Stream: Internet Engineering Task Force (IETF)
RFC: 9864
Updates: 7518, 8037, 9053
Category: Standards Track
Published: October 2025
ISSN: 2070-1721

Authors: M.B. Jones *Self-Issued Consulting*
O. Steele *Tradeverifyd*

RFC 9864
Fully-Specified Algorithms for JSON Object Signing and Encryption (JOSE) and CBOR Object Signing and Encryption (COSE)

Abstract

This specification refers to cryptographic algorithm identifiers that fully specify the cryptographic operations to be performed, including any curve, key derivation function (KDF), and hash functions, as being "fully specified". It refers to cryptographic algorithm identifiers that require additional information beyond the algorithm identifier to determine the cryptographic operations to be performed as being "polymorphic". This specification creates fully-specified algorithm identifiers for registered JSON Object Signing and Encryption (JOSE) and CBOR Object Signing and Encryption (COSE) polymorphic algorithm identifiers, enabling applications to use only fully-specified algorithm identifiers. It deprecates those polymorphic algorithm identifiers.

This specification updates RFCs 7518, 8037, and 9053. It deprecates polymorphic algorithms defined by RFCs 8037 and 9053 and provides fully-specified replacements for them. It adds to the instructions to designated experts in RFCs 7518 and 9053.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The IANA algorithm registries for JSON Object Signing and Encryption (JOSE) algorithms [IANA.JOSE] and CBOR Object Signing and Encryption (COSE) algorithms [IANA.COSE] contain two kinds of algorithm identifiers:

Fully Specified
: Those that fully determine the cryptographic operations to be performed, including any curve, key derivation function (KDF), and hash functions. Examples are RS256 and ES256K in both JOSE [IANA.JOSE] and COSE [IANA.COSE] and ES256 in JOSE.

Polymorphic
: Those requiring information beyond the algorithm identifier to determine the cryptographic operations to be performed. Such additional information could include the actual key value and a curve that it uses. Examples are the Edwards-curve Digital Signature Algorithm (EdDSA) in both JOSE [IANA.JOSE] and COSE [IANA.COSE] and ES256 in COSE.

This matters because many protocols negotiate supported operations using only algorithm identifiers. For instance, OAuth Authorization Server Metadata [RFC8414] uses negotiation parameters like these (from an example in that specification):

```
  "token_endpoint_auth_signing_alg_values_supported":
    ["RS256", "ES256"]
```

OpenID Connect Discovery [OpenID.Discovery] likewise negotiates supported algorithms using `"alg"` and `"enc"` values. W3C Web Authentication [WebAuthn] and the FIDO Client to Authenticator Protocol (CTAP) [FIDO2] negotiate using COSE `"alg"` numbers.

This does not work for polymorphic algorithms. For instance, with `EdDSA`, it is not known which of the curves `Ed25519` and/or `Ed448` are supported. This causes real problems in practice.

WebAuthn contains this de facto algorithm definition to work around this problem:

```
   -8 (EdDSA), where crv is 6 (Ed25519)
```

This redefines the COSE `EdDSA` algorithm identifier for the purposes of WebAuthn to restrict it to using the `Ed25519` curve -- making it non-polymorphic so that algorithm negotiation can succeed, but also effectively eliminating the possibility of using `Ed448`. Other similar workarounds for polymorphic algorithm identifiers are used in practice.

Note that using fully-specified algorithms is sometimes referred to as the "cipher suite" approach; using polymorphic algorithms is sometimes referred to as the "à la carte" approach.

This specification creates fully-specified algorithm identifiers for registered polymorphic JOSE and COSE algorithms and their parameters, enabling applications to use only fully-specified algorithm identifiers. Furthermore, it deprecates the practice of registering polymorphic algorithm identifiers.

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. Fully-Specified Digital Signature Algorithm Identifiers

This section creates fully-specified digital signature algorithm identifiers for a set of registered polymorphic JOSE and COSE algorithms and their parameters.

