Stream: Internet Engineering Task Force (IETF)
RFC: 9679
Category: Standards Track
Published: December 2024
ISSN: 2070-1721
Authors: K. Isobe *SECOM CO., LTD.* H. Tschofenig *H-BRS* O. Steele *Transmute*

RFC 9679
CBOR Object Signing and Encryption (COSE) Key Thumbprint

Abstract

This specification defines a method for computing a hash value over a CBOR Object Signing and Encryption (COSE) Key. It specifies which fields within the COSE Key structure are included in the cryptographic hash computation, the process for creating a canonical representation of these fields, and how to hash the resulting byte sequence. The resulting hash value, referred to as a "thumbprint", can be used to identify or select the corresponding key.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9679.

Copyright Notice

Table of Contents

1. Introduction

This specification defines a method for applying a cryptographic hash function to a CBOR Object Signing and Encryption (COSE) Key structure [RFC9052], resulting in a hash value known as a "thumbprint". To achieve this, the document specifies which fields in the COSE Key structure are included in the hash computation, the process for creating a canonical form of these fields, and how to hash the resulting byte sequence. One of the primary use cases for this thumbprint is as a naming scheme for identifying or selecting the key, such as by using the COSE Key Thumbprint value as a "kid" (key ID). Another key use case involves key derivation functions that use the thumbprints of public keys from the endpoints, along with other application context, to derive a symmetric key.

This specification outlines how thumbprints of COSE Keys are generated for both asymmetric and symmetric keys (see Sections 3 and 4). Additionally, it introduces a new CBOR Web Token (CWT) confirmation method, which has been added to the IANA "CWT Confirmation Methods" registry established by [RFC8747]. For further details on the use of a confirmation claim in a CWT with a proof-of-possession key, refer to Section 3.1 of [RFC8747].

2. Terminology

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.

3. COSE Key Thumbprint

The thumbprint of a COSE Key **MUST** be computed as follows:

1. Construct a COSE_Key structure (see Section 7 of [RFC9052]) containing only the required parameters representing the key as described in Section 4 of this document.
2. Apply the deterministic encoding described in Section 4.2.1 of [RFC8949] to the representation constructed in step 1.
3. Hash the bytes produced in step 2 with a cryptographic hash function H. For example, SHA-256 [RFC6234] may be used as a hash function.

The details of this computation are further described in subsequent sections.

The SHA-256 hash algorithm **MUST** be supported; other algorithms **MAY** be supported.

4. Required COSE Key Parameters

Only the required parameters of a key's representation are used when computing its COSE Key Thumbprint value. This section summarizes the required parameters.

The "kty" (label: 1) element **MUST** be present for all key types, and the integer value specified in the IANA "COSE Key Types" registry **MUST** be used. The tstr data type is not used with the "kty" element.

Many COSE Key parameters are specific to the chosen key type. The following subsections list the required parameters for commonly used key types.

4.1. Octet Key Pair (OKP)

The required parameters for elliptic curve public keys that use the Octet Key Pair (OKP) key type, such as X25519, are:

- "kty" (label: 1, data type: int, value: 1)
- "crv" (label: -1, value: int)
- "x" (label: -2, value: bstr)

Further details are described in Section 7.1 of [RFC9053].

4.2. Elliptic Curve Keys with X- and Y-Coordinates

The required parameters for elliptic curve public keys that use the EC2 key type, such as NIST P-256, are:

- "kty" (label: 1, data type: int, value: 2)
- "crv" (label: -1, data type: int)
- "x" (label: -2, data type: bstr)
- "y" (label: -3, data type: bstr)

Further details are described in Section 7.1 of [RFC9053].

Note: [RFC9052] supports both compressed and uncompressed point representations. For interoperability, implementations adhering to this specification **MUST** use the uncompressed point representation. Therefore, the y-coordinate is expressed as a bstr. If an implementation uses the compressed point representation, it **MUST** first convert it to the uncompressed form for the purpose of thumbprint calculation.

4.3. RSA Public Keys

The required parameters for an RSA public key are:

- "kty" (label: 1, data type: int, value: 3)
- "n" (label: -1, data type: bstr)
- "e" (label: -2, data type: bstr)

4.4. Symmetric Keys

The required parameters for a symmetric key are:

- "kty" (label: 1, data type: int, value: 4)
- "k" (label: -1, data type: bstr)

4.5. HSS-LMS Keys

The required parameters for HSS-LMS keys are:

- "kty" (label: 1, data type: int, value: 5)
- "pub" (label: -1, data type: bstr)

4.6. Others

As other key type values are defined, their defining specifications should be similarly consulted to determine which parameters, in addition to the "kty" element, are required.

