



FIDO Authenticator Allowed Cryptography List

FIDO Alliance Final Requirements Document 02 November 2021

This version:

<https://fidoalliance.org/specs/fido-security-requirements/fido-authenticator-allowed-cryptography-list-v1.4-fd-20211102.html>

Previous version:

<https://fidoalliance.org/specs/fido-security-requirements/fido-authenticator-allowed-cryptography-list-v1.3-fd-20201102.html>

Editor:

[Dr. Beatrice Peirani](#), [Thales](#)

Contributors:

[Dr. Joshua E. Hill](#), [InfoGard Laboratories](#)
[Douglas Biggs](#), [InfoGard Laboratories](#)
[Rolf Lindemann](#), [Nok Nok Labs, Inc.](#)

Copyright © 2013-2021 [FIDO Alliance](#) All Rights Reserved.

Abstract

This document helps support the FIDO Authenticator Security Certification program. This list does not in any way alter the protocol specifications provided in other FIDO Authenticator documents, so the presence or absence of an algorithm in this list does not suggest that this algorithm is or is not allowed within any FIDO protocol. For certified FIDO Authenticators, there are various requirements that limit “internal” algorithms, those that are not explicitly specified within the FIDO Authenticator protocol. Additionally, the procedure for determining the “Overall Authenticator Claimed Security Strength” involves locating the security level for each algorithm used by the FIDO Authenticator within this document; this procedure applies to all cryptographic algorithms used by the FIDO Authenticator.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The most recent version of this document can be found on the [FIDO Alliance Website](https://fidoalliance.org) at <https://fidoalliance.org>.

This document was published by the [FIDO Alliance](https://fidoalliance.org) as a Final Requirements Document. If you wish to make comments regarding this document, please [Contact Us](#). All comments are welcome.

No rights are granted to prepare derivative works of this document. Entities seeking permission to reproduce portions of this document for other uses must contact the FIDO Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this Requirements Document may require licenses under third party intellectual property rights, including without limitation, patent rights. The FIDO Alliance, Inc. and its Members and any other contributors to the Requirements Document are not, and shall not be held, responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

THIS FIDO ALLIANCE REQUIREMENTS DOCUMENT IS PROVIDED “AS IS” AND WITHOUT ANY WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTY OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

- 1. [Notation](#)
 - 1.1 [Version](#)
- 2. [Requirements for Additional Candidates](#)
- 3. [Allowed Cryptographic Functions](#)
 - 3.1 [Post-quantum cryptography](#)
 - 3.2 [Confidentiality Algorithms](#)
 - 3.3 [Hashing Algorithms](#)
 - 3.4 [Data Authentication Algorithms](#)
 - 3.5 [Key Protection Algorithms](#)
 - 3.6 [Random Number Generator](#)
 - 3.6.1 [Physical/True \(TRNG\)/Non-Deterministic Random Number/Bit Generator\(NRBG\) Requirements](#)
 - 3.6.2 [Deterministic Random Number \(DRNG\)/Bit Generator \(DRBG\) Requirements](#)
 - 3.7 [Key Agreement Algorithms](#)
 - 3.8 [Key Derivation Functions \(KDFs\)](#)
 - 3.9 [Signature Algorithms](#)
 - 3.10 [Anonymous Attestation Algorithms](#)
- A. [References](#)
 - A.1 [Normative references](#)
 - A.2 [Informative references](#)

1. Notation

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

1.1 Version

This document specifies version 1.3.0 of the allowed cryptography (CV).

2. Requirements for Additional Candidates

If a vendor wants to use a cryptographic security function for an internal use that requires an Allowed algorithm, or to claim a non-zero security strength, then the vendor / lab shall provide a written argument that it:

- Additional candidates for algorithms shall at least support a security strength of 112 bits.
- Is not a proprietary solution,
- Fulfills the required security attributes (e.g., if the use requires confidentiality and data authentication, the primitive provides this),
- Has a security strength that can be readily characterized,
- Is accepted or recommended by at least one major international standardization group (e.g., ISO, IETF), or one national or European organizations (e.g. NIST [SP800-131Ar2], ANSI, SOGIS [SOGISCrypto]) and
- Has undergone extensive public review.