2.1. Elliptic Curve Digital Signature Algorithm (ECDSA)

[RFC9053] defines a way to use the Elliptic Curve Digital Signature Algorithm (ECDSA) with COSE. The COSE algorithm registrations for ECDSA are polymorphic, since they do not specify the curve used. For instance, `ES256` is defined as "ECDSA w/ SHA-256" in Section 2.1 of [RFC9053]. (The corresponding JOSE registrations in [RFC7518] are fully specified.)

The following fully-specified COSE ECDSA algorithms are defined by this specification:

Name	COSE Value	Description	COSE Recommended
ESP256	-9	ECDSA using P-256 curve and SHA-256	Yes
ESP384	-51	ECDSA using P-384 curve and SHA-384	Yes
ESP512	-52	ECDSA using P-521 curve and SHA-512	Yes
ESB256	-265	ECDSA using BrainpoolP256r1 curve and SHA-256	No
ESB320	-266	ECDSA using BrainpoolP320r1 curve and SHA-384	No
ESB384	-267	ECDSA using BrainpoolP384r1 curve and SHA-384	No
ESB512	-268	ECDSA using BrainpoolP512r1 curve and SHA-512	No

Table 1: ECDSA Algorithm Values

2.2. Edwards-curve Digital Signature Algorithm (EdDSA)

[RFC8037] defines a way to use EdDSA with JOSE, and [RFC9053] defines a way to use it with COSE. Both register polymorphic `EdDSA` algorithm identifiers.

The following fully-specified JOSE and COSE EdDSA algorithms are defined by this specification:

Name	COSE Value	Description	JOSE Implementation Requirements	COSE Recommended
Ed25519	-19	EdDSA using the Ed25519 parameter set in Section 5.1 of [RFC8032]	Optional	Yes
Ed448	-53	EdDSA using the Ed448 parameter set in Section 5.2 of [RFC8032]	Optional	Yes

Table 2: EdDSA Algorithm Values

3. Fully-Specified Encryption

This section describes the construction of fully-specified encryption algorithm identifiers in the context of the JOSE and COSE encryption schemes JSON Web Encryption (JWE), as described in [RFC7516] and [RFC7518], and COSE encryption, as described in [RFC9052] and [RFC9053].

Using fully-specified encryption algorithms enables the sender and receiver to agree on all mandatory security parameters. They also enable protocols to specify an allow list of algorithm combinations that does not include polymorphic combinations, preventing problems such as cross-curve key establishment, cross-protocol symmetric encryption, or mismatched KDF size to symmetric key scenarios.

Both JOSE and COSE have operations that take multiple algorithms as parameters. Encrypted objects in JOSE [RFC7516] use two algorithm identifiers: the first in the `"alg"` (Algorithm) Header Parameter, which specifies how to determine the content encryption key, and the second in the `"enc"` (Encryption Algorithm) Header Parameter, which specifies the content encryption algorithm. Likewise, encrypted COSE objects can use multiple algorithms for corresponding purposes. This section describes how to fully specify encryption algorithms for JOSE and COSE.

To perform fully-specified encryption in JOSE, the `"alg"` value **MUST** specify all parameters for key establishment or derive some of them from the accompanying `"enc"` value, and the `"enc"` value **MUST** specify all parameters for symmetric encryption. For example, encryption via JWE using an `"alg"` value of "A128KW" (AES Key Wrap using 128-bit key) and an `"enc"` value of "A128GCM" (AES GCM using 128-bit key) uses fully-specified algorithms.

Note that in JOSE, there is the option to derive some cryptographic parameters used in the `"alg"` computation from the accompanying `"enc"` value. For example, the keydatalen KDF parameter value for "ECDH-ES" is determined from the `"enc"` value, as described in Section 4.6.2 of [RFC7518]. For the purposes of an `"alg"` value being fully specified, deriving parameters from `"enc"` does not make the algorithm polymorphic, as the computation is still fully determined by the algorithm identifiers used. This option is not present in COSE.

To perform fully-specified encryption in COSE, the outer `"alg"` value **MUST** specify all parameters for key establishment, and the inner `"alg"` value **MUST** specify all parameters for symmetric encryption. For example, encryption via COSE using an outer `"alg"` value of "A128KW" and an inner `"alg"` value of "A128GCM" uses fully-specified algorithms. Note that when using COSE_Encrypt, as specified in Section 5.1 of [RFC9052], the outer `"alg"` is communicated in the headers of the COSE_Encrypt object and the inner `"alg"` is communicated in the headers of the COSE_recipient object.

