) is **acceptable** through December 31, 2030. These signatures **shall** be generated using private keys and a modulus *n* such that 2048 ≤ **len**(*n*) < 3072.

    After December 31, 2030, the use of these moduli and keys is **deprecated** for digital signature generation.
  - Signature generation providing at least 128 bits of security is **acceptable**. These signatures **shall** be generated using a modulus *n* and public keys such that **len**(*n*) ≥ 3072.
- Signature verification:
  - Signature verification using public keys providing less than 112 bits of security is allowed for **legacy use** when the modulus *n* and the public keys are such that 1024 ≤ **len**(*n*) and **len**(*n*) is a multiple of 256.
  - Signature verification using public keys providing at least 112 of security is **acceptable**. Verification requires the use of a modulus *n* and public keys such that **len**(*n*) ≥ 2048.

RSA (ASC [X9.31]): Approved in FIPS 186-4.

- Signature generation:
  - Signature generation in accordance with ASC [X9.31] is **disallowed**.
- Signature verification:
  - Signature verification of signatures generated in accordance with ASC [X9.31] is allowed for **legacy use**.

ML-DSA:

- Signature generation and verification are **acceptable** using the parameter sets listed in [FIPS 204].

SLH-DSA:

- Signature generation and verification are **acceptable** using the parameter sets listed in [FIPS 205].

LMS, HSS, XMSS, $XMSS^{MT}$:

- Signature generation and verification are **acceptable** using the parameter sets listed in [SP 800-208].

4. Cryptographic Key Generation

[SP 800-133] addresses the generation of the cryptographic keys used in cryptography. The keys are either 1) generated using mathematical processing on the output of **approved** random bit generators (RBGs) and possibly other parameters or 2) generated based on keys that are generated in this fashion. These keys **shall** be obtained directly or indirectly from the output of an **approved** RBG as specified in the [SP 800-90] series and generated in accordance with appropriate FIPS or NIST SPs (e.g., [SP 800-108] for key derivation from a preexisting shared key). [SP 800-133] includes methods for producing a key by:

- Combining the output of an **approved** RBG with independently determined data of the same length by exclusive-oring the values and
- Combining independently generated keys with other (independently generated) keys and/or independently determined data by concatenation, XOR-oring, or using a specified key-extraction process.

These methods for determining keys are **acceptable** when consistent with the requirements of the application for which the keys will be used.

5. Random Bit Generation

Random numbers are used for various purposes, such as the generation of keys, nonces, and authentication challenges. The SP 800-90 series of documents specifies methods for generating random bits for these purposes.

5.1. Deterministic Random Bit Generator Mechanisms (DRBGs)

Several deterministic random bit generator (DRBG) mechanisms have been specified for use by the Federal Government. [SP 800-90A] includes three DRBGs: Hash_DRBG, HMAC_DRBG, and CTR_DRBG, which are specified to include either a hash function or a block cipher (e.g., AES) as cryptographic primitives.

The approval status for the DRBGs and the cryptographic primitives they use is provided in Table 4.

Table 4. Approval status of algorithms used for random bit generation

DRBG	Crypto. Primitive	Status
Hash_DRBG and HMAC_DRBG	SHA-1, SHA-224, SHA-512/224, and SHA3-224	Deprecated through 2030 Disallowed after 2030
	All other SHA-2 and SHA-3 hash functions	Acceptable
CTR_DRBG	TDEA	Disallowed
	AES-128, AES-192, AES-256	Acceptable

Hash_DRBG and HMAC_DRBG:

- The use of SHA-1 or a 224-bit hash function (i.e., SHA-224, SHA-512/224, or SHA3-224) as the cryptographic primitive in Hash_DRBG and HMAC DRBG is **deprecated** through December 31, 2030, and **disallowed** after 2030.
- The use of Hash_DRBG and HMAC_DRBG is **acceptable** with any other SHA-2 or SHA-3 hash function specified in [FIPS 180] or [FIPS 202] (i.e., SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512).

CTR_DRBG:

- The use of CTR_DRBG using TDEA as the cryptographic primitive is **disallowed**.
- The use of CTR_DRBG using AES-128, AES-192, or AES-256 is **acceptable**.

5.2. Entropy Sources

Entropy sources provide entropy for RBGs. [SP 800-90B] provides guidance for designing and testing entropy sources. Additional guidance related to the validation of entropy sources is provided in the FIPS 140 Implementation Guide [FIPS 140 IG], Sec. 9.3.A and Annexes D.J, D.K, and D.O.

5.3. Random Bit Generator (RBG) Constructions

[SP 800-90C] provides constructions for designing RBGs based on the use of entropy sources that are compliant with [SP 800-90B] and the DRBGs specified in [SP 800-90A]. [SP 800-90C] is currently in draft form; the RBG constructions in [SP 800-90C] are **acceptable**.

6. Key Agreement Using Diffie-Hellman and MQV

Key agreement is a technique for establishing keying material via an electronic key-agreement transaction between two entities that intend to communicate. Both parties contribute information so that neither party can predetermine the value of the resulting secret keying material independently from the contributions of the other party. The agreed-upon keys are not transmitted between the two entities.

[SP 800-56A] specifies two families of key-agreement schemes: Diffie-Hellman (DH) and Menezes-Qu-Vanstone (MQV). Each has been defined over two different mathematical structures: finite fields and elliptic curves.

Key agreement, as specified in [SP 800-56A], 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 resulting shared secret. [SP 800-56A] contains the DH and MQV primitives and refers to [SP 800-56C] for KDMs. The approval status for these key-derivation methods is discussed in Sec. 9.1.

The security strength of a key-agreement scheme specified in [SP 800-56A] depends on the key-agreement algorithm, the parameters used with that algorithm (e.g., the keys), and its form (i.e., finite field or elliptic curve).

