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 at https://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
   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

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

NOTE

Provide confidentiality, up to the stated security level.
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.

<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], [ISOIEC-10118-3]</td>
<td>128</td>
</tr>
<tr>
<td>SHA-384</td>
<td>[FIPS180-4], [ISOIEC-10118-3]</td>
<td>192</td>
</tr>
<tr>
<td>SHA-512</td>
<td>[FIPS180-4], [ISOIEC-10118-3]</td>
<td>256</td>
</tr>
<tr>
<td>SHA-512/t, 256 ≤ t &lt; 512</td>
<td>[FIPS180-4], [ISOIEC-10118-3]</td>
<td>t/2</td>
</tr>
<tr>
<td>SHA3-256</td>
<td>[FIPS202], [ISOIEC-10118-3]</td>
<td>128</td>
</tr>
<tr>
<td>SHA3-384</td>
<td>[FIPS202], [ISOIEC-10118-3]</td>
<td>192</td>
</tr>
<tr>
<td>SHA3-512</td>
<td>[FIPS202], [ISOIEC-10118-3]</td>
<td>256</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMAC</td>
<td>[FIPS198-</td>
<td>Minimum of the length of the output of the hash used...</td>
</tr>
</tbody>
</table>
bits in the hash state[^3], or the number of bits in the HMAC key.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMAC</td>
<td>[SP800-38B]</td>
<td>Equal to the minimum of the security 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 security strength of the underlying cipher and the length of the output MAC.</td>
</tr>
</tbody>
</table>

[^1]: Both due to the obvious guessing attack, and covers the case where the supplied key is hashed for the HMAC.

[^2]: 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.
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).

### 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**

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**

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
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
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>
<tbody>
<tr>
<td>HMAC_DRBG</td>
<td>[SP800-90ar1], Revision 1, section 10.1.2</td>
<td>The instantiated security level, as defined in [SP800-90ar1].</td>
</tr>
<tr>
<td>CTR_DRBG</td>
<td>[SP800-90ar1], Revision 1, section 10.2.1</td>
<td>The instantiated security level, as defined in [SP800-90ar1].</td>
</tr>
<tr>
<td>HASH_DRBG</td>
<td>[SP800-90ar1], Revision 1, section 10.1.1</td>
<td>The instantiated security level, as defined in [SP800-90ar1].</td>
</tr>
</tbody>
</table>

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

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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffie-Hellmann (DH) 2048-bit key</td>
<td>[SP800-56Ar3], [ISO11770-3]</td>
<td>$\geq 112$</td>
</tr>
<tr>
<td>EC-DH on P-256 or BrainpoolP256r1</td>
<td>[SP800-56Ar3], [ISO11770-3]</td>
<td>$\geq 128$</td>
</tr>
</tbody>
</table>
3.8 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 $K_i$ 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 $K_i$ 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 $K_i$ used as input, Security level of PRF)</td>
</tr>
<tr>
<td>HKDF</td>
<td>[SP800-56cr1], [RFC5869]</td>
<td>min(Bit length of key derivation key $K_i$ used as input, Security level of HMAC)</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECDSA on NIST P-256</td>
<td>[ECDSA-ANSI], [FIPS186-4], [ISOIEC-14888-3]</td>
<td>128</td>
</tr>
<tr>
<td>ECDSA on NIST P-384</td>
<td>[ECDSA-ANSI], [FIPS186-4], [ISOIEC-14888-3]</td>
<td>192</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Reference(s)</td>
<td>Security Strength</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ECDSA on NIST P-521</td>
<td>[ECDSA-ANSI], [FIPS186-4], [ISOIEC-14888-3]</td>
<td>256</td>
</tr>
<tr>
<td>ECDSA on BrainpoolP256r1</td>
<td>[RFC5639]</td>
<td>128</td>
</tr>
<tr>
<td>ECDSA on secp256k1</td>
<td>[ECDSA-ANSI], [FIPS186-4], [[!Certicom SEC 2]]</td>
<td>126 [7]</td>
</tr>
<tr>
<td>2048-bit RSA PSS</td>
<td>[FIPS186-4],[ISOIEC-9796-2]</td>
<td>112</td>
</tr>
<tr>
<td>1024*n-bit RSA PKCS v1.5 (n=2,3,4)</td>
<td>[FIPS186-4],[ISOIEC-9796-2]</td>
<td>112</td>
</tr>
<tr>
<td>1024*n-bit RSA PKCS v2.1 (PSS) (n=2,3,4)</td>
<td>[FIPS186-4],[ISOIEC-9796-2]</td>
<td>112</td>
</tr>
<tr>
<td>SM2 digital signatures (SM2 part 2)</td>
<td>SM2 [ISOIEC-14888-3] and SM3 [ISOIEC-10118-3]</td>
<td>128</td>
</tr>
<tr>
<td>SM2 digital signatures (SM2 part 2)</td>
<td>SM2 [ISOIEC-14888-3] and SM3 [ISOIEC-10118-3]</td>
<td>128</td>
</tr>
<tr>
<td>Ed25519</td>
<td>EDDSA [RFC8032]</td>
<td>128 [8]</td>
</tr>
<tr>
<td>Ed448</td>
<td>EDDSA [RFC8032]</td>
<td>224 [8]</td>
</tr>
</tbody>
</table>

[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 (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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Specified in</th>
<th>Security Level (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ED256-2</td>
<td>[FIDOEcdaaAlgorithm], section Object Formats and Algorithm Details, [DevScoDah2007]</td>
<td>112</td>
</tr>
<tr>
<td>ED512</td>
<td>[FIDOEcdaaAlgorithm], section Object Formats and Algorithm Details, [ISO15946-5]</td>
<td>128</td>
</tr>
<tr>
<td>ED638</td>
<td>[FIDOEcdaaAlgorithm], section Object Formats and Algorithm Details, [TPMv2-Part4]</td>
<td>128</td>
</tr>
</tbody>
</table>

A. References

A.1 Normative references

[DevScoDah2007]  

[ECDSA-ANSI]  

[FIDOEcdaaAlgorithm]  

[FIPS140-2]  

[FIPS140-3]  

[FIPS180-4]  
FIPS PUB 180-4: Secure Hash Standard (SHS). August 2015. URL:

[FIPS186-4]

[FIPS197]

[FIPS198-1]

[FIPS202]

[ISO15946-5]

[ISOIEC-10118-3]
- **SM3 Cryptographic Hash Algorithm**. October 2018. URL: https://www.iso.org/standard/67116.html

[ISOIEC-11770-3]

[ISOIEC-14888-3]

[ISOIEC-18033-3]

[ISOIEC-9796-2]

[RFC3447]

[RFC5639]

[RFC5869]

[RFC8032]

[SOGISCrypto]

[SP800-108]
A.2 Informative references