While this specification provides a definition of what fully-specified encryption algorithm identifiers are for both JOSE and COSE, it does not deprecate any polymorphic encryption algorithms, since replacements for them are not provided by this specification. This is discussed in Section 6.2.

3.1. Fully-Specified Encryption Algorithms

Many of the registered JOSE and COSE algorithms used for encryption are already fully specified. This section discusses them.

All the symmetric encryption algorithms registered by [RFC7518] and [RFC9053] are fully specified. An example of a fully-specified symmetric encryption algorithm is "A128GCM" (AES GCM using 128-bit key).

In both JOSE and COSE, all registered key wrapping algorithms are fully specified, as are the algorithms performing key wrapping using AES GCM. An example of a fully-specified key wrapping algorithm is "A128KW" (AES Key Wrap using 128-bit key).

The JOSE "dir" and COSE "direct" algorithms are fully specified. The COSE direct+HKDF algorithms are fully specified.

The JOSE algorithms performing Key Encryption with PBES2 are fully specified.

3.2. Polymorphic Encryption Algorithms

Some of the registered JOSE and COSE algorithms used for encryption are polymorphic. This section discusses them.

The Elliptic Curve Diffie-Hellman (ECDH) key establishment algorithms in both JOSE and COSE are polymorphic because they do not specify the elliptic curve to be used for the key. This is true of the ephemeral key for the Ephemeral-Static (ES) algorithms registered for JOSE and COSE and of the static key for the Static-Static (SS) algorithms registered by COSE. See more discussion of ECDH algorithms in Section 6.2.

4. IANA Considerations

4.1. JOSE Algorithm Registrations

IANA has registered the values in this section in the "JSON Web Signature and Encryption Algorithms" registry [IANA.JOSE] established by [RFC7518] and has listed this document as an additional reference for the registry.

4.1.1. Fully-Specified JOSE Algorithm Registrations

Algorithm Name: Ed25519
Algorithm Description: EdDSA using the Ed25519 parameter set in Section 5.1 of [RFC8032]
Algorithm Usage Locations: alg
JOSE Implementation Requirements: Optional
Change Controller: IETF
Reference: Section 2.2 of RFC 9864
Algorithm Analysis Document(s): [RFC8032]

Algorithm Name: Ed448
Algorithm Description: EdDSA using the Ed448 parameter set in Section 5.2 of [RFC8032]
Algorithm Usage Locations: alg
JOSE Implementation Requirements: Optional
Change Controller: IETF
Reference: Section 2.2 of RFC 9864
Algorithm Analysis Document(s): [RFC8032]

4.1.2. Deprecated Polymorphic JOSE Algorithm Registration

IANA has updated the status to "Deprecated" for the following registration.

Algorithm Name: EdDSA
Algorithm Description: EdDSA signature algorithms
Algorithm Usage Locations: alg
JOSE Implementation Requirements: Deprecated
Change Controller: IETF
Reference: Section 2.2 of RFC 9864
Algorithm Analysis Document(s): [RFC8032]

4.2. COSE Algorithm Registrations

IANA has registered the following values in the "COSE Algorithms" registry [IANA.COSE] established by [RFC9053] and [RFC9054] and has added this document as an additional reference for the registry.

4.2.1. Fully-Specified COSE Algorithm Registrations

Name: ESP256
Value: -9
Description: ECDSA using P-256 curve and SHA-256
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.1 of RFC 9864
Recommended: Yes

Name: ESP384
Value: -51
Description: ECDSA using P-384 curve and SHA-384
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.1 of RFC 9864
Recommended: Yes

Name: ESP512

Value: -52
Description: ECDSA using P-521 curve and SHA-512
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.1 of RFC 9864
Recommended: Yes

Name: ESB256
Value: -265
Description: ECDSA using BrainpoolP256r1 curve and SHA-256
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.1 of RFC 9864
Recommended: No

Name: ESB320
Value: -266
Description: ECDSA using BrainpoolP320r1 curve and SHA-384
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.1 of RFC 9864
Recommended: No