- **Finite field DH and MQV:** The keys for these algorithms are generated and used with domain parameters p, q, and g. The security strength that can be provided by the algorithm depends on the length of p, the length of q, and the proper generation of the domain parameters and the key.
- **Elliptic Curve DH and MQV:** The keys for these algorithms are generated and used with respect to domain parameters that define elliptic curves. The length of n (i.e., the order of the base point G) is used to determine the security strength that can be provided by a properly generated curve.

Table 6 contains the Federal Government approval status for the DH and MQV key-agreement schemes. In some cases, a scheme is only allowed for **legacy use**; additional details below the table indicate the conditions for allowing continued use (e.g., the associated dates and parameters). The approval status of the schemes in this section will be impacted by the transition to quantum-resistant key-establishment methods. This is indicated in the table using an asterisk (*).

Table 5. Approval status for SP 800-56A key agreement (DH and MQV) schemes

Scheme	Domain Parameters	Status
	< 112 bits of security strength	Legacy use

Scheme	Domain Parameters	Status
SP 800-56A DH and MQV schemes using finite fields and elliptic curves	112 bits of security strength	Acceptable through 2030 Deprecated after 2030*
	≥ 128 bits of security strength	Acceptable*
Non-conformance to SP 800-56A	Any	Legacy use

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

- For key-agreement transactions providing less than 112 bits of security strength (i.e., using domain parameters where **len**(*p*) < 2048 or **len**(*q*) < 224):

  The initiation of a key-agreement transaction providing less than 112 bits of security is **disallowed**.

  The processing of information in a key-agreement transaction is allowed for **legacy use** when **len**(*p*) = 1024 or **len**(*q*) = 160. See parameter set FA in [SP 800-56Ar2].

- Key-agreement transactions providing 112 bits of security strength are **acceptable** through December 31, 2030, using the following domain parameters:
  - The MODP-2048 safe-prime group specified in [RFC 3526] (listed in [SP 800-56A])
  - The ffdhe2048 safe-prime group specified in [RFC 7919] (listed in [SP 800-56A])
  - For FIPS 186-type domain parameters, (**len**(*p*), **len**(*q*)) = (2048, 224) or (2048, 256); these domain parameters are provided as parameter sets FB and FC in [SP 800-56A]

  After December 31, 2030, the use of these domain parameters is **deprecated**.

- Key-agreement transactions providing at least 128 bits of security strength are **acceptable** in the following cases:
  - The following safe-prime groups are used:
    - The MODP-X safe-prime group specified in [RFC 3526] or
    - The ffdheX safe-prime group specified in [RFC 7919],

    where X = 3072, 4096, 6144, or 8192 (listed in [SP 800-56A]).

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

- For key-agreement transactions providing less than 112 bits of security strength (i.e., using curves where **len**(*n*) < 224):

  Initiating a key-agreement transaction providing less than 112 bits of security strength is **disallowe**d.

The processing of information in a key-agreement transaction is allowed for **legacy use** when **len**(*n*) = 160 to 223. See parameter set EA in [SP 800-56Ar2].

- Key-agreement transactions providing 112 bits of security strength are **acceptable** through December 31, 2030, using the following curves:
  - The P-224 curve specified in [SP 800-186] (see parameter set EB in [SP 800-56A]) or
  - The brainpoolP224r1 and brainpoolP224t1 curves specified in [RFC 5639] (see FIPS 140 Implementation Guide [FIPS 140_IG], Annex C.A.).

  After December 31, 2030, the use of these curves is **deprecated**.

- Key-agreement transactions providing at least 128 bits of security strength using the following elliptic curves are **acceptable**:
  - P-256, P-384, P-521, K-283, K-409, K-571, B-283, B-409, B-571, and sep256k1, as specified in [SP 800-186] (see parameter sets EC, ED, and EE in [SP 800-56A])
  - The brainpool curves and twisted variants of these curves specified in [RFC 5639]: brainpoolP256r1, brainpoolP320r1, brainpoolP384r1, brainpoolP512r1, brainpoolP256t1, brainpoolP320t1, brainpoolP384t1, and brainpoolP512t1 (see [FIPS 140 IG], Annex C.A)

Schemes not compliant with [SP 800-56A]:

Initiating a key-agreement transaction using these schemes is **disallowed**.

Processing the information in a key-agreement transaction using these schemes is allowed for **legacy use** when using parameters that were previously **acceptable**.

7. Key Agreement and Key Transport Using RSA

[SP 800-56B] specifies the use of RSA for both key-agreement and key-transport transactions. Key agreement is a technique in which both parties contribute information to the generation of keying material so that neither party can predetermine the value of the secret keying material independently from the contributions of the other party. Two key-agreement schemes are specified: KAS1 and KAS2. Key transport is a key-establishment technique in which only one party determines the key and sends it to the other party. One key-transport scheme is specified (i.e., RSA-OAEP), and another general hybrid method is described.

RSA keys are generated with respect to the desired bit length of a modulus *n*. The length of *n* is used to determine the security strength of a key-establishment scheme that uses *n*, assuming that *n* and the RSA keys are generated as specified in [SP 800-56B].[7]

[SP 800-56B] provides guidance on key lengths for RSA and explicitly specifies several key lengths and the intended security strengths beginning with **len**(*n*) = 2048, which is estimated to support a security strength of 112 bits. Additional key lengths greater than 2048 bits and not explicitly listed in [SP 800-56B] may also be used. The approximate security strength that is supported by a given key length may be estimated using a formula in [SP 800-56B] and Annex D.B of [FIPS 140 IG].

Table 7 provides the approval status for the choice of **len**(*n*). The approval status of the schemes in this section will be impacted by the transition to quantum-resistant key-establishment methods. This is indicated by an asterisk (*) in the table.

Table 6. Approval status for the RSA-based key-agreement and key-transport schemes

Scheme	Domain Parameters	Status
SP 800-56B Key-Agreement and Key-Transport schemes	< 112 bits of security strength	Legacy use
	112 bits of security strength	Acceptable through 2030 Deprecated after 2030*
	≥ 128 bits of security strength	Acceptable*
Non-conformance to SP 800-56B	Any	Legacy use

[SP 800-56B] RSA key-agreement and key-transport schemes:

- For key establishment transactions providing less than 112 bits of security strength (i.e., **len**(*n*) < 2048):

  Initiating a key-establishment transaction providing less than 112 bits of security strength is **disallowed**.

