FIDO Authenticator Allowed Cryptography List

FIDO Alliance Final Requirements Document 13 December 2017

This version:  
https://fidoalliance.org/specs/fido-security-requirements/fido-authenticator-allowed-cryptography-list-v1.1-fd-20171108.html

Previous version:  
https://fidoalliance.org/specs/fido-security-requirements-v1.0-fd-20170524/fido-authenticator-allowed-cryptography-list_20170524.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
supersedes this document. The most recent version of this document can be found on the FIDO Alliance Website at https://www.fidoalliance.org.

This document was published by the FIDO Alliance 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
- 2. Requirements for Additional Candidates
- 3. Allowed Cryptographic Functions
  - 3.1 Confidentiality Algorithms
  - 3.2 Hashing Algorithms
  - 3.3 Data Authentication Algorithms
  - 3.4 Key Protection Algorithms
  - 3.5 Random Number Generator
    - 3.5.1 Physical/True (TRNG)/Non-Deterministic Random Number/Bit Generator (NRBG) Requirements
    - 3.5.2 Deterministic Random Number (DRNG)/Bit Generator (DRBG) Requirements
  - 3.6 Key Derivation Functions (KDFs)
  - 3.7 Signature Algorithms
  - 3.8 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].
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 $2^{80}$ 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.

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-128</td>
<td>[FIPS197]</td>
<td>128</td>
</tr>
<tr>
<td>AES-192</td>
<td>[FIPS197]</td>
<td>192</td>
</tr>
<tr>
<td>AES-256</td>
<td>[FIPS197]</td>
<td>256</td>
</tr>
</tbody>
</table>


3.2 Hashing Algorithms

NOTE
Provide confidentiality, up to the stated security level.
### Algorithm Specified in Security Level (bits)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-256</td>
<td>[FIPS180-4]</td>
<td>128</td>
</tr>
<tr>
<td>SHA-384</td>
<td>[FIPS180-4]</td>
<td>192</td>
</tr>
<tr>
<td>SHA-512</td>
<td>[FIPS180-4]</td>
<td>256</td>
</tr>
<tr>
<td>SHA-512/t, 256 ≤ t &lt; 512</td>
<td>[FIPS180-4]</td>
<td>t/2</td>
</tr>
<tr>
<td>SHA3-256</td>
<td>[FIPS202]</td>
<td>128</td>
</tr>
<tr>
<td>SHA3-384</td>
<td>[FIPS202]</td>
<td>192</td>
</tr>
<tr>
<td>SHA3-512</td>
<td>[FIPS202]</td>
<td>256</td>
</tr>
</tbody>
</table>

### 3.3 Data Authentication Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMAC</td>
<td>[FIPS198-1]</td>
<td>Minimum of the length of the output of the hash used, one-half of the number of bits in the hash state, or the number of bits in the HMAC key.</td>
</tr>
<tr>
<td>CMAC</td>
<td>[SP800-38B]</td>
<td>Equal to the minimum of the strength of the underlying cipher and the length of the output MAC.</td>
</tr>
<tr>
<td>GMAC</td>
<td>[SP800-38D]</td>
<td>Equal to the minimum of the strength of the underlying cipher and the length of the output MAC.</td>
</tr>
</tbody>
</table>

[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.
Provide confidentiality and data authentication.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Wrapping</td>
<td>[SP800-38F]</td>
<td>Equal to the strength of the underlying cipher.</td>
</tr>
<tr>
<td>GCM Mode, with fixed length 96 bit IVs</td>
<td>[SP800-38D]</td>
<td>Equal to the strength of the underlying cipher.</td>
</tr>
<tr>
<td>CCM Mode</td>
<td>[SP800-38C]</td>
<td>Equal to the strength of the underlying cipher.</td>
</tr>
<tr>
<td>Encrypt-then-HMAC</td>
<td>Encryption specification depends on the cipher selected. HMAC specification [FIPS198-1]</td>
<td>The minimum of the strength of the cipher and the HMAC.</td>
</tr>
<tr>
<td>Encrypt-then-CMAC</td>
<td>Encryption specification depends on the cipher selected. CMAC specification [SP800-38B]</td>
<td>The minimum of the strength of the cipher and the CMAC.</td>
</tr>
</tbody>
</table>

[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 proof 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.

#### 3.5.1 Physical/True (TRNG)/Non-Deterministic Random Number/Bit Generator (NRBG)

**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.
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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source RBG is DRBG with access to Live Entropy Source or it is an NRBG.</td>
<td>[SP800-90C], section 6</td>
<td>Any security strength.</td>
</tr>
</tbody>
</table>

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],

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
</table>
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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDF in counter mode</td>
<td>[SP800-108]</td>
<td>min(Bit length of key derivation key Ki used as input, Security level of PRF)</td>
</tr>
<tr>
<td>KDF in feedback mode</td>
<td>[SP800-108]</td>
<td>min(Bit length of key derivation key Ki used as input, Security level of PRF)</td>
</tr>
<tr>
<td>KDF in double pipeline iteration mode</td>
<td>[SP800-108]</td>
<td>min(Bit length of key derivation key Ki used as input, Security level of PRF)</td>
</tr>
</tbody>
</table>

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

### 3.7 Signature Algorithms

**NOTE**

Provide data authentication, and non-repudiation.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECDSA on P-256</td>
<td>[ECDSA-ANSI], [FIPS186-4]</td>
<td>128</td>
</tr>
<tr>
<td>2048-bit RSA PSS</td>
<td>[FIPS186-4]</td>
<td>112</td>
</tr>
</tbody>
</table>
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

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 | [FIDOEcdaaaAlgorithm], section Object Formats and Algorithm Details, [TPMv2-Part4] | 128
A. References

A.1 Normative references

[ANSI-X9-52]

[DevScoDah2007]

[ECDSA-ANSI]

[FIDOEcdaaAlgorithm]

[FIPS140-2]
FIPS PUB 140-2: Security Requirements for Cryptographic Modules. May 2001. URL:

[FIPS180-4]
FIPS PUB 180-4: Secure Hash Standard (SHS). March 2012. URL:

[FIPS186-4]
FIPS PUB 186-4: Digital Signature Standard (DSS). July 2013. URL:

[FIPS197]

[FIPS198-1]

[FIPS202]

[ISO15946-5]

[SP800-108]
Lily Chen. NIST Special Publication 800-107: Recommendation for Key Derivation Using
A.2 Informative references

[SP800-38B]

[SP800-38C]

[SP800-38D]

[SP800-38F]

[SP800-90C]

[SP800-90ar1]

[TPMv2-Part4]

[RFC2119]