Stream: Internet Engineering Task Force (IETF)
RFC: 8747
Category: Standards Track
Published: March 2020
ISSN: 2070-1721

Authors:
M. Jones, *Microsoft*
L. Seitz, *Combitech*
G. Selander, *Ericsson AB*
S. Erdtman, *Spotify*
H. Tschofenig, *Arm Ltd.*

RFC 8747
Proof-of-Possession Key Semantics for CBOR Web Tokens (CWTs)

Abstract

This specification describes how to declare in a CBOR Web Token (CWT) (which is defined by RFC 8392) that the presenter of the CWT possesses a particular proof-of-possession key. Being able to prove possession of a key is also sometimes described as being the holder-of-key. This specification provides equivalent functionality to "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)" (RFC 7800) but using Concise Binary Object Representation (CBOR) and CWTs rather than JavaScript Object Notation (JSON) and JSON Web Tokens (JWTs).

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/rfc8747.

Copyright Notice

Table of Contents

1. Introduction

2. Terminology

3. Representations for Proof-of-Possession Keys

  3.1. Confirmation Claim

  3.2. Representation of an Asymmetric Proof-of-Possession Key

  3.3. Representation of an Encrypted Symmetric Proof-of-Possession Key

  3.4. Representation of a Key ID for a Proof-of-Possession Key

  3.5. Specifics Intentionally Not Specified

4. Security Considerations

5. Privacy Considerations

6. Operational Considerations

7. IANA Considerations

  7.1. CBOR Web Token Claims Registration

    7.1.1. Registry Contents

  7.2. CWT Confirmation Methods Registry

    7.2.1. Registration Template

    7.2.2. Initial Registry Contents

8. References

  8.1. Normative References

  8.2. Informative References

Acknowledgements

Authors' Addresses

1. Introduction

This specification describes how a CBOR Web Token (CWT) [RFC8392] can declare that the presenter of the CWT possesses a particular proof-of-possession (PoP) key. Proof of possession of a key is also sometimes described as being the holder-of-key. This specification provides equivalent functionality to "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)" [RFC7800] but using Concise Binary Object Representation (CBOR) [RFC7049] and CWTs [RFC8392] rather than JavaScript Object Notation (JSON) [RFC8259] and JSON Web Tokens (JWTs) [JWT].

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.

This specification uses terms defined in the CBOR Web Token (CWT) [RFC8392], CBOR Object Signing and Encryption (COSE) [RFC8152], and Concise Binary Object Representation (CBOR) [RFC7049] specifications.

These terms are defined by this specification:

Issuer

Party that creates the CWT and binds the claims about the subject to the proof-of-possession key.

Presenter

Party that proves possession of a private key (for asymmetric key cryptography) or secret key (for symmetric key cryptography) to a recipient of a CWT.

In the context of OAuth, this party is also called the OAuth Client.

Recipient

Party that receives the CWT containing the proof-of-possession key information from the presenter.

In the context of OAuth, this party is also called the OAuth Resource Server.

This specification provides examples in CBOR extended diagnostic notation, as defined in Appendix G of [RFC8610]. The examples include line breaks for readability.

3. Representations for Proof-of-Possession Keys

By including a `cnf` (confirmation) claim in a CWT, the issuer of the CWT declares that the presenter possesses a particular key and that the recipient can cryptographically confirm that the presenter has possession of that key. The value of the `cnf` claim is a CBOR map (which is defined in Section 2.1 of [RFC7049]) and the members of that map identify the proof-of-possession key.

The presenter can be identified in one of several ways by the CWT, depending upon the application requirements. For instance, some applications may use the CWT `sub` (subject) claim [RFC8392] to identify the presenter. Other applications may use the `iss` (issuer) claim [RFC8392] to identify the presenter. In some applications, the subject identifier might be relative to the issuer identified by the `iss` claim. The actual mechanism used is dependent upon the application. The case in which the presenter is the subject of the CWT is analogous to Security Assertion Markup Language (SAML) 2.0 [OASIS.saml-core-2.0-os] SubjectConfirmation usage.

3.1. Confirmation Claim

The `cnf` claim in the CWT is used to carry confirmation methods. Some of them use proof-of-possession keys, while others do not. This design is analogous to the SAML 2.0 [OASIS.saml-core-2.0-os] SubjectConfirmation element in which a number of different subject confirmation methods can be included (including proof-of-possession key information).

The set of confirmation members that a CWT must contain to be considered valid is context dependent and is outside the scope of this specification. Specific applications of CWTs will require implementations to understand and process some confirmation members in particular ways. However, in the absence of such requirements, all confirmation members that are not understood by implementations **MUST** be ignored.

Section 7.2 establishes the IANA "CWT Confirmation Methods" registry for CWT `cnf` member values and registers the members defined by this specification. Other specifications can register other members used for confirmation, including other members for conveying proof-of-possession keys using different key representations.

The `cnf` claim value **MUST** represent only a single proof-of-possession key. At most one of the `COSE_Key` and `Encrypted_COSE_Key` confirmation values defined in Table 1 may be present. Note that if an application needs to represent multiple proof-of-possession keys in the same CWT, one way for it to achieve this is to use other claim names (in addition to `cnf`) to hold the additional proof-of-possession key information. These claims could use the same syntax and semantics as the `cnf` claim. Those claims would be defined by applications or other specifications and could be registered in the IANA "CBOR Web Token (CWT) Claims" registry [IANA.CWT.Claims].