3. Allowed Cryptographic Functions

The stated security level identifies the expected number of computations that a storage-constrained attacker (who has access to no more than 2^{80} bytes of storage) shall expend in order to compromise the security of the cryptographic security function, under the currently best known attack that can be conducted under this storage constraint. This has been extracted from the currently best known relevant attacks against each cryptographic primitive, and is expected to shift over time as attacks improve.

At the time the document is published, there are not yet any standardized (NIST Project <https://csrc.nist.gov/projects/post-quantum-cryptography> or ISO/IEC JTC 1/SC 27/WG 2 SD8 on Post-Quantum Cryptography) quantum-safe cryptographic algorithms for asymmetric algorithms (e.g. signature, key protection based on RSA, and anonymous attestation). It is also not yet clear if the key size should (or should not) be increased for symmetric algorithms.

If the security level stated is n (where the security level is here defined with a classical computing power, and does not take into account quantum cryptanalysis), then the expected number of computations is less than the expected number of computations required to guess an $(n+1)$ -bit random binary string, and not less than the number of computations required to guess an n bit random binary string (i.e., on average, the number of computations required is less than 2^n computations and greater than or equal to $2^{(n-1)}$ computations).

3.1 Post-quantum cryptography

Quantum computers are expected to solve problems faster than conventional computing can do. To what extent quantum computers will shorten the time needed to solve some difficult mathematical problems is a matter of

controversy. In theory a large scale, stable and fault-tolerant quantum computer leveraging Shor’s algorithm could break asymmetric cryptography based on either RSA or Elliptic Curve technology. This means that the factorization of large composite numbers (on which RSA security is based) or the computation of discrete logarithm (on which DSA and ECDSA are based) would become feasible with the use of a quantum computer whatever the sizes of the keys.

When such mature quantum computers will be available for cryptanalysis purposes is another unknown. Yet because of the serious consequences of this threat becoming a reality, the NIST has decided to start developing standards for asymmetric quantum safe cryptography (<https://csrc.nist.gov/projects/post-quantum-cryptography>). A call for contributions has started the process for the selection of quantum safe algorithms eligible for standardization back in 2016.

Discussions on standardization of quantum safe cryptographic primitives are also ongoing at ETSI CYBER and ISO/IEC JTC 1/SC 27.

Stateful Hash-based signatures are the first family of quantum safe cryptographic mechanisms standardized (NIST SP 800-208 and ISO 14888-4).

NIST has recently published the list of selected post-quantum algorithms for asymmetric cryptography after the conclusion of the Round 3 of the screening process (<https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions>). This list includes 3 finalists (Crystals-Dilithium, Falcon, Rainbow) and 3 alternates (GeMSS, Picnic, Sphincs+) candidates for a signature algorithm, and 4 finalists (Kyber, NTRU, SABER, Classic McEliece) and 5 alternates (Bike, FrodoKEM, HQC, NTRUprime, SIKE) candidates for a Key Encryption Mechanism/Encryption algorithm.

NIST expects to select at most one between Kyber, NTRU and Saber for KEM, and one among Dilithium and Falcon (all based on structured lattices).

The final standard will be released, as draft for public comment in 2022-2023, and finalized by 2024.

NIST has not planned to standardize key agreement mechanisms to replace DH/ECDH. NIST explained the rationale in its FAQ: “*NIST believes that in its most widely used applications, such as those requiring forward secrecy, Diffie-Hellman can be replaced by any secure KEM with an efficient key generation algorithm*”.

Another solution is to use a hybrid approach, mixing a “classical” algorithm with a quantum safe one. In this case, keys derived by a hybrid key-establishment scheme remain secure if at least one of the underlying schemes is secure. NIST plans to incorporate a hybrid key establishment construction in a future revision of NIST SP 800-56C.

Dealing with symmetric cryptography, a conservative approach regarding post-quantum symmetric cryptography is to double the key size (i.e. migrating from AES-128 to AES-256) and increase the digest size (i.e. migrating from SHA-256 to SHA-384). But the Grover algorithm (which could theoretically be used to weaken the security of block ciphers and hash functions) will provide little or no advantage for attacking symmetric cryptography or hash functions.