[7] [SP 800-56B] refers to [FIPS 186] for generation guidance.

The processing of information in a key-establishment transaction is allowed for **legacy use** when **len**(n) = 1024 or **len**(q) = 160. See parameter set FA in [SP 800-56Ar2].

- Key-establishment transactions that provide 112 bits of security strength (i.e., 2048 ≤ **len**(*n*) < 3072):

  Key establishment is **acceptable** through December 31, 2030, using the schemes specified in [SP 800-56B].

  After December 31, 2030, key establishment using these values of **len**(n) is **deprecated**.

- Key-establishment transactions providing at least 128 bits of security strength (i.e., **len**(*n*) ≥ 3072):

  Key establishment is **acceptable** using the schemes specified in [SP 800-56B].

Non-[SP 800-56B]-compliant RSA key-establishment schemes that were previously allowed in [FIPS 140 IG]:

- Initiating a key-establishment transaction using a non-[SP 800-56B]-compliant scheme is **disallowed**.
- The processing of information in a key-establishment transaction using a non-[SP 800-56B]-compliant scheme is allowed for **legacy use** when **len**(*n*) ≥ 1024.

8. Key Establishment Using a Key Encapsulation Mechanism (KEM)

[FIPS 203] specifies a quantum-resistant key encapsulation mechanism (KEM) — a set of algorithms that can be used by two parties to establish a secret key over a public channel under certain conditions. A key that is securely established using a KEM can then be used with symmetric-key cryptographic algorithms to perform basic tasks in secure communications, such as encryption and authentication.

[FIPS 203] specifies three parameter sets for a key-encapsulation mechanism (ML-KEM): ML-KEM-512, ML-KEM-768, and ML-KEM-1024.

The use of any of these approved KEMs is **acceptable** for establishing keying material between two parties.[8]

[8] Guidance for the use of a KEM for key establishment is under development.

9. Key Derivation Methods

One or more keys or other keying material may be derived using pre-established key-derivation keys (KDKs). A KDK may be established using a key-establishment scheme (see Sec. 6 and 7) or manual key-distribution method or generated using an RBG (see Sec. 5) or a previous instance of a key-derivation function.

9.1. Key-Derivation Methods in SP 800-56C

[SP 800-56C] provides key-derivation methods (KDMs) for key-establishment schemes in [SP 800-56A] and [SP 800-56B] (see Sec. 6 and 7 herein, respectively). [SP 800-56C] specifies one-step key-derivation functions (KDFs) and two-step key-derivation procedures. When a key-derivation method is allowed for **legacy use**, other conditions specified in Sec. 6 and 7 for the key-establishment schemes also apply. **Approved** key derivation methods are also provided in [SP 800-135] for specific applications.

9.1.1. One-Step Key-Derivation Functions

One-step key-derivation functions use a hash function, as specified in [FIPS 180] or [FIPS202]; HMAC, as specified in [SP 800-224]; or KMAC, as specified in [SP 800-185].

Table 8 provides the approval status of the one-step key-derivation functions specified in [SP 800-56C].

Table 7. Approval status for the one-step KDFs in SP 800-56C

H(x)	Crypto. Primitive	Status
Hash function and HMAC	SHA-1, SHA-224, SHA-512/224, and SHA3-224	Deprecated through 2030 Legacy use after 2030
	All other hash functions	Acceptable
KMAC	KMAC128 and KMAC256	Acceptable

H(*x*) is a hash function or HMAC:

- The use of SHA-1 and the 224-bit hash functions (i.e., SHA-224, SHA-512/224, or SHA3-224):

  The use of these hash functions for one-step key derivation during a key-establishment transaction is **deprecated** through December 31, 2030.

  After 2030, the use of these hash functions is allowed for **legacy use** to derive keying material using the information from a key-establishment transaction (also see Sec. 6 and 7).

- The use of all other hash functions specified in [FIPS 180] and [FIPS 202] for one-step key derivation is **acceptable** (i.e., SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512).

H(*x*) is KMAC:

- The use of KMAC128 and KMAC256 for one-step key derivation is **acceptable**.

9.1.2. Two-Step Key-Derivation Procedures

Two-step key-derivation procedures use separate steps for randomness extraction and key expansion based on HMAC, as specified in [SP 800-224], or CMAC, as specified in [SP 800-38B].

Table 9 provides the approval status of the two-step key-derivation methods specified in [SP 800-56C].

Table 8. Approval status for the two-step KDMs in SP 800-56C

MAC Algorithm	Crypto. Primitive	Status
HMAC-hash	SHA-1, SHA-224, SHA-512/224, and SHA3-224	Deprecated through 2030 Legacy use after 2030
	All other hash functions	Acceptable
AES-CMAC	AES-128, AES-192, and AES-256	Acceptable

HMAC-hash:

- The use of SHA-1 and the 224-bit hash functions (i.e., SHA-224, SHA-512/224, and SHA3-224) for two-step key derivation using HMAC is **deprecated** through December 31, 2030, and allowed for **legacy use** thereafter.
- The use all other hash functions specified in [FIPS 180] and [FIPS 202] for two-step key derivation using HMAC is **acceptable** (i.e., using SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512).

AES-CMAC:

1. The use of AES-128, AES-192, or AES-256 as the cryptographic primitive for two-step key derivation using CMAC is **acceptable**.

9.2. Key-Derivation Functions in SP 800-108

[SP 800-108] specifies KDFs that use pseudorandom functions (PRFs) and a cryptographic key (called a key-derivation key) to generate additional keys. Three PRFs are used in the KDFs specified in [SP 800-108]:

1. HMAC, as specified in [SP 800-224], requires the use of a hash function (see Sec. 10).
2. CMAC, as specified in [SP 800-38B], requires the use of a block cipher algorithm (e.g., AES-128, which is specified in [FIPS 197]).
3. KMAC, as specified in [SP 800-185].

