Abstract

The W3C Web Authentication (WebAuthn) specification and the FIDO Alliance FIDO2 Client to Authenticator Protocol (CTAP) specification use CBOR Object Signing and Encryption (COSE) algorithm identifiers. This specification registers the following algorithms in the IANA "COSE Algorithms" registry, which are used by WebAuthn and CTAP implementations: RSASSA-PKCS1-v1_5 using SHA-256, SHA-384, SHA-512, and SHA-1, and ECDSA using the secp256k1 curve and SHA-256. It registers the secp256k1 elliptic curve in the IANA "COSE Elliptic Curves" registry. Also, for use with JSON Object Signing and Encryption (JOSE), it registers the algorithm ECDSA using the secp256k1 curve and SHA-256 in the IANA "JSON Web Signature and Encryption Algorithms" registry and the secp256k1 elliptic curve in the IANA "JSON Web Key Elliptic Curve" registry.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 26, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

This specification defines how to use several algorithms with CBOR Object Signing and Encryption (COSE) [RFC8152] that are used by implementations of the W3C Web Authentication (WebAuthn) [WebAuthn] and FIDO Alliance FIDO2 Client to Authenticator Protocol (CTAP) [CTAP] specifications. These specification registers these algorithms in the IANA "COSE Algorithms" registry [IANA.COSE.Algorithms] and registers a related elliptic curve in the IANA "COSE Elliptic Curves" registry [IANA.COSE.Curves]. This specification also registers a corresponding algorithm for use with JSON Object Signing and Encryption (JOSE) [RFC7515] in the IANA "JSON
Web Signature and Encryption Algorithms" registry [IANA.JOSE.Algorithms] and registers a related elliptic curve in the IANA "JSON Web Key Elliptic Curve" registry [IANA.JOSE.Curves].

1.1. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. RSASSA-PKCS1-v1_5 Signature Algorithm

The RSASSA-PKCS1-v1_5 signature algorithm is defined in [RFC8017]. The RSASSA-PKCS1-v1_5 signature algorithm is parameterized with a hash function (h).

A key of size 2048 bits or larger MUST be used with these algorithms. Implementations need to check that the key type is ‘RSA’ when creating or verifying a signature.

The RSASSA-PKCS1-v1_5 algorithms specified in this document are in the following table.
### Table 1: RSASSA-PKCS1-v1_5 Algorithm Values

Note that these algorithms are already present in the IANA "JSON Web Signature and Encryption Algorithms" registry [IANA.JOSE.Algorithms], and so these registrations are only for the IANA "COSE Algorithms" registry [IANA.COSE.Algorithms].

#### 3. Using secp256k1 with JOSE and COSE

This section defines algorithm encodings and representations enabling the Standards for Efficient Cryptography Group (SECG) elliptic curve secp256k1 [SEC2] to be used for JOSE [RFC7515] and COSE [RFC8152] messages.

##### 3.1. JOSE and COSE secp256k1 Curve Key Representations

The Standards for Efficient Cryptography Group (SECG) elliptic curve secp256k1 [SEC2] is represented in a JSON Web Key (JWK) [RFC7517] using these values:

- "kty": "EC"
- "crv": "secp256k1"
plus the values needed to represent the curve point, as defined in
Section 6.2.1 of [RFC7518]. As a compressed point encoding
representation is not defined for JWK elliptic curve points, the
uncompressed point encoding defined there MUST be used. The "x" and
"y" values represented MUST both be exactly 256 bits, with any
leading zeros preserved. Other optional values such as "alg" MAY
also be present.

It is represented in a COSE_Key [RFC8152] using these values:

- "kty" (1): "EC2" (2)
- "crv" (-1): "secp256k1" (TBD - requested assignment 8)

plus the values needed to represent the curve point, as defined in
Section 13.1.1 of [RFC8152]. Either the uncompressed or compressed
point encoding representations defined there can be used. The "x"
value represented MUST be exactly 256 bits, with any leading zeros
preserved. If the uncompressed representation is used, the "y" value
represented MUST likewise be exactly 256 bits, with any leading zeros
preserved; if the compressed representation is used, the "y" value
MUST be a boolean value, as specified in Section 13.1.1 of [RFC8152].
Other optional values such as "alg" (3) MAY also be present.