This is why AES-128, AES-192 and SHA-256 are still recommended in this version of the document.

3.2 Confidentiality Algorithms

NOTE

Provide confidentiality, up to the stated security level.

Algorithm	Specified in	Security Level (bits)
-----------	--------------	-----------------------

AES-128	[FIPS197], [ISOIEC-18033-3]	128
AES-192	[FIPS197], [ISOIEC-18033-3]	192
AES-256	[FIPS197], [ISOIEC-18033-3]	256

3.3 Hashing Algorithms

NOTE

Provide pre-image resistance, 2nd pre-image resistance, and collision resistance.

Algorithm	Specified in	Security Level (bits)
SHA-256	[FIPS180-4], [ISOIEC-10118-3]	128
SHA-384	[FIPS180-4], [ISOIEC-10118-3]	192
SHA-512	[FIPS180-4], [ISOIEC-10118-3]	256
SHA-512/t, $256 \leq t < 512$	[FIPS180-4], [ISOIEC-10118-3]	t/2
SHA3-256	[FIPS202], [ISOIEC-10118-3]	128
SHA3-384	[FIPS202], [ISOIEC-10118-3]	192
SHA3-512	[FIPS202], [ISOIEC-10118-3]	256

3.4 Data Authentication Algorithms

NOTE

Provide data authentication.

NOTE

It is not uncommon to truncate the result of a MAC, but to be resistant, the final length should be at least 96 bits (see SOGIS Note 14-MACTruncation96 in [SOGISCrypto]). The Security Level cannot exceed the final length.

Algorithm	Specified in	Security Level (bits)
HMAC	[FIPS198-	Minimum of the length of the output of the hash used [2] , one-half of the number of

	1]	bits in the hash state ^[3] , or the number of bits in the HMAC key.
CMAC	[SP800-38B]	Equal to the minimum of the security strength of the underlying cipher and the length of the output MAC.
GMAC	[SP800-38D]	Equal to the minimum of the security strength of the underlying cipher and the length of the output MAC.

[2]Both due to the obvious guessing attack, and covers the case where the supplied key is hashed for the HMAC.

[3]Based on a birthday attack; a collision of the final state can lead to an existential forgery of longer messages with the same prefix.

NOTE

The underlying cryptographic algorithms used in HMAC, CMAC or GMAC need to be 112-bit security strength at least and part of the list of FIDO allowed cryptographic algorithms.

NOTE

SOGIS gives some additional recommendations for the use of GMAC, including IV (length must be equal to 96 bits) or MAC length (must be one of the values 96, 104, 112, 120, and 128 bits). See [SOGISCrypto] for the complete list of recommendations.

3.5 Key Protection Algorithms

NOTE

Provide confidentiality and data authentication.

Algorithm	Specified in	Security Level (bits)
Key Wrapping	[SP800-38F]	Equal to the security strength of the underlying cipher.
GCM Mode, with length 96 bit or larger IVs. For any given key, the IV length must be fixed.	[SP800-38D]	Equal to the security strength of the underlying cipher.
RSA OAEP	[RFC3447]. Key generation must be according to [FIPS186-4].	Depends on the parameter size: according to NIST, 112 bits for RSA 2048 and 128 bits for RSA 3072
CCM Mode	[SP800-38C]	Equal to the security strength of the underlying cipher.

Encrypt-then-HMAC [4]	Encryption specification depends on the cipher selected. HMAC specification [FIPS198-1]	The minimum of the security strength of the cipher and the HMAC.
Encrypt-then-CMAC [5]	Encryption specification depends on the cipher selected. CMAC specification [SP800-38B]	The minimum of the security strength of the cipher and the CMAC.

[4]The cipher and HMAC shall use independent keys, and the information HMACed shall include any IV / Nonce / Counter (if sent/stored), and, if the message size varies, the length of the message; when present, this message length shall reside prior to any variable length message components.

[5]The cipher and CMAC shall use independent keys, and the information CMACed shall include any IV / Nonce / Counter (if sent/stored).

NOTE

The underlying cryptographic algorithms used in the above table need to be 112-bit security strength at least and part of the list of FIDO allowed cryptographic algorithms.

3.6 Random Number Generator

In FIDO an allowed random number generator shall meet the requirements of one of the following sub sections.