HMAC, CMAC, and KMAC are also known as message authentication code (MAC) algorithms. Section 13 discusses these algorithms and the keys used with them.

Table 10 provides the approval status of the PRFs for key derivation, as specified in [SP 800-108].

Table 9. Approval status of the algorithms used for a key derivation function (KDF)

KDF Type	Crypto. Primitive	Status
HMAC-based KDF	SHA-1, SHA-224, SHA-512/224, and SHA3-224	Deprecated through 2030 Legacy use after 2030
	All other approved hash functions	Acceptable
CMAC-based KDF	TDEA	Legacy use
	AES-128, AES-192, and AES-256	Acceptable
KMAC-based KDF	KMAC128 and KMAC 256	Acceptable

HMAC-based KDF:

- The use of SHA-1 and the 224-bit hash functions (i.e., SHA-224, SHA-512/224, and SHA3-224) for key derivation using HMAC is **deprecated** through December 31, 2030, and allowed for **legacy use** thereafter.
- The use of all other hash functions specified in [FIPS 180] and [FIPS 202] for key derivation using HMAC is **acceptable** (i.e., using SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512).

CMAC-based KDF:

- The use of TDEA, as specified in [SP 800-67], is **disallowed** for initiating a transaction that uses a CMAC-based KDF for the key-derivation process.

  The use of TDEA for CMAC-based key derivation is allowed for **legacy use** when processing a transaction.
- The use of AES for CMAC-based key derivation is **acceptable**.

KMAC-based KDF:

- For KMAC-based KDFs, the use of KMAC128 and KMAC256 (as specified in [SP 800-185]) is **acceptable**.

9.3. Key-Derivation in SP 800-132

[SP 800-132] specifies a family of password-based key-derivation functions (PBKDFs) for deriving cryptographic keys from passwords or passphrases to protect electronically stored data or data protection keys. The PBKDFs require the use of a randomly generated salt of at least 128 bits, a minimum iteration counter of 1000, and an HMAC with an **approved** hash function.

Table 11 provides the approval status for PBKDFs.

Table 10. Approval status of the PBKDFs

HMAC Crypto. Primitive	Status
SHA-1, SHA-224, SHA-512/224, and SHA3-224	Deprecated through 2030 Legacy use after 2030
All other hash functions	Acceptable

Password-based key derivation using HMAC:

- The use of SHA-1 and the 224-bit hash functions (i.e., SHA-224, SHA-512/224, and SHA3-224) for password-based key derivation using HMAC is **deprecated** through December 31, 2030, and allowed for **legacy use** thereafter.
- The use of all other hash functions specified in [FIPS 180] and [FIPS 202] for password-based key derivation using HMAC is **acceptable** (i.e., using SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512).

10. Key Wrapping

Key wrapping is the encryption and integrity protection of keying material using a block-cipher key-wrapping algorithm and a symmetric key. **Approved** methods for key wrapping using a block cipher are provided in [SP 800-38F].

[SP 800-38F] specifies three algorithms for key wrapping that use block ciphers: KW and KWP, which use AES (as specified in [FIPS 197]); and TKW, which uses TDEA (as specified in [SP 800-67]). [SP 800-38F] also approves the CCM and GCM authenticated-encryption modes specified in [SP 800-38C] and [SP 800-38D] for key wrapping using AES, as well as combinations of an **approved** encryption mode (e.g., AES-CBC) with an **approved** authentication method (e.g., HMAC or a digital signature).

Table 12 provides the approval status of the block cipher algorithms that may be used for key wrapping.

Table 11. Approval status of block cipher algorithms used for key wrapping

Algorithm	Use	Status
TDEA (using TKW)	Key wrapping	Disallowed
	Key unwrapping	Legacy use
KW, KWP, CCM and GCM (using AES)	Key wrapping and unwrapping	Acceptable
The combination of an **approved** encryption mode and **approved** authentication method other than KW, KWP, CCM, or GCM	Key wrapping using separate encryption and authentication processes	Deprecated
	Key unwrapping using separate decryption and authenticity/integrity verification processes	Acceptable

TDEA (using TKW, as specified in [SP 800-38F]):

- The use of TDEA for key wrapping is **disallowed**.
- The use of TDEA for unwrapping is allowed for **legacy use**.

KW and KWP (specified in [SP 800-38F]), CCM (specified in [SP 800-38C]), and GCM (specified in [SP 800-38D]):

- The use of KW, KWP, CCM, and GCM using AES for both key wrapping and unwrapping is **acceptable**.

The combination of an **approved** encryption mode and an **approved** authentication method other than KW, KWP, CCM, and GCM:

- The use of an **approved** encryption mode and an **approved** authentication method for key wrapping is **deprecated** until additional guidance is provided for using these combinations securely.

  The **approved** AES encryption modes include:

  - The CBC, CFB, OFB, and CTR modes specified in [SP 800-38A].

The authentication methods include:

- The CMAC mode specified in [SP 800-38B];
- The GMAC mode specified in [SP 800-38D];
- A digital signature scheme specified in [FIPS 186], [FIPS 204], [FIPS 205], or [SP 800-208] (see Sec. 3);
- HMAC, as specified in [SP 800-224]; and
- KMAC, as specified in [SP 800-185].

11. Hash Functions

A hash function produces a condensed representation of its input by taking an input of arbitrary length and outputting a value with a predetermined length. Hash functions are used in the generation and verification of digital signatures, key derivation, random number generation, computation of message authentication codes, and hash-only applications.

Several hash functions have been **approved** and specified:

- [FIPS 180] specifies SHA-1 and the SHA-2 family of hash functions (i.e., SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256). Information about the security strengths that can be provided by these hash functions is given in [SP 800-57].
- [FIPS 202] specifies the SHA-3 family of hash functions (i.e., SHA3-224, SHA3-256, SHA3-384, and SHA3-512). Discussions about the SHA-3 hash functions are provided, and the security strengths that can be provided by these functions are given in [SP 800-57].
- [SP 800-185] specifies two SHA-3-derived hash functions (i.e., TupleHash and ParallelHash) that are based on the XOFs specified in [FIPS 202] and discusses their use and the security strengths that they can support.