3.2. ECDSA Signature with secp256k1 Curve

The ECDSA signature algorithm is defined in [DSS]. This
specification defines the "ES256K" algorithm identifier, which is
used to specify the use of ECDSA with the secp256k1 curve and the
SHA-256 [DSS] cryptographic hash function. Implementations need to
check that the key type is "EC" for JOSE or "EC2" (2) for COSE and
that the curve of the key is secp256k1 when creating or verifying a
signature.

The ECDSA secp256k1 SHA-256 digital signature is generated as
follows:

1. Generate a digital signature of the JWS Signing Input or the COSE
   Sig_structure using ECDSA secp256k1 SHA-256 with the desired
   private key. The output will be the pair (R, S), where R and S
   are 256-bit unsigned integers.

2. Turn R and S into octet sequences in big-endian order, with each
   array being be 32 octets long. The octet sequence
   representations MUST NOT be shortened to omit any leading zero
   octets contained in the values.
3. Concatenate the two octet sequences in the order R and then S.  
   (Note that many ECDSA implementations will directly produce this 
   concatenation as their output.)

4. The resulting 64-octet sequence is the JWS Signature or COSE 
   signature value.

Implementations SHOULD use a deterministic algorithm to generate the 
ECDSA nonce, k, such as [RFC6979].  However, in situations where 
devices are vulnerable to physical attacks, deterministic ECDSA has 
been shown to be susceptible to fault injection attacks [Kudelski17] 
[EuroSP18].  Where this is a possibility, implementations SHOULD 
implement appropriate countermeasures.  Where there are specific 
certification requirements (such as FIPS approval), implementors 
should check whether deterministic ECDSA is an approved nonce 
generation method.

The ECDSA secp256k1 SHA-256 algorithm specified in this document uses 
these identifiers:

+----------+-------------------+----------------------+-------------+
| JOSE Alg | COSE Alg Value    | Description          | Recommended |
| Name     |                   |                      |             |
+----------+-------------------+----------------------+-------------+
| ES256K   | TBD (requested    | ECDSA using          | Yes         |
|          | assignment -46)   | secp256k1 curve and  |             |
|          |                   | SHA-256              |             |
+----------+-------------------+----------------------+-------------+

Table 2: ECDSA Algorithm Values

Implementation of this algorithm is RECOMMENDED because of its 
widespread use in decentralized systems and those that chose it over 
the NIST curves.

When using a JWK or COSE_Key for this algorithm, the following checks 
are made:

- The "kty" field MUST be present and it MUST be "EC" for JOSE or 
  "EC2" for COSE.

- The "crv" field MUST be present and it MUST represent the 
  "secp256k1" elliptic curve.

- If the "alg" field is present, it MUST represent the "ES256K" 
  algorithm.
If the "key_ops" field is present, it MUST include "sign" when creating an ECDSA signature.

If the "key_ops" field is present, it MUST include "verify" when verifying an ECDSA signature.

If the JWK _use_ field is present, its value MUST be "sig".

### 3.3. Other Uses of the secp256k1 Elliptic Curve

This specification defines how to use the secp256k1 curve for ECDSA signatures for both JOSE and COSE implementations. While in theory, the curve could also be used for ECDH-ES key agreement, it is beyond the scope of this specification to state whether this is or is not advisable. Thus, whether to recommend its use with ECDH-ES is left for experts to decide in future specifications.

When used for ECDSA, the secp256k1 curve MUST be used only with the "ES256K" algorithm identifier and not any others, including not with "ES256".

### 4. IANA Considerations

#### 4.1. COSE Algorithms Registrations

This section registers the following values in the IANA "COSE Algorithms" registry [IANA.COSE.Algorithms].