5. Miscellaneous Considerations

5.1. Why Not Include Optional COSE Key Parameters?

Optional parameters of COSE Keys are intentionally not included in the COSE Key Thumbprint computation so that their absence or presence in the COSE Key does not alter the resulting value. The COSE Key Thumbprint is a digest of the ordered essential parameters needed to represent a COSE Key, with all other parameters excluded.

By excluding optional parameters, the COSE Key Thumbprint consistently refers to the key itself, not to a key with additional attributes. Different application contexts may include various optional attributes in the COSE Key structure. If these optional parameters were included in the thumbprint calculation, the resulting values could differ for the same key depending on the attributes present. Including only the required parameters ensures that the COSE Key Thumbprint remains consistent for a given key, regardless of any additional attributes.

Different kinds of thumbprints could be defined by other specifications that might include some or all additional COSE Key parameters, if use cases arise where such different kinds of thumbprints would be useful.

5.2. Selection of Hash Function

A specific hash function must be chosen by an application to compute the hash value of the hash input. For instance, SHA-256 [RFC6234] may be used as the hash function. While SHA-256 is a good default choice at the time of writing, the preferred hash function may evolve as the cryptographic landscape develops.

In many cases, only the party that generates the key needs to be aware of the hash function used. For example, the key producer might use the thumbprint value as a "kid" (key ID). In such scenarios, the consumer of the "kid" treats it as an opaque value solely for key selection.

However, when multiple parties are involved in reproducing and comparing the COSE Key Thumbprint, it is crucial that they know and use the same hash function to ensure consistent results.

5.3. Thumbprints of Keys Not in COSE Key Format

Keys that are in other formats can be represented as COSE Keys. The only prerequisites are that the COSE_Key representation of the key be defined and the party creating the COSE Key Thumbprint be in possession of the necessary key material.

5.4. Relationship to Digests of X.509 Values

COSE Key Thumbprint values are computed on the COSE Key object containing only essential parameters in a specific order. Thus, they are more analogous to applications that use digests of X.509 Subject Public Key Info (SPKI) values, which are defined in Section 4.1.2.7 of [RFC5280], than to applications that use digests of complete certificate values, as the "x5t" (X.509 certificate SHA-1 thumbprint) [RFC9360] value defined for X.509 certificate objects does. While logically equivalent to a digest of the SPKI representation of the key, a COSE Key Thumbprint is computed over the CBOR representation of that key rather than over an ASN.1 representation of it.

5.5. Relationship to JSON Web Key Thumbprints

The ckt of a COSE Key, as described in Section 7 of [RFC9052], and the jkt of a JSON Web Key, as described in Section 4 of [RFC7517], are different even when the underlying cryptographic key material is the same.

This document does not register a JWT confirmation method [RFC7800] for using "ckt" as a confirmation method for a JWT or a CWT confirmation method [RFC8747] for using "jkt" as a confirmation method for a CWT.

5.6. Confirmation Method

[RFC8747] introduces confirmation methods for use with CWTs with the addition of the "cnf" claim. CWTs are defined in [RFC8392]. This specification adds a new confirmation method based on COSE Key Thumbprints.

The proof-of-possession key is identified using the "ckt" member of the CWT confirmation claim "cnf". This member contains the value of the COSE Key Thumbprint encoded as a binary string. Instead of communicating the actual COSE Key, only the thumbprint is conveyed. This approach assumes that the recipient is able to obtain the identified COSE Key using the thumbprint contained in the "ckt" member. In this approach, the issuer of a CWT declares that the presenter possesses a particular key and that the recipient can cryptographically confirm the presenter's proof of possession of the key by including a "ckt" CWT confirmation method member in the CWT.

The following example demonstrates the use of the "ckt" member in a CWT as part of the confirmation method (with line breaks inserted for editorial reasons):

```
   {
    /iss/ 1 : "coaps://as.example.com",
    /aud/ 3 : "coaps://resource.example.org",
    /exp/ 4 : 1361398824,
    /cnf/ 8 : {
      /ckt/ 5 : h'496bd8afadf307e5b08c64b0421bf9dc
                  01528a344a43bda88fadd1669da253ec'
     }
   }
```

Section 8 registers the "ckt" CWT confirmation method member. The "ckt" member is used in the "cnf" claim.

5.7. COSE Key Thumbprint URIs

This specification defines Uniform Resource Identifiers (URIs) to represent a COSE Key Thumbprint value. The design follows the work of JSON Web Key (JWK) Thumbprint URIs, as specified in [RFC9278]. This enables COSE Key Thumbprints to be used, for example, as key identifiers in contexts requiring URIs. This specification defines a URI prefix indicating that the portion of the URI following the prefix is a COSE Key Thumbprint.