Table 13 provides the approval status of the hash functions.

Table 12. Approval status of hash functions

Hash Function	Use	Status
SHA-1	Digital signature generation	Disallowed
	Digital signature verification	Legacy use
	Applying protection for non-digital-signature applications	Deprecated through 2030 Disallowed after 2030
	Processing already protected information using non-digital signature applications	Acceptable through 2030 Legacy use after 2030
SHA-224, SHA-512/224, and SHA3-224	Applying protection	Deprecated through 2030 Disallowed after 2030
	Processing already-protected information	Acceptable through 2030 Legacy use after 2030
All other SHA-2 and SHA-3 hash functions	Acceptable for all hash function applications	
TupleHash and ParallelHash	Acceptable	

SHA-1:

- Digital signature generation:

  SHA-1 is **disallowed** for digital signature generation.

- Digital signature verification:

  When used for digital signature verification, SHA-1 is allowed for **legacy use**.

- Applying cryptographic protection for non-digital signature applications:

  The use of SHA-1 is **deprecated** through December 31, 2030, for applying protection in non-digital signature applications and **disallowed** thereafter.

- Processing already protected information using SHA-1 for non-digital signature applications:

  The use of SHA-1 is **acceptable** for processing already-protected information through December 31, 2030, and allowed for **legacy use** thereafter.

Hash functions with a 224-bit output (i.e., SHA-224, SHA-512/224, and SHA3-224):

- The use of 224-bit hash functions for applying cryptographic protection is **deprecated** through December 31, 2030, and **disallowed** thereafter.
- The use of 224-bit hash functions to process already protected information is **acceptable** through December 31, 2030, and allowed for **legacy use** thereafter.

All other SHA-2 and SHA-3 hash functions (i.e., SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, and SHA3-512):

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

TupleHash and ParallelHash:

The use of TupleHash and ParallelHash is **acceptable** for the purposes specified in [SP 800-185].

12. Extendable-Output Functions (XOFs)

Like hash functions, XOFs operate on input of an arbitrary length. The output of an XOF can be extended to any desired length, whereas the output of a hash function is a predetermined fixed length. Two XOFs are **approved** in [FIPS 202]: SHAKE128 and SHAKE256. [SP 800-185] provides **approved** uses for these XOFs. Table 14 provides the approval status of the XOFs.

Table 13. Approval status of eXtendable-Output Functions (XOFs)

XOF	Status
SHAKE128	Acceptable
SHAKE256	

The use of SHAKE128 and SHAKE256 is **acceptable** when used as specified in **approved** cryptographic applications.

13. Message Authentication Codes (MACs)

A MAC is a cryptographic checksum on the data over which it is computed and is used to provide assurance of data integrity and source authentication. It is generated using a MAC algorithm and a cryptographic key. The MAC can provide assurance that the data has not been modified since the MAC was generated and that the MAC was computed by one of the parties sharing the key. The key **shall** be generated and/or established using an **approved** method (e.g., using an **approved** RBG or key-establishment scheme).

Three types of message authentication code mechanisms are specified for use:

1. [SP 800-224] specifies a keyed-hash message authentication code (HMAC) that uses a hash function.
2. [SP 800-38B] and [SP 800-38D][9] specify the CMAC and GMAC modes, respectively, for block ciphers (i.e., AES and TDEA).
3. [SP 800-185] defines the KMAC algorithm that is based on the SHA-3 XOFs specified in [FIPS 202].

The security strength that can be supported by a given MAC algorithm depends on the primitive algorithm used (e.g., the hash function or block cipher) and the strength of the cryptographic key.[10]

Table 15 provides the approval status and required key strengths for the MAC algorithms and the associated cryptographic primitives, as appropriate.

Table 14. Approval status of MAC algorithms

MAC Algorithm	Cryptographic Primitive	Key Strengths (in bits)	Status
HMAC Generation	SHA-1, SHA-2, SHA-3	< 112	Disallowed
	SHA-1, SHA-224, SHA-512/224, SHA3-224	≥ 112	Deprecated through 2030 Disallowed after 2030
	All other SHA-2 and SHA3 hash functions	112 ≤ *strength* < 128	Acceptable through 2030 Disallowed after 2030
		≥ 128	Acceptable
HMAC Verification	SHA-1, SHA-2, SHA-3	< 112	Legacy use
	SHA-1, SHA-224, SHA-512/224, SHA3-224	≥ 112	Acceptable through 2030 Legacy use after 2030
	All other SHA-2 and SHA3 hash functions	112 ≤ *strength* < 128	Acceptable through 2030 Legacy use after 2030
		≥ 128 bits	Acceptable
CMAC Generation	Two-key TDEA	< 112	Disallowed

[9] The CCM authenticated encryption mode specified in [SP 800-38C] also generates a MAC. However, the CCM mode cannot be used to generate a MAC without also performing encryption. The modes listed in this section are used only to generate a MAC.
[10] The strength of the key is less than or equal to its length.