Evidence that the requirements are met could be given by providing a proof that the implementation uses the underlying platform certified RNG/RBG through Common Criteria, FIPS 140-3 (issued on March 22nd 2019 or after) or an equivalent evaluation scheme against the listed standards, or by having a FIDO approved lab conducting an evaluation of the RNG/RBG implementation against the standards listed below. In other words, the following standards define the metrics required to assess the quality of the RNG implementation.

NOTE

FIPS 140-3 became effective on September 22, 2019.

- FIPS 140-3 testing, through the Cryptographic Module Validation Program (CMVP), will begin September 22, 2020; and
- FIPS 140-2 testing will continue for at least a year after FIPS 140-3 testing begins, i.e. testing of cryptographic modules against FIPS 140-2 will end on September 22, 2021.

NOTE

If the designer is interested in retaining the security of an (EC)DSA private key in the event of an entropy source failure or Deterministic Random Number Generator state compromise, then RFC6979-like properties can be obtained by providing the hash of the message being signed and the private key in use to the Deterministic Random Number Generator in a secure fashion (e.g., via the SP800-90A additional input parameter). Additional parameters (e.g., the KeyID / Key Handle, if it was randomly generated) may also be

used to increase resistance to attack in certain scenarios.

NOTE

In the last version of FIPS 140-2, IG7.18 gives a clear rule on the seeding sources for allowed DRBGs: Entropy Estimation and Compliance with [SP800-90b] is mandatory.

3.6.1 Physical/True (TRNG)/Non-Deterministic Random Number/Bit Generator(NRBG) Requirements

The (physical) random number generator shall meet the requirements specified in:

1. AIS 20/31 PTG.2 or PTG.3 or in

NOTE

If PTG.2 is used, an application-specific post processing may additionally be required to prevent any bias in the output function.

For instance, these requirements are met if a certified hardware platform is used (e.g. according to Global Platform TEE Protection Profile or Eurosmart Security IC Platform Protection Profile) and the Security Target contains Extended Component FCS_RNG.1 including at least one of the allowed classes PTG.2, or PTG.3.

2. NIST SP800-90C NRBG [SP800-90C] or in

Algorithm	Specified in	Security Level (bits)
Source RBG is DRBG with access to Live Entropy Source or it is an NRBG.	[SP800-90C], section 6	Any security strength.

3. NIST FIPS 140-2 [FIPS140-2] validation (issued on August 7th 2015 or after), with Entropy Source Health Tests. The related security level is as defined in the module's security policy.

We consider this a physical RNG if at least as much entropy is added into the RNG as is retrieved per request.

NOTE

It is uncommon for the DRBGs in FIPS modules to meet these requirements, unless their design anticipates one of the SP800-90C NRBG designs.

3.6.2 Deterministic Random Number (DRNG)/Bit Generator (DRBG) Requirements

NOTE

Provide computational indistinguishability from an ideal random sequence, cycle resistance, non-destructive reseeding, insensitivity of a seeded generator to seed source failure or compromise, backtracking resistance. Ideally, the ability to provide additional input, and ability to recover from a compromised internal state.

The (deterministic) random number generator shall meet the requirements specified in:

1. AIS 20/31 DRG.3 or DRG.4 (having an entropy of the seed of at least N bits, where N is the targeted security level) or in
2. NIST SP800-90A DRBG [[SP800-90ar1](#)],

Algorithm	Specified in	Security Level (bits)
HMAC_DRBG	[SP800-90ar1], Revision 1, section 10.1.2	The instantiated security level, as defined in [SP800-90ar1].
CTR_DRBG	[SP800-90ar1], Revision 1, section 10.2.1	The instantiated security level, as defined in [SP800-90ar1].
HASH_DRBG	[SP800-90ar1], Revision 1, section 10.1.1	The instantiated security level, as defined in [SP800-90ar1].

3. or in NIST FIPS 140-3 [[FIPS140-3](#)] validation (issued on March 22nd 2019 or after).

NOTE

We consider this a deterministic RNG if less entropy is added into the RNG than is retrieved.

NOTE

The last version of FIPS 140-2 [[FIPS140-2](#)] IG 7.18 requires transition to full compliance with [[SP800-90b](#)] for seed for DRBG.