Name	Key	Value type
COSE_Key	1	COSE_Key

Name	Key	Value type
Encrypted_COSE_Key	2	COSE_Encrypt or COSE_Encrypt0
kid	3	binary string

Table 1: Summary of the `cnf` Names, Keys, and Value Types

3.2. Representation of an Asymmetric Proof-of-Possession Key

When the key held by the presenter is an asymmetric private key, the `COSE_Key` member is a COSE_Key [RFC8152] representing the corresponding asymmetric public key. The following example demonstrates such a declaration in the CWT Claims Set of a CWT:

```
 {
 /iss/ 1 : "coaps://server.example.com",
 /aud/ 3 : "coaps://client.example.org",
 /exp/ 4 : 1879067471,
 /cnf/ 8 :{
   /COSE_Key/ 1 :{
     /kty/ 1 : /EC2/ 2,
     /crv/ -1 : /P-256/ 1,
     /x/ -2 : h'd7cc072de2205bdc1537a543d53c60a6acb62eccd890c7fa27c9
                e354089bbe13',
     /y/ -3 : h'f95e1d4b851a2cc80fff87d8e23f22afb725d535e515d020731e
                79a3b4e47120'
    }
  }
}
```

The COSE_Key **MUST** contain the required key members for a COSE_Key of that key type and **MAY** contain other COSE_Key members, including the `kid` (Key ID) member.

The `COSE_Key` member **MAY** also be used for a COSE_Key representing a symmetric key, provided that the CWT is encrypted so that the key is not revealed to unintended parties. The means of encrypting a CWT is explained in [RFC8392]. If the CWT is not encrypted, the symmetric key **MUST** be encrypted as described in Section 3.3. This procedure is equivalent to the one defined in Section 3.3 of [RFC7800].

3.3. Representation of an Encrypted Symmetric Proof-of-Possession Key

When the key held by the presenter is a symmetric key, the `Encrypted_COSE_Key` member is an encrypted COSE_Key [RFC8152] representing the symmetric key encrypted to a key known to the recipient using COSE_Encrypt or COSE_Encrypt0.

The following example illustrates a symmetric key that could subsequently be encrypted for use in the `Encrypted_COSE_Key` member:

```
{
 /kty/ 1 : /Symmetric/ 4,
 /alg/ 3 : /HMAC 256-256/ 5,
 /k/ -1 : h'6684523ab17337f173500e5728c628547cb37df
            e68449c65f885d1b73b49eae1'
}
```

The COSE_Key representation is used as the plaintext when encrypting the key.

The following example CWT Claims Set of a CWT illustrates the use of an encrypted symmetric key as the `Encrypted_COSE_Key` member value:

```
{
 /iss/ 1 : "coaps://server.example.com",
 /sub/ 2 : "24400320",
 /aud/ 3: "s6BhdRkqt3",
 /exp/ 4 : 1311281970,
 /iat/ 5 : 1311280970,
 /cnf/ 8 : {
 /Encrypted_COSE_Key/ 2 : [
    /protected header/ h'A1010A' /{ \alg\ 1:10 \AES-CCM-16-64-128\}/,
    /unprotected header/ { / iv / 5: h'636898994FF0EC7BFCF6D3F95B'},
    /ciphertext/  h'0573318A3573EB983E55A7C2F06CADD0796C9E584F1D0E3E
                    A8C5B052592A8B2694BE9654F0431F38D5BBC8049FA7F13F'
   ]
  }
}
```

The example above was generated with the key:

```
h'6162630405060708090a0b0c0d0e0f10'
```

3.4. Representation of a Key ID for a Proof-of-Possession Key

The proof-of-possession key can also be identified using a Key ID instead of communicating the actual key, provided the recipient is able to obtain the identified key using the Key ID. In this case, 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 `cnf` claim in the CWT whose value is a CBOR map containing a `kid` member identifying the key.

The following example demonstrates such a declaration in the CWT Claims Set of a CWT:

```
{
 /iss/ 1 : "coaps://as.example.com",
 /aud/ 3 : "coaps://resource.example.org",
 /exp/ 4 : 1361398824,
 /cnf/ 8 : {
   /kid/ 3 : h'dfd1aa976d8d4575a0fe34b96de2bfad'
  }
}
```

The content of the `kid` value is application specific. For instance, some applications may choose to use a cryptographic hash of the public key value as the `kid` value.

Note that the use of a Key ID to identify a proof-of-possession key needs to be carefully circumscribed, as described below and in Section 6. In cases where the Key ID is not a cryptographic value derived from the key or where not all of the parties involved are validating the cryptographic derivation, implementers should expect collisions where different keys are assigned the same Key ID. Recipients of a CWT with a PoP key linked through only a Key ID should be prepared to handle such situations.

In the world of constrained Internet of Things (IoT) devices, there is frequently a restriction on the size of Key IDs, either because of table constraints or a desire to keep message sizes small.

Note that