MAC Algorithm	Cryptographic Primitive	Key Strengths (in bits)	Status
	Three-key TDEA	112	Disallowed
	AES	128, 192, 256	Acceptable
CMAC Verification	TDEA	≤ 112	Legacy use
	AES-128, AES-192 AES-256,	128, 192, 256	Acceptable
GMAC Generation and Verification	AES-128, AES-192, 256	128, 192, 256	Acceptable
KMAC Generation	KMAC	< 112	Disallowed
		112 ≤ *strength* < 128	Acceptable through 2030 Disallowed after 2030
		≥ 128	Acceptable
KMAC Verification	KMAC	< 112	Disallowed
		112 ≤ *strength* < 128	Acceptable through 2030 Legacy use after 2030
		≥ 128	Acceptable

HMAC:

- HMAC generation:
    - HMAC generation using keys providing less than 112 bits of security strength is **disallowed**, regardless of the hash function used as the cryptographic primitive.
    - HMAC generation using SHA-1 or the 224-bit hash functions (i.e., SHA-224, SHA-512/224, or SHA3-224) and keys providing ≥ 112 bits of security strength is **deprecated** through December 31, 2030, and **disallowed** thereafter.
    - HMAC generation using all other hash functions (i.e., SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, or SHA3-512):
        - When using keys providing at least 112 bits of security strength but less than 128 bits of security strength, the use of these hash functions is **acceptable** through December 31, 2030, for HMAC generation and **disallowed** thereafter.
        - When using keys providing at least 128 bits of security strength, the use of these hash functions is **acceptable** for HMAC generation.
- HMAC Verification:
    - HMAC verification using keys with less than 112 bits of security strength is allowed for **legacy use**.
    - HMAC verification using SHA-1 or the 224-bit hash functions (i.e., SHA-224, SHA-512/224, or SHA3-224) and keys providing ≥ 112 bits of security strength is **acceptable** through December 31, 2030, and allowed for **legacy use** thereafter.

- HMAC verification using all other hash functions (i.e., SHA-256, SHA-384, SHA-512, SHA-512/256, SHA3-256, SHA3-384, or SHA3-512):
  - When using keys providing at least 112 bits of security strength but less than 128 bits of security strength, the use of these hash functions is **acceptable** through December 31, 2030, for HMAC verification and allowed for **legacy use** thereafter.
  - When using keys providing at least 128 bits of security strength, the use of these hash functions is **acceptable** for HMAC verification.

CMAC:

- CMAC Generation:
  - The use of TDEA for CMAC generation is **disallowed**.
  - The use of AES-128, AES-192, or AES-256 for CMAC generation is **acceptable**.
- CMAC Verification:
  - The use of TDEA for CMAC verification is allowed for **legacy use**.
  - The use of AES for CMAC verification is **acceptable**.

GMAC Generation and Verification:

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

KMAC:

- KMAC generation:
  - The use of KMAC for MAC generation using keys with less than 112 bits of security strength is **disallowed**.
  - The use of KMAC for MAC generation using keys with security strengths of at least 112 bits but less than 128 bits is **acceptable** through December 31, 2030, and **disallowed** thereafter.
  - The use of KMAC for MAC generation using keys with at least 128 bits of security strength is **acceptable**.
- KMAC verification:
  - The use of KMAC for MAC verification using keys with less than 112 bits of security strength is **disallowed**.[11]
  - The use of KMAC for MAC verification using keys with security strengths of at least 112 bits but less than 128 bits is **acceptable** through December 31, 2030, and allowed for **legacy use** thereafter.

[11] KMAC was initially approved after a security strength of 112 bits was required.

- The use of KMAC for MAC verification using keys with security strengths of at least 128 bits is **acceptable.**

References

[FIPS140] National Institute of Standards and Technology (2019) Security Requirements for Cryptographic Modules. (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 140-3. https://doi.org/10.6028/NIST.FIPS.140-3

[FIPS140_IG] National Institute of Standards and Technology, Canadian Centre for Cyber Security Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program. Available at https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf

[FIPS180] National Institute of Standards and Technology (2015) Secure Hash Standard (SHS). (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 180-4. https://doi.org/10.6028/NIST.FIPS.180-4

[FIPS186-4] National Institute of Standards and Technology (2013) Digital Signature Standard (DSS). (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 186-4 [withdrawn]. https://doi.org/10.6028/NIST.FIPS.186-4

[FIPS186-5] National Institute of Standards and Technology (2023) Digital Signature Standard (DSS). (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 186-5. https://doi.org/10.6028/NIST.FIPS.186-5

[FIPS197] National Institute of Standards and Technology (2001) Advanced Encryption Standard (AES). (U.S. Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 197-upd1, updated May 9, 2023. https://doi.org/10.6028/NIST.FIPS.197-upd1

[FIPS202] National Institute of Standards and Technology (2015) SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. (U.S. Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 202. https://doi.org/10.6028/NIST.FIPS.202

[FIPS203] National Institute of Standards and Technology (2024) Module-Lattice-Based Key-Encapsulation Mechanism Standard. (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 203. https://doi.org/10.6028/NIST.FIPS.203

[FIPS204] National Institute of Standards and Technology (2024) Module-Lattice-Based Digital Signature Standard. (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 204. https://doi.org/10.6028/NIST.FIPS.204

[FIPS205] National Institute of Standards and Technology (2024) Stateless Hash-Based Digital Signature Standard. (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 205. https://doi.org/10.6028/NIST.FIPS.205

[Grover] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

[RFC3526] Kivinen T, Kojo M (2003) More Modular Exponential (MODP) Diffie-Hellman Groups for Internet Key Exchange (IKE), Internet Engineering Task Force (IETF) Request for Comments (RFC) 3526. https://doi.org/10.17487/RFC3526

[RFC4492] Blake-Wilson S, Bolyard N (2006) Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS), Internet Engineering Task Force (IETF) Request for Comments (RFC) 4492. https://doi.org/10.17487/RFC4492

[RFC5639] Lochter M (2010) Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, Internet Engineering Task Force (IETF) Request for Comments (RFC)5639. https://doi.org/10.17487/RFC5639

[RFC5903] Fu D, Salinas J (2010) Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2, Internet Engineering Task Force (IETF) Request for Comments (RFC) 5903. https://doi.org/10.17487/RFC5903

[RFC7919] Gillmor D (2016) Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS), Internet Engineering Task Force (IETF) Request for Comments (RFC) 7919. https://doi.org/10.17487/RFC7919

[RFC8017] Moriarty K (ed.), Kaliski B, Jonsson J, Rusch A (2016) PKCS #1: RSA Cryptography Specifications Version 2.2. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 8017. https://doi.org/10.17487/RFC8017

[SP80038A] Dworkin MJ (2001) Recommendation for Block Cipher Modes of Operation: Methods and Techniques. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38A. https://doi.org/10.6028/NIST.SP.800-38A