3.7 Key Agreement Algorithms

Allowing two or more parties to generate a shared secret. It is generally followed by a key derivation function (KDF), as described in next section, to generate one or several keys from the shared secret.

Algorithm	Specified in	Security Level (bits)
Diffie-Hellmann (DH) 2048-bit key	[SP800-56Ar3], [ISOIEC-11770-3]	≥ 112
EC-DH on P-256 or BrainpoolP256r1	[SP800-56Ar3], [ISOIEC-11770-3]	≥ 128

3.8 Key Derivation Functions (KDFs)

Deriving keys.

Algorithm	Specified in	Security Level (bits)
KDF in counter mode	[SP800-108]	min(Bit length of key derivation key K_i used as input, Security level of PRF)
KDF in feedback mode	[SP800-108]	min(Bit length of key derivation key K_i used as input, Security level of PRF)
KDF in double pipeline iteration mode	[SP800-108]	min(Bit length of key derivation key K_i used as input, Security level of PRF)
HKDF	[SP800-56cr1], [RFC5869]	min(Bit length of key derivation key K_i used as input, Security level of HMAC)

Where PRF denotes an acceptable pseudorandom function (based on HMAC or CMAC), and key-derivation key denotes an acceptable pre-shared cryptographic key (with key length at least 112 bits), as defined in [SP800-108].

NOTE

The underlying cryptographic algorithms used in the above table need to be 112-bit security strength at least and part of the list of FIDO allowed cryptographic algorithms.

NOTE

The KDF used in CTAP for PIN protocol version 1 is slightly different from [SP800-56cr1], but is acceptable from a security point of view.

3.9 Signature Algorithms

NOTE

Provide data authentication, and non-repudiation.

Algorithm	Specified in	Security Level (bits)
ECDSA on NIST P-256	[ECDSA-ANSI], [FIPS186-4], [ISOIEC-14888-3]	128
ECDSA on NIST P-384	[ECDSA-ANSI], [FIPS186-4], [ISOIEC-14888-3]	192

ECDSA on NIST P-521	[ECDSA-ANSI], [FIPS186-4], [ISOIEC-14888-3]	256
ECDSA on BrainpoolP256r1	[RFC5639]	128
ECDSA on secp256k1	[ECDSA-ANSI], [FIPS186-4], [!Certicom SEC 2]	126 [7]
2048-bit RSA PSS	[FIPS186-4],[ISOIEC-9796-2]	112
1024*n-bit RSA PKCS v1.5 (n=2,3,4)	[FIPS186-4],[ISOIEC-9796-2]	112
1024*n-bit RSA PKCS v2.1 (PSS) (n=2,3,4)	[FIPS186-4],[ISOIEC-9796-2]	112
SM2 digital signatures (SM2 part 2) using the SM3 hash on the SM2 curve specified by OSCCA .	SM2 [ISOIEC-14888-3] and SM3 [ISOIEC-10118-3]	128
Ed25519	EDDSA [RFC8032]	128 [8]
Ed448	EDDSA [RFC8032]	224 [8]

[\[7\]](#) Based on an attack using Pollard rho on the equivalence classes defined by the curve's easily computable endomorphism.

[\[8\]](#) Based on the difficulty of performing discrete logs on the group defined by the recommended curve parameters.

NOTE

NIST [[SP800-131Ar2](#)] disallows the use of PKCS#1 version 1.5 after December 31, 2023. SOGIS [[SOGISCrypto](#)] only allow it for legacy reasons (until end 2022).

3.10 Anonymous Attestation Algorithms

NOTE

Provide anonymous attestation.

The security strength in this section is the minimum of three values:

1. The security strength of the underlying hash.
2. The difficulty of conducting a discrete log within the Elliptic Curve.
3. The difficulty of conducting a discrete log within a finite field in which the Elliptic Curve can be embedded (we'll refer to this field as the embedding field).

In most cases, the limiting factor was the difficulty of performing the discrete log calculation within the embedding field.