Name: ESB384
Value: -267
Description: ECDSA using BrainpoolP384r1 curve and SHA-384
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.1 of RFC 9864
Recommended: No

Name: ESB512
Value: -268
Description: ECDSA using BrainpoolP512r1 curve and SHA-512
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.1 of RFC 9864
Recommended: No

Name: Ed25519
Value: -19
Description: EdDSA using the Ed25519 parameter set in Section 5.1 of [RFC8032]
Capabilities: [kty]

Change Controller: IETF
Reference: Section 2.2 of RFC 9864
Recommended: Yes

Name: Ed448
Value: -53
Description: EdDSA using the Ed448 parameter set in Section 5.2 of [RFC8032]
Capabilities: [kty]
Change Controller: IETF
Reference: Section 2.2 of RFC 9864
Recommended: Yes

4.2.2. Deprecated Polymorphic COSE Algorithm Registrations

IANA has updated the status to "Deprecated" and has added this document as a reference for the following registrations.

Name: ES256
Value: -7
Description: ECDSA w/ SHA-256
Capabilities: [kty]
Change Controller: IETF
Reference: [RFC9053] and RFC 9864
Recommended: Deprecated

Name: ES384
Value: -35
Description: ECDSA w/ SHA-384
Capabilities: [kty]
Change Controller: IETF
Reference: [RFC9053] and RFC 9864
Recommended: Deprecated

Name: ES512
Value: -36
Description: ECDSA w/ SHA-512
Capabilities: [kty]
Change Controller: IETF
Reference: [RFC9053] and RFC 9864
Recommended: Deprecated

Name: EdDSA
Value: -8
Description: EdDSA

Capabilities: [kty]
Change Controller: IETF
Reference: [RFC9053] and RFC 9864
Recommended: Deprecated

4.3. Updated Review Instructions for Designated Experts

4.3.1. JSON Web Signature and Encryption Algorithms

The review instructions for the designated experts [RFC8126] for the "JSON Web Signature and Encryption Algorithms" registry [IANA.JOSE] in Section 7.1 of [RFC7518] have been updated to include an additional review criterion:

- Only fully-specified algorithm identifiers may be registered. Polymorphic algorithm identifiers must not be registered.

4.3.2. COSE Algorithms

The review instructions for the designated experts [RFC8126] for the "COSE Algorithms" registry [IANA.COSE] in Section 10.4 of [RFC9053] have been updated to include an additional review criterion:

- Only fully-specified algorithm identifiers may be registered. Polymorphic algorithm identifiers must not be registered.

4.4. Defining "Deprecated" and "Prohibited"

The terms "Deprecated" and "Prohibited" as used by JOSE and COSE registrations are currently undefined. Furthermore, while in [RFC7518] JOSE specifies that both "Deprecated" and "Prohibited" can be used, in [RFC8152] COSE specifies the use of "Deprecated" but not "Prohibited". (Note that [RFC8152] has been obsoleted by [RFC9052].) This section defines these terms for use by both JOSE and COSE IANA registrations in a consistent manner, eliminating this potentially confusing inconsistency.

For purposes of use in the "JOSE Implementation Requirements" columns in the IANA JOSE registries [IANA.JOSE] and in the "Recommended" columns in the IANA COSE registries [IANA.COSE], these terms are defined as follows:

Deprecated
    There is a preferred mechanism to achieve functionality similar to that referenced by the identifier; this replacement functionality **SHOULD** be utilized in new deployments in preference to the deprecated identifier, unless there exist documented operational or regulatory requirements that prevent migration away from the deprecated identifier.

Prohibited
    The identifier and the functionality that it references **MUST NOT** be used. (Identifiers may be designated as "Prohibited" due to security flaws, for instance.)

For completeness, these definitions bring the set of defined terms for use in the "Recommended" columns in the IANA COSE registries [IANA.COSE] to "Yes" [RFC8152], "No" [RFC8152], "Filter Only" [RFC9054], "Prohibited", and "Deprecated". This updates the definitions of the "Recommended" columns in these registries to be:

Recommended

Does the IETF have a consensus recommendation to use the algorithm? The legal values are "Yes", "No", "Filter Only", "Prohibited", and "Deprecated".