- **Name**: RS256
  - **Value**: TBD (temporary assignment -257 already in place)
  - **Description**: RSASSA-PKCS1-v1_5 using SHA-256
  - **Reference**: Section 2 of this document
  - **Recommended**: No

- **Name**: RS384
  - **Value**: TBD (temporary assignment -258 already in place)
  - **Description**: RSASSA-PKCS1-v1_5 using SHA-384
  - **Reference**: Section 2 of this document
  - **Recommended**: No

- **Name**: RS512
  - **Value**: TBD (temporary assignment -259 already in place)
  - **Description**: RSASSA-PKCS1-v1_5 using SHA-512
  - **Reference**: Section 2 of this document
  - **Recommended**: No

- **Name**: RS1
  - **Value**: TBD (temporary assignment -65535 already in place)
4.2. COSE Elliptic Curves Registrations

This section registers the following value in the IANA "COSE Elliptic Curves" registry [IANA.COSE.Curves].

- Name: secp256k1
- Value: TBD (requested assignment 8)
- Key Type: EC2
- Description: SECG secp256k1 curve
- Change Controller: IESG
- Reference: Section 3.1 of [[this specification]]
- Recommended: Yes

4.3. JOSE Algorithms Registrations

This section registers the following value in the IANA "JSON Web Signature and Encryption Algorithms" registry [IANA.JOSE.Algorithms].

- Algorithm Name: ES256K
- Algorithm Description: ECDSA using secp256k1 curve and SHA-256
- Algorithm Usage Locations: alg
- JOSE Implementation Requirements: Optional
- Change Controller: IESG
- Reference: Section 3.2 of [[this specification]]
- Algorithm Analysis Document(s): [SEC2]

4.4. JSON Web Key Elliptic Curves Registrations

This section registers the following value in the IANA "JSON Web Key Elliptic Curve" registry [IANA.JOSE.Curves].

- Curve Name: secp256k1
- Curve Description: SECG secp256k1 curve
- JOSE Implementation Requirements: Optional
- Change Controller: IESG
- Specification Document(s): Section 3.1 of [[this specification]]
5. Security Considerations

5.1. RSA Key Size Security Considerations

The security considerations on key sizes for RSA algorithms from Section 6.1 of [RFC8230] also apply to the RSA algorithms in this specification.

5.2. RSASSA-PKCS1-v1_5 with SHA-2 Security Considerations

The security considerations on the use of RSASSA-PKCS1-v1_5 with SHA-2 hash functions from Section 8.3 of [RFC7518] also apply to their use in this specification. For that reason, these algorithms are registered as being "Not Recommended".

5.3. RSASSA-PKCS1-v1_5 with SHA-1 Security Considerations

The security considerations on the use of the SHA-1 hash function from [RFC6194] apply in this specification. For that reason, the "RS1" algorithm is registered as "Deprecated". Likewise, the exponent restrictions described in Section 8.3 of [RFC7518] also apply.

A COSE algorithm identifier for this algorithm is nonetheless being registered because deployed TPMs continue to use it, and therefore WebAuthn implementations need a COSE algorithm identifier for "RS1" when TPM attestations using this algorithm are being represented. New COSE applications MUST NOT use this algorithm.

5.4. secp256k1 Security Considerations

Care should be taken that a secp256k1 key is not mistaken for a P-256 [RFC7518] key, given that their representations are the same except for the "crv" value.

The procedures and security considerations described in the [SEC1], [SEC2], and [DSS] specifications apply to implementations of this specification.

6. References

6.1. Normative References


6.2. Informative References


[IANA.JOSE.Curves] IANA, "JSON Web Key Elliptic Curve", <https://www.iana.org/assignments/jose/jose.xhtml#web-key-elliptic-curve>.

Internet-DrafCOSE and JOSE Registrations for WebAuthn Algor October 2019


Acknowledgements


Document History

[ [ to be removed by the RFC Editor before publication as an RFC ] ]

-02

- Addressed working group last call comments. Thanks to J.C. Jones, Kevin Jacobs, Jim Schaad, Neil Madden, and Benjamin Kaduk for their useful feedback.

-01

- Changed the JOSE curve identifier from "P-256K" to "secp256k1".

- Specified that secp256k1 signing is done using the SHA-256 hash function.

-00

- Created the initial working group draft from draft-jones-cose-additional-algorithms-00, changing only the title, date, and history entry.

Jones Expires April 26, 2020 [Page 12]
Author’s Address

Michael B. Jones
Microsoft

Email: mbj@microsoft.com
URI: http://self-issued.info/