The security level values here were taken from NIST guidance. This NIST guidance is based on conducting the discrete log calculation within prime ordered fields; the structure of the fields here is richer, and this structure could possibly allow for a more advanced discrete log approach that could be considerably faster. Currently, the best known algorithms in both cases have the same asymptotic complexity ($L_q[1/3]$), but without extensive testing, it isn't clear how the number of computations compares.

In addition, the NIST guidance does not allow for security levels other than a few specific proscribed values: if the number of bits required to represent the order of the embedding field is between 3072 and 7679, the security level is reported as 128 bits. Similarly, if the number of bits required to represent the order of the embedding field is between 2048 and 3071, the security strength is reported as 112 bits.

Algorithm	Specified in	Security Level (bits)
ED256-2	[FIDOEcdaaAlgorithm], section Object Formats and Algorithm Details, [DevScoDah2007]	112
ED512	[FIDOEcdaaAlgorithm], section Object Formats and Algorithm Details, [ISO15946-5]	128
ED638	[FIDOEcdaaAlgorithm], section Object Formats and Algorithm Details, [TPMv2-Part4]	128

A. References

A.1 Normative references

[[DevScoDah2007](#)]

Augusto Jun Devegili; Michael Scott; Ricardo Dahab. [Implementing Cryptographic Pairings over Barreto-Naehrig Curves](#). 2007. URL: <https://eprint.iacr.org/2007/390.pdf>

[[ECDSA-ANSI](#)]

. [Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography ANSI X9.63-2011 \(R2017\)](#). 2017. URL: [https://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+X9.63-2011+\(R2017\)](https://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+X9.63-2011+(R2017))

[[FIDOEcdaaAlgorithm](#)]

R. Lindemann; J. Camenisch; M. Drijvers; A. Edgington; A. Lehmann; R. Urian. [FIDO ECDA A Algorithm](#). 25 May 2021. Review Draft. URL: <https://fidoalliance.org/specs/common-specs/fido-ecdaa-algorithm-v2.1-rd-20210525.html>

[[FIPS140-2](#)]

. [FIPS PUB 140-2: Security Requirements for Cryptographic Modules](#). May 2001. URL: <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>

[[FIPS140-3](#)]

. [FIPS PUB 140-3: Security Requirements for Cryptographic Modules](#). March 2019. URL: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf>

[[FIPS180-4](#)]

. [FIPS PUB 180-4: Secure Hash Standard \(SHS\)](#). August 2015. URL:

<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>

[FIPS186-4]

. [FIPS PUB 186-4: Digital Signature Standard \(DSS\)](http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf). July 2013. URL: <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>

[FIPS197]

. [FIPS PUB 197: Specification for the Advanced Encryption Standard \(AES\)](http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf). November 2001. URL: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

[FIPS198-1]

. [FIPS PUB 198-1: The Keyed-Hash Message Authentication Code \(HMAC\)](http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf). July 2008. URL: http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

[FIPS202]

. [FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions](http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf). August 2015. URL: <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>

[ISO15946-5]

. [ISO/IEC 15946-5 Information Technology - Security Techniques - Cryptographic techniques based on elliptic curves - Part 5: Elliptic curve generation](https://webstore.iec.ch/publication/61253). URL: <https://webstore.iec.ch/publication/61253>

[ISOIEC-10118-3]

. [SM3 Cryptographic Hash Algorithm](https://www.iso.org/standard/67116.html). October 2018. URL: <https://www.iso.org/standard/67116.html>

[ISOIEC-11770-3]

. [ISO/IEC 11770-3: Information Technology – Security Techniques - Key Management - Part 3: Mechanisms using asymmetric techniques](https://www.iso.org/standard/60237.html). 2015-08. URL: <https://www.iso.org/standard/60237.html>

[ISOIEC-14888-3]

. [SM2: Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves: Part 1: General](https://www.iso.org/standard/76382.html). November 2018. URL: <https://www.iso.org/standard/76382.html>

[ISOIEC-18033-3]

. [ISO/IEC 18033-3 Information Technology - Security Techniques - Encryption algorithms – Part 3: Block ciphers](https://www.iso.org/standard/54531.html). 2010-12. URL: <https://www.iso.org/standard/54531.html>

[ISOIEC-9796-2]