The set of defined terms for use in the "JOSE Implementation Requirements" columns in the IANA JOSE registries [IANA.JOSE] are unchanged.

Note that the terms "Deprecated" and "Prohibited" have been used with a multiplicity of different meanings in various specifications, sometimes without actually being defined in those specifications. For instance, a variation of the term "Deprecated" is used in the title of [RFC8996], but the actual specification text uses the terminology "**MUST NOT** be used".

The definitions above were chosen because they are consistent with all existing registrations in both JOSE and COSE; none will need to change. Furthermore, they are consistent with their existing usage in JOSE. The only net change is to enable a clear distinction between "Deprecated" and "Prohibited" in future COSE registrations.

5. Key Representations

The key representations for the new fully-specified algorithms defined by this specification are the same as those for the polymorphic algorithms that they replace, other than the `"alg"` value, if included. For instance, the representation for a key used with the `Ed25519` algorithm is the same as that specified in [RFC8037], except that the `"alg"` value would be `Ed25519` rather than `EdDSA`, if included.

6. Notes on Algorithms Not Updated

Some existing polymorphic algorithms are not updated by this specification. This section discusses why they have not been updated.

6.1. RSA Signing Algorithms

There are different points of view on whether the `RS256`, `RS384`, and `RS512` algorithms should be considered fully specified or not, because they can operate on keys of different sizes. For instance, they can use both 2048- and 4096-bit keys. The same is true of the `PS*` algorithms.

This document does not describe or request registration of any fully-specified RSA algorithms. Some RSA signing implementations, such as FIPS-compliant Hardware Security Modules (HSMs) [FIPS.140-3] limit RSA key parameters to specific values with acceptable security characteristics. This approach could be extended to define fully-specified RSA algorithms in the future.

That said, should it be useful at some point to have RSA algorithm identifiers that are specific to particular key characteristics, a future specification could always register them.

6.2. ECDH Key Agreement Algorithms

This specification does not update the ECDH algorithms, but it describes how to potentially do so in the future, if needed. The registered JOSE and COSE ECDH algorithms are polymorphic because they do not specify the curve to be used for the ephemeral key.

Fully-specified versions of these algorithms would specify all choices needed, including the KDF and the curve. For instance, an algorithm performing ECDH-ES using the Concat KDF and the P-256 curve would be fully specified and could be defined and registered. While this specification does not define and register such replacement algorithms, other specifications could do so in the future, if desired.

6.3. HSS/LMS Hash-Based Digital Signature Algorithm

The HSS-LMS algorithm registered by COSE is polymorphic. It is polymorphic because the algorithm identifier does not specify the hash function to be used. Like ECDH, this specification does not register replacement algorithms, but future specifications could do so.

7. Security Considerations

The security considerations for ECDSA in [RFC7518], for EdDSA in [RFC8037], and for ECDSA and EdDSA in [RFC9053] apply.

The security considerations for preventing cross-protocol attacks described in [RFC9459] apply.

An "attack signature" is a unique pattern or characteristic used to identify malicious activity, enabling systems to detect and respond to known threats. The digital signature and key establishment algorithms used by software can contribute to an attack signature. By varying the identifier used for an algorithm, some software systems may attempt to evade rule-based detection and classification. Rule-based detection and classification systems may need to update their rules to account for fully-specified algorithms. These systems should be aware that writing rules for polymorphic algorithms is more difficult, as each variant of the algorithm must be accounted for. For example, ES384 in COSE might be used with three different keys, each with a different curve.

A cryptographic key **MUST** be used with only a single algorithm unless the use of the same key with different algorithms is proven secure. See [Reuse25519] for an example of such a proof. As a result, it is **RECOMMENDED** that the algorithm parameter of JSON Web Keys and COSE Keys be present, unless there exists some other mechanism for ensuring that the key is used as intended.

In COSE, preventing cross-protocol attacks, such as those described in [RFC9459], can be accomplished in two ways:

1. Allow only authenticated content encryption (Authenticated Encryption with Associated Data (AEAD)) algorithms.

2. Bind the potentially unauthenticated content encryption algorithm to be used to the key protection algorithm so that different content encryption algorithms result in different content encryption keys.

