# FIDO Authenticator Allowed Cryptography List

## FIDO Alliance 05 October 2017

**This version:**
https://fidoalliance.org/specs/fido-uaf-v1.0-fd-20171005/fido-allowed-crypto-v1.0-fd-20171005.html
**Previous version:**
http://fidoalliance.org/specs/fido-allowed-crypto-v1.0-ps-20141208.html
**Editors:**
Dr. Joshua E. Hill, InfoGard Laboratories
Douglas Biggs, InfoGard Laboratories



---

## 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 Cryptographic 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. A list of current FIDO Alliance publications and the latest revision*

*of this technical report can be found in the FIDO Alliance specifications index at https://www.fidoalliance.org/specifications/.*

This document was published by the FIDO Alliance as a Final Document. If you wish to make comments regarding this document, please Contact Us. All comments are welcome.

Implementation of certain elements of this 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 this 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 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.

This document was published by the FIDO Alliance as a . If you wish to make comments regarding this document, please Contact Us. All comments are welcome.

Implementation of certain elements of this Specification 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 Specification 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 SPECIFICATION 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

## Version

This Document version (DV) is DV 1.0.1.

## Key Words

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].

# 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 cryptographic 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 by at least one major standards group (e.g., NIST, ANSI, ISO, IETF), 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 280 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.

If the security level stated is n, 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 Confidentiality Algorithms

> NOTE

Provide confidentiality, up to the stated security level.

| Algorithm | Specified in | Security Level (bits) |
|---|---|---|
| Three-Key Triple-DES | [ANSI-X9-52] | 112[1] |
| AES-128 | [FIPS197] | 128 |
| AES-192 | [FIPS197] | 192 |
| AES-256 | [FIPS197] | 256 |

[1] Based on the standard meet-in-the-middle attack.

## 3.2 Hashing Algorithms

NOTE

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

| Algorithm | Specified in | Security Level (bits) |
|---|---|---|
| SHA-256 | [FIPS180-4] | 128 |
| SHA-384 | [FIPS180-4] | 192 |
| SHA-512 | [FIPS180-4] | 256 |
| SHA-512/t, 256 ≤ t < 512 | [FIPS180-4] | t/2 |
| SHA3-256 | [FIPS202] | 128 |
| SHA3-384 | [FIPS202] | 192 |
| SHA3-512 | [FIPS202] | 256 |

## 3.3 Data Authentication Algorithms

NOTE

Provide data authentication.

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

| Algorithm | Specified in | Security Level (bits) |
|---|---|---|
| HMAC | [FIPS198-1] | Minimum of the length of the output of the hash used[2], one-half of the number of bits in the hash state[3], or the number of bits in the HMAC key. |
| CMAC | [SP800-38B] | Equal to the minimum of the strength of the underlying cipher and the length of the output MAC. |
| GMAC | [SP800-38D] | Equal to the minimum of the 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.

## 3.4 Key Protection Algorithms

> NOTE
>
> Provide confidentiality and data authentication.

| Algorithm | Specified in | Security Level (bits) |
|---|---|---|
| Key Wrapping | [SP800-38F] | Equal to the strength of the underlying cipher. |
| GCM Mode, with fixed length 96 bit IVs | [SP800-38D] | Equal to the strength of the underlying cipher. |
| CCM Mode | [SP800-38C] | Equal to the strength of the underlying cipher. |
| Encrypt-then-HMAC[4] | Encryption specification depends on the cipher selected. HMAC specification [FIPS198-1] | The minimum of the 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 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).

## 3.5 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 prove that the implementation uses the underlying platform certified RNG/RBG through Common Criteria, FIPS 140-2 (issued on August 7th 2015 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
>
> 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.

### 3.5.1 Physical/True (TRNG)/Non-Derterministic 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 Checks. 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.

The security strength (in bits) of an allowed physical/true random number generator is equivalent to the size (in bits) of the random bytes retrieved from it.

### 3.5.2 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 140-2 [FIPS140-2] validation.

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

## 3.6 Key Derivation Functions (KDFs)

Deriving keys.

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

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

Where PRF denotes an acceptable pseudorandom function as defined in [SP800-108].

## 3.7 Signature Algorithms

> NOTE
>
> Provide data authentication, and non-repudiation.

| Algorithm | Specified in | Security Level (bits) |
|---|---|---|
| ECDSA on P-256 | [ECDSA-ANSI], [FIPS186-4] | 128 |
| 2048-bit RSA PSS | [FIPS186-4] | 112 |
| 1024*n-bit RSA PKCS v1.5 (n=2,3,4) | [FIPS186-4] | 112 |
| ECDSA on secp256k1 | [ECDSA-ANSI], [FIPS186-4], Certicom SEC 2 | 126[7] |
| SM2 digital signatures (SM2 part 2) using the SM3 hash on the SM2 curve specified by OSCCA. | SM2椭圆曲线公钥密码算法 第1部分：总则, SM3密码杂凑算法 | 128 |
| Ed25519 | Draft RFC EDDSA | 128[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.

## 3.8 Anonymous Attestation Algorithms

> NOTE
>
> Provide anonymous attestation.

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

1. The 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 (Lq [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 | [FIDOEcdaaAlgorithm], section Object Formats and Algorithm Details, [TPMv2-Part4] | 128 |
| 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


**[ANSI-X9-52]**
. *Triple Data Encryption Algorithm Modes of Operation*. July 29, 1998. Current. URL:

**[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: The Elliptic Curve Digital Signature Algorithm (ECDSA), ANSI X9.62-2005*. November 2005. URL:

http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+X9.62%3A2005

**[FIDOEcdaaAlgorithm]**

R. Lindemann; J. Camenisch; M. Drijvers; A. Edgington; A. Lehmann; R. Urian. *FIDO ECDAA Algorithm*. Implementation Draft. URL: https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-ecdaa-v1.1-id-20170202.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

**[FIPS180-4]**

. *FIPS PUB 180-4: Secure Hash Standard (SHS)*. March 2012. URL: http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

**[FIPS186-4]**

. *FIPS PUB 186-4: Digital Signature Standard (DSS)*. 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)*. 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)*. 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*. 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*. URL: https://webstore.iec.ch/publication/10468

**[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

**[SP800-38B]**

M. Dworkin. *NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*. May 2005. URL: http://dx.doi.org/10.6028/NIST.SP.800-38B

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

**[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

**[SP800-90C]**

Elaine Barker; John Kelsey. *NIST Special Publication 800-90C: Recommendation for Random Bit Generator (RBG) Constructions*. August 2012. 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: Recommendation for Random Number Generation Using Deterministic Random Bit Generators*. August 2012.

URL: http://dx.doi.org/10.6028/NIST.SP.800-90Ar1

**[TPMv2-Part4]**

. *Trusted Platform Module Library, Part 4: Supporting Routines*. URL: http://www.trustedcomputinggroup.org/files/static_page_files/8C6CABBC-1A4B-B294-D0DA8CE1B452CAB4/TPM%20Rev%202.0%20Part%204%20-%20Supporting%20Routines%2001.16-code.pdf

## A.2 Informative references

**[RFC2119]**

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