The following URI prefix is defined to indicate that the portion of the URI following the prefix is a COSE Key Thumbprint:

urn:ietf:params:oauth:ckt

To make the hash algorithm being used explicit in a URI, the prefix is followed by a hash algorithm identifier and a COSE Key Thumbprint value, each separated by a colon character to form a URI representing a COSE Key Thumbprint.

Hash algorithm identifiers used in COSE Key Thumbprint URIs **MUST** be values from the "Hash Name String" column in the IANA "Named Information Hash Algorithm Registry" [IANA.Hash.Algorithms]. COSE Key Thumbprint URIs with hash algorithm identifiers not found in this registry are not considered valid, and applications **MUST** detect and handle this error, should it occur.

Since the URN is encoded as a string, the output of the COSE Key Thumbprint computation described in Section 3 **MUST** be base64url encoded without padding.

[RFC7515] specifies base64url encoding as follows:

> Base64 encoding using the URL- and filename-safe character set defined in Section 5 of RFC 4648 [RFC4648], with all trailing '=' characters omitted (as permitted by Section 3.2 of [RFC7515]) and without the inclusion of any line breaks, whitespace, or other additional characters. Note that the base64url encoding of the empty octet sequence is

the empty string. (See Appendix C of [RFC7515] for notes on implementing base64url encoding without padding.)

The base64url encoding of the thumbprint shown in Section 6 is shown below (with a line break added for readability purposes).

SWvYr63zB-WwjGSwQhv53AFSijRKQ72oj63RZp2iU-w

The full example of a COSE Key Thumbprint URI is shown below (with a line break added for readability).

urn:ietf:params:oauth:ckt:sha-256:

SWvYr63zB-WwjGSwQhv53AFSijRKQ72oj63RZp2iU-w

Note that the use of oauth in the namespace is to align with JWK Thumbprint URIs as described in [RFC9278]; however, these URIs are intended for use with applications and specifications not necessarily related to OAuth.

6. Example

This section demonstrates the COSE Key Thumbprint computation for the following example COSE Key containing an Elliptic Curve Cryptography (ECC) public key.

For better readability, the example is first presented in CBOR diagnostic format (with the long line broken for display purposes only).

```
  {
    / kty set to EC2 = Elliptic Curve Keys /
    1:2,
    / crv set to P-256 /
    -1:1,
    / public key: x-coordinate /
    -2:h'65eda5a12577c2bae829437fe338701a10aaa375e1bb5b5de108de439c0
8551d',
    / public key: y-coordinate /
    -3:h'1e52ed75701163f7f9e40ddf9f341b3dc9ba860af7e0ca7ca7e9eecd008
4d19c',
    / kid is bstr, not used in COSE Key Thumbprint /
    2:h'496bd8afadf307e5b08c64b0421bf9dc01528a344a43bda88fadd1669da2
53ec'
  }
```

The example above corresponds to the following CBOR encoding (with link breaks added for display purposes only):

```
A50102200121582065EDA5A12577C2BAE829437FE338701A10AAA375E1BB5B5DE108D
E439C08551D2258201E52ED75701163F7F9E40DDF9F341B3DC9BA860AF7E0CA7CA7E9
EECD0084D19C025820496BD8AFADF307E5B08C64B0421BF9DC01528A344A43BDA88FA
DD1669DA253EC
```

Not all of the parameters from the example above are used in the COSE Key Thumbprint computation because the required parameters of an elliptic curve public key are (as listed in Section 4.2) "kty", "crv", "x", and "y".

The resulting COSE Key structure, in CBOR diagnostic format with line breaks added for better readability, with the minimum parameters in the correct order are:

```
{
   1:2,
  -1:1,
  -2:h'65eda5a12577c2bae829437fe338701a
        10aaa375e1bb5b5de108de439c08551d',
  -3:h'1e52ed75701163f7f9e40ddf9f341b3d
        c9ba860af7e0ca7ca7e9eecd0084d19c'
}
```

In CBOR encoding, the result is (with line breaks added for display purposes only):

```
A40102200121582065EDA5A12577C2BAE829437FE338701A10AAA375E1BB5B5DE
108DE439C08551D2258201E52ED75701163F7F9E40DDF9F341B3DC9BA860AF7E0
CA7CA7E9EECD0084D19C
```

Using SHA-256, the resulting thumbprint is:

```
496bd8afadf307e5b08c64b0421bf9dc01528a344a43bda88fadd1669da253ec
```

7. Security Considerations

A COSE Key Thumbprint will only uniquely identify a particular key if a single unambiguous COSE Key representation for that key is defined and used when computing the COSE Key Thumbprint. Key identifiers are not included in the thumbprint calculation (similarly to other optional parameters in the COSE_Key structure). If the inclusion of specific optional parameters in the thumbprint calculation is important for a particular application, this specification would not be suitable.