Which choice to use in which circumstances is beyond the scope of this specification.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/info/rfc7516>.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curve Diffie-Hellman (ECDH) and Signatures in JSON Object Signing and Encryption (JOSE)", RFC 8037, DOI 10.17487/RFC8037, January 2017, <https://www.rfc-editor.org/info/rfc8037>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

[RFC9053] Schaad, J., "CBOR Object Signing and Encryption (COSE): Initial Algorithms", RFC 9053, DOI 10.17487/RFC9053, August 2022, <https://www.rfc-editor.org/info/rfc9053>.

8.2. Informative References

[FIDO2] Bradley, J., Jones, M.B., Kumar, A., Lindemann, R., Verrept, J., and D. Waite, "Client to Authenticator Protocol (CTAP)", FIDO Alliance Proposed Standard, 14 July 2025, <https://fidoalliance.org/specs/fido-v2.2-ps-20250714/fido-client-to-authenticator-protocol-v2.2-ps-20250714.html>.

[FIPS.140-3] NIST, "Security Requirements for Cryptographic Modules", NIST FIPS 140-3, DOI 10.6028/NIST.FIPS.140-3, March 2019, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf>.

[IANA.COSE] IANA, "CBOR Object Signing and Encryption (COSE)", <https://www.iana.org/assignments/cose/>.

[IANA.JOSE] IANA, "JSON Object Signing and Encryption (JOSE)", <https://www.iana.org/assignments/jose/>.

[OpenID.Discovery] Sakimura, N., Bradley, J., Jones, M., and E. Jay, "OpenID Connect Discovery 1.0 incorporating errata set 2", 15 December 2023, <https://openid.net/specs/openid-connect-discovery-1_0.html>.

[Reuse25519] Thormarker, E., "On using the same key pair for Ed25519 and an X25519 based KEM", 23 April 2021, <https://eprint.iacr.org/2021/509.pdf>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/info/rfc7518>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[RFC8414] Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Authorization Server Metadata", RFC 8414, DOI 10.17487/RFC8414, June 2018, <https://www.rfc-editor.org/info/rfc8414>.

[RFC8996] Moriarty, K. and S. Farrell, "Deprecating TLS 1.0 and TLS 1.1", BCP 195, RFC 8996, DOI 10.17487/RFC8996, March 2021, <https://www.rfc-editor.org/info/rfc8996>.

[RFC9054] Schaad, J., "CBOR Object Signing and Encryption (COSE): Hash Algorithms", RFC 9054, DOI 10.17487/RFC9054, August 2022, <https://www.rfc-editor.org/info/rfc9054>.

[RFC9459] Housley, R. and H. Tschofenig, "CBOR Object Signing and Encryption (COSE): AES-CTR and AES-CBC", RFC 9459, DOI 10.17487/RFC9459, September 2023, <https://www.rfc-editor.org/info/rfc9459>.

[WebAuthn] Hodges, J., Ed., Jones, J.C., Ed., Jones, M.B., Ed., Kumar, A., Ed., and E. Lundberg, Ed., "Web Authentication: An API for accessing Public Key Credentials - Level 2", W3C Recommendation, 8 April 2021, <https://www.w3.org/TR/2021/REC-webauthn-2-20210408/>.

Acknowledgements

The authors thank Mike Bishop, Carsten Bormann, Mohamed Boucadair, John Bradley, Tim Bray, Brian Campbell, Deb Cooley, Roman Danyliw, Stephen Farrell, Vijay Gurbani, Ilari Liusvaara, Tobias Looker, Neil Madden, Kathleen Moriarty, Jeremy O'Donoghue, John Preuß Mattsson, Anders Rundgren, Göran Selander, Filip Skokan, Oliver Terbu, Hannes Tschofenig, Sean Turner, Éric Vyncke, David Waite, Paul Wouters, and Jiankang Yao for their contributions to this specification.

Authors' Addresses

Michael B. Jones
Self-Issued Consulting
Email: michael_b_jones@hotmail.com
URI: https://self-issued.info/

Orie Steele
Tradeverifyd
Email: orie@or13.io