[SP80038A_addendum]
Dworkin MJ (2010) Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38A, Addendum. https://doi.org/10.6028/NIST.SP.800-38A-Add

[SP80038B] Dworkin MJ (2005) Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38B, Includes updates as of October 6, 2016. https://doi.org/10.6028/NIST.SP.800-38B

[SP80038C] Dworkin MJ (2004) Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38C, Includes updates as of July 20, 2007. https://doi.org/10.6028/NIST.SP.800-38C

[SP80038D] Dworkin MJ (2007) Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38D. https://doi.org/10.6028/NIST.SP.800-38D

[SP80038E] Dworkin MJ (2010) Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38E. https://doi.org/10.6028/NIST.SP.800-38E

[SP80038F] Dworkin MJ (2012) Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38F. https://doi.org/10.6028/NIST.SP.800-38F

[SP80038G] Dworkin MJ (2016) Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-38G. https://doi.org/10.6028/NIST.SP.800-38G

[SP80056Ar2] Barker EB, Chen L, Roginsky A, Smid M (2013) Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-56Ar2 [withdrawn]. https://doi.org/10.6028/NIST.SP.800-56Ar2

[SP80056Ar3] Barker EB, Chen L, Roginsky A, Vassilev A, Davis R (2018) Recommendation for Pair-Wise Key-Establishment Using Schemes Using Discrete Logarithm Cryptography. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-56Ar3. https://doi.org/10.6028/NIST.SP.800-56Ar3

[SP80056B] Barker E, Chen L, Regenscheid A, Smid M (2009) Recommendation for Pair-Wise Key-Establishment Schemes Integer Factorization Cryptography. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-56B [withdrawn]. https://doi.org/10.6028/NIST.SP.800-56B

[SP80056Br2] Barker EB, Chen L, Roginsky A, Vassilev A, Davis R, Simon S (2019) Recommendation for Pair-Wise Key-Establishment Schemes Integer Factorization Cryptography. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-56Br2. https://doi.org/10.6028/NIST.SP.800-56Br2

[SP80056C] Barker EB, Chen L, Davis R (2020) Recommendation for Key-Derivation Methods in Key-Establishment Schemes. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-56Cr2. https://doi.org/10.6028/NIST.SP.800-56Cr2

[SP80057] Barker EB (2020) Recommendation for Key Management: Part 1 – General. (National Institute of Standards and Technology, Gaithersburg, MD), NIST

Special Publication (SP) NIST SP 800-57pt1r5. https://doi.org/10.6028/NIST.SP.800-57pt1r5

[SP80067] Barker EB, Mouha N (2017) Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-67r2. https://doi.org/10.6028/NIST.SP.800-67r2

[SP80090A] Barker EB, Kelsey JM (2015) Recommendation for Random Number Generation Using Deterministic Random Bit Generators. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-90Ar1. https://doi.org/10.6028/NIST.SP.800-90Ar1

[SP80090B] Turan MS, Barker EB, Kelsey JM, McKay KA, Baish ML, Boyle VM (2018) Recommendation for the Entropy Sources Used for Random Bit Generation. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-90B. https://doi.org/10.6028/NIST.SP.800-90B

[SP80090C] Barker EB, Kelsey JM, McKay KA, Roginsky A, Turan MS (2024) Recommendation for Random Bit Generator (RBG) Constructions. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-90C 4pd. https://doi.org/10.6028/NIST.SP.800-90C.4pd

[SP800108] Chen L (2022) Recommendation for Key Derivation Using Pseudorandom Functions (Revised). (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-108r1-upd1, Includes updates as of February 2, 2024. https://doi.org/10.6028/NIST.SP.800-108r1-upd1

[SP800131Ar2] Barker E, Roginsky A (2019) Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-131Ar2. http://dx.doi.org/10.6028/NIST.SP.800-131Ar2

[SP800132] Sönmez Turan M, Barker EB, Burr WE, Chen L (2010) Recommendation for Password-Based Key Derivation: Part 1: Storage Applications. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-132. https://doi.org/10.6028/NIST.SP.800-132

[SP800133] Barker E, Roginsky A, Davis R (2020) Recommendation for Key Generation. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-133r2. https://doi.org/10.6028/NIST.SP.800-133r2

[SP800135] Dang Q (2011) Recommendation for Existing Application-Specific Key Derivation Functions. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-135r1. https://doi.org/10.6028/NIST.SP.800-135r1

[SP800185] Kelsey JM, Chang S-jH, Perlner RA (2016) SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash. (National Institute of Standards and

Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-185. https://doi.org/10.6028/NIST.SP.800-185

[SP800186] Chen L, Moody D, Regenscheid A, Robinson A, Randall K (2023) Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-186. https://doi.org/10.6028/NIST.SP.800-186

[SP800208] Cooper DA, Apon DC, Dang QH, Davidson MS, Dworkin MJ, Miller CA (2020) Recommendation for Stateful Hash-Based Signature Schemes, (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-208. https://doi.org/10.6028/NIST.SP.800-208

[SP800224] Sönmez Turan M, Brandão LTAN (2024) Keyed-Hash Message Authentication Code (HMAC): Specification of HMAC and Recommendations for Message Authentication (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-224 ipd. https://doi.org/10.6028/NIST.SP.800-224.ipd

[NISTIR8413] Alagic G, Apon D, Cooper D, Dang Q, Dang T, Kelsey J, Lichtinger J, Liu Y, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D (2022) Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) NIST IR 8413-upd1, Includes updates as of September 26, 2022. https://doi.org/10.6028/NIST.IR.8413-upd1

[NISTIR8459] Mouha N, Dworkin M (2024) Report on the Block Cipher Modes of Operation in the NIST SP 800-38 Series. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) NIST IR 8459 ipd. https://doi.org/10.6028/NIST.IR.8459

[X931] American National Standard (ANS) X9.31-1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA). Withdrawn.

[Zalka] C. Zalka, Grover's quantum searching algorithm is optimal, Phys. Rev. A 60, 2746 (1999).