While thumbprint values are useful for identifying legitimate keys, comparing thumbprint values is not a reliable means of excluding the use of particular keys (or transformations thereof). The reason is because an attacker may supply a key that is a transformation of a key in order for it to appear as a different key. For instance, if a legitimate RSA key uses a modulus value N and an attacker supplies a key with modulus 3*N, the modified key would still work about 1/3 of the time, but it would appear to be a different key.

Producing thumbprints of symmetric keys needs to be done with care. Developers **MUST** ensure that the symmetric key has sufficient entropy to prevent attackers from precomputing tables of symmetric keys with their corresponding hash values. This can be prevented if the symmetric key is a randomly selected key of at least a 128-bit length. Thumbprints **MUST NOT** be used with passwords or other low-entropy secrets. If a developer is unable to determine whether all symmetric keys used in an application have sufficient entropy, then thumbprints of symmetric keys **MUST NOT** be used. In general, using thumbprints of symmetric keys should only be used in special applications. In most other deployment scenarios, it is more appropriate to utilize a different naming scheme for key identifiers.

8. IANA Considerations

IANA has added the following entry to the "CWT Confirmation Methods" registry [IANA-CWT] established by [RFC8747]:

Confirmation Method Name: ckt
Confirmation Method Description: COSE Key SHA-256 Thumbprint
JWT Confirmation Method Name: (none)
Confirmation Key: 5
Confirmation Value Type(s): binary string
Change Controller: IETF
Specification Document(s): RFC 9679

Furthermore, IANA has added a value to the "OAuth URI" registry [IANA-OAuth] established by [RFC6755]:

URN: urn:ietf:params:oauth:ckt
Common Name: COSE Key Thumbprint URI
Change Controller: IETF
Specification Document(s): RFC 9679

9. References

9.1. 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>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC6755] Campbell, B. and H. Tschofenig, "An IETF URN Sub-Namespace for OAuth", RFC 6755, DOI 10.17487/RFC6755, October 2012, <https://www.rfc-editor.org/info/rfc6755>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[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>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[RFC8747] Jones, M., Seitz, L., Selander, G., Erdtman, S., and H. Tschofenig, "Proof-of-Possession Key Semantics for CBOR Web Tokens (CWTs)", RFC 8747, DOI 10.17487/RFC8747, March 2020, <https://www.rfc-editor.org/info/rfc8747>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[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>.

9.2. Informative References

[IANA-CWT] IANA, "CWT Confirmation Methods", <https://www.iana.org/assignments/cwt>.

[IANA-OAuth] IANA, "OAuth URI", <https://www.iana.org/assignments/oauth-parameters>.

[IANA.Hash.Algorithms] IANA, "Named Information Hash Algorithm Registry", <https://www.iana.org/assignments/named-information>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[RFC7638] Jones, M. and N. Sakimura, "JSON Web Key (JWK) Thumbprint", RFC 7638, DOI 10.17487/RFC7638, September 2015, <https://www.rfc-editor.org/info/rfc7638>.

[RFC7800] Jones, M., Bradley, J., and H. Tschofenig, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", RFC 7800, DOI 10.17487/RFC7800, April 2016, <https://www.rfc-editor.org/info/rfc7800>.

[RFC9278] Jones, M. and K. Yasuda, "JWK Thumbprint URI", RFC 9278, DOI 10.17487/RFC9278, August 2022, <https://www.rfc-editor.org/info/rfc9278>.

[RFC9360] Schaad, J., "CBOR Object Signing and Encryption (COSE): Header Parameters for Carrying and Referencing X.509 Certificates", RFC 9360, DOI 10.17487/RFC9360, February 2023, <https://www.rfc-editor.org/info/rfc9360>.

Acknowledgements

We would like to thank the authors of [RFC7638] for their work on the JWK Thumbprint specification. This document applies JWK Thumbprints to COSE Key structures.

Additionally, we would like to thank Carsten Bormann, Ilari Liusvaara, Laurence Lundblade, Daisuke Ajitomi, Michael Richardson, Michael B. Jones, Mallory Knodel, Joel Jaeggli, Derrell Piper, Patrik Fältström, Warren Kumari, Deb Cooley, and Brendan Moran for their feedback.

Authors' Addresses

Kohei Isobe
SECOM CO., LTD.
Japan
Email: isobekohei@gmail.com

Hannes Tschofenig
University of Applied Sciences Bonn-Rhein-Sieg
Germany
Email: hannes.tschofenig@gmx.net

Orie Steele
Transmute
United States of America
Email: orie@transmute.industries