. [ISO/IEC 9796-2: Information Technology – Security Techniques - Digital signature schemes giving message recovery — Part 2: Integer factorization based mechanisms](https://www.iso.org/standard/54788.html). 2010-12. URL: <https://www.iso.org/standard/54788.html>

[RFC3447]

J. Jonsson; B. Kaliski. [Public-Key Cryptography Standards \(PKCS\) #1: RSA Cryptography Specifications Version 2.1](https://tools.ietf.org/html/rfc3447). February 2003. obsoleted by RFC 8017. URL: <https://tools.ietf.org/html/rfc3447>

[RFC5639]

M. Lochter; J. Merkle. [Elliptic Curve Cryptography \(ECC\) Brainpool Standard](https://tools.ietf.org/html/rfc5639). March 2010. URL: <https://tools.ietf.org/html/rfc5639>

[RFC5869]

H. Krawczyk; P. Eronen. [HMAC-based Extract-and-Expand Key Derivation Function \(HKDF\)](https://www.rfc-editor.org/rfc/rfc5869). May 2010. Informational. URL: <https://www.rfc-editor.org/rfc/rfc5869>

[RFC8032]

S. Josefsson; I. Liusvaara. [Edwards-Curve Digital Signature Algorithm \(EdDSA\)](https://www.rfc-editor.org/rfc/rfc8032). January 2017. Informational. URL: <https://www.rfc-editor.org/rfc/rfc8032>

[SOGISCrypto]

SOG-IS Crypto Working Group. [SOG-IS crypto evaluation scheme agreed cryptographic mechanisms](https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf). January 2020. URL: <https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf>

[SP800-108]

Lily Chen. [*NIST Special Publication 800-107: Recommendation for Key Derivation Using Pseudorandom Functions*](#). October 2009. URL: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-108.pdf>

[SP800-131Ar2]

E. Barker; A. Roginsky. [*NIST Special Publication 800-131A: Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*](#). March 2019. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf>

[SP800-38B]

M. Dworkin. [*NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*](#). June 2016. URL: <https://csrc.nist.gov/publications/detail/sp/800-38b/final>

[SP800-38C]

M. Dworkin. [*NIST Special Publication 800-38C: Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*](#). July 2007. URL: http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf

[SP800-38D]

M. Dworkin. [*NIST Special Publication 800-38C: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode \(GCM\) and GMAC*](#). November 2007. URL: <https://csrc.nist.gov/publications/detail/sp/800-38d/final>

[SP800-38F]

M. Dworkin. [*NIST Special Publication 800-38F: Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*](#). December 2012. URL: <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf>

[SP800-56Ar3]

Elaine Barker; Lily Chen; Allen Roginsky; Miles Smid. [*Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*](#). April 2018. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf>

[SP800-56cr1]

Elaine Barker; Lily Chen; Rich Davis. [*NIST Special Publication 800-56C revision 1: Recommendation for Key Derivation Methods in Key Establishment Schemes*](#). April 2018. URL: <https://doi.org/10.6028/NIST.SP.800-56Cr1>

[SP800-90C]

Elaine Barker; John Kelsey. [*NIST Special Publication 800-90C: Recommendation for Random Bit Generator \(RBG\) Constructions*](#). April 2016. Draft. URL: http://csrc.nist.gov/publications/drafts/800-90/sp800_90c_second_draft.pdf

[SP800-90ar1]

Elaine Barker; John Kelsey. [*NIST Special Publication 800-90A Rev. 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators*](#). June 2015. URL: <http://dx.doi.org/10.6028/NIST.SP.800-90Ar1>

[SP800-90b]

Meltem Sönmez Turan; Elaine Barker; John Kelsey; Kerry McKay; Mary Baish; Michael Boyle. [*NIST Special Publication 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation*](#). January 2018. URL: <https://csrc.nist.gov/publications/detail/sp/800-90b/final>

[TPMv2-Part4]

[*Trusted Platform Module Library, Part 4: Supporting Routines*](#). 29 September 2016. URL: <https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-4-Supporting-Routines-01.38.pdf>

A.2 Informative references

[RFC2119]

S. Bradner. [*Key words for use in RFCs to Indicate Requirement Levels*](https://tools.ietf.org/html/rfc2119). March 1997. Best Current Practice. URL: <https://tools.ietf.org/html/rfc2119>