Appendix A. Continued Use of AES

Grover's algorithm [Grover] allows a quantum computer to perform a brute-force key search using (approximately) the square root of the number of steps that would be required using a classical computer. This suggests that an attacker with access to a quantum computer might be able to attack a symmetric cipher with a key up to twice as long as could be attacked by an attacker with access to only classical computers. However, quantum computing hardware will likely be more expensive to build and use than classical hardware, and Grover's algorithm might not speed up brute-force key search as dramatically as suspected.

In 1997, Zalka proved that in order to obtain the full quadratic speedup, all of the steps of Grover's algorithm must be performed in series [Zalka]. The advantage of Grover's algorithm will be smaller in the real world, where attacks on cryptography use massively parallel processing. Taking this into account, it is quite likely that Grover's algorithm will provide less than the expected advantage in attacking AES. Furthermore, even if quantum computers become much less expensive than anticipated, the known difficulty of parallelizing Grover's algorithm suggests that AES will still be safe for a very long time. This, of course, assumes that no new cryptographic weaknesses with respect to classical or quantum cryptanalysis are found in AES.

Based on this understanding, current applications can continue to use AES with key sizes of 128, 192, or 256 bits. When NIST foresees the need for a transition of symmetric key algorithms, hash functions, key-establishment methods, or digital signature schemes to protect against threats from quantum computers, NIST and the CMVP will issue guidance regarding such transitions.

Appendix B. Change History

Revision 3 includes the following changes to SP 800-131A:

1. The document has been reformatted using a revised template for NIST Special Publications (SPs).
2. Section 1.2.2 expands the discussion on **legacy use** to include how users should consider the legacy-use status.
3. Section 1.2.3 is a new section that discusses the strategy for transitioning from a 112-bit security strength to a 128-bit security strength for block ciphers and hash functions or continuing the acceptability of the 112-bit security strength until further PQC guidance is provided for digital signatures and key establishment.
4. In Sec. 2, the Skipjack algorithm has been removed, TDEA is disallowed for applying cryptographic protection, and a subsection on the block cipher modes of operation has been added.
5. In Sec. 3, EdDSA (specified in [FIPS 186]), the new quantum-resistant digital signature algorithms (specified in [FIPS 204] and [FIPS 205]), and the stateful hash-based signature algorithms (specified in [SP 800-208]) have been added. DSA and RSA (as specified in [X9.31]) are now disallowed for generating digital signatures.
6. Sections 4, 8, and 12 have been added to discuss key generation [SP 800-133], key encapsulation mechanisms [FIPS 203], and XOFs [FIPS 202].
7. Section 5 has been augmented to include entropy sources [SP 800-90B] and RBG constructions [SP 800-90C]. The discussion of Dual_EC_DRBG RNGs has been removed.
8. Sections 6 and 7 have been updated to show that DH, MQV, and RSA schemes that do not comply with [SP 800-56A] or [SP 800-56B] are only allowed for **legacy use**.
9. Section 9 has added the one- and two-step key-derivation methods specified in [SP 800-56C] and key derivation [SP 800-108]. HMAC using SHA-1 and the 224-bit hash functions has been deprecated.
10. In Sec. 10, key wrapping using TDEA is **disallowed**, and CCM and GCM have been added. The combination of an **approved** encryption mode and an **approved** authentication method for key wrapping has been deprecated.
11. In Sec. 11, the use of SHA-1 and the 224-bit hash functions has been deprecated.
12. In Sec. 13, the use of SHA-1 and the 224-bit hash functions for generating a MAC has been deprecated.
13. The References section has been updated.
14. Appendix A has been added to discuss the continued use of AES when quantum computers become available.
15. Appendix C includes a list of the acronyms used in this document.

16. Appendix D provides a glossary of terms.

Appendix C. List of Symbols, Abbreviations, and Acronyms

AES
Advanced Encryption Standard

CAVP
Cryptographic Algorithm Validation Program

CMVP
Cryptographic Module Validation Program

DRBG
Deterministic Random Bit Generator

FIPS
Federal Information Processing Standards

ITL
Information Technology Laboratory

MAC
Message Authentication Code

MQV
Menezes-Qu-Vanstone (algorithm)

NIST
National Institute of Standards and Technology

RBG
Random Bit Generator

SP
(NIST) Special Publication

TDEA
Triple Data Encryption Algorithm

Appendix D. Glossary

acceptable
The algorithm and key length/strength in a FIPS or SP is approved for use in accordance with any associated guidance.

apply cryptographic protection
Depending on the algorithm, to encrypt or sign data, generate a hash function or message authentication code (MAC), or establish keys, including wrapping and deriving keys.

approval status
Used to designate usage by the U.S. Federal Government.

approved
FIPS-**approved** or NIST-recommended. An algorithm or technique that is either 1) specified in a FIPS or NIST recommendation or 2) adopted in a FIPS or NIST recommendation and specified in (a) an appendix to the FIPS or NIST recommendation or (b) a document referenced by a FIPS or NIST recommendation.

deprecated
The algorithm and key length may be used, but there is some security risk.

disallowed
The algorithm or key length is no longer allowed for applying cryptographic protection.

entropy
A measure of disorder, randomness, or variability in a closed system.

hash(ing) method
A hash function or extendable-output function.

legacy use
The algorithm or key length may only be used to process already protected information (e.g., decrypt ciphertext data or verify a digital signature).

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

security strength
A number associated with the amount of work (i.e., the number of operations) that is required to break a cryptographic algorithm or system. If 2^N execution operations of the algorithm (or system) are required to break the cryptographic algorithm, then the security strength is N bits. As used herein, security strength is a measure of the difficulty of subverting cryptographic protection (e.g., discovering the key) using classical computers.

shall
A requirement for Federal Government use. **Shall** may be coupled with **not** to become **shall not**.

XOR(ing)
Bit-wise exclusive-or. A mathematical operation that is defined as $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, and $1 \oplus 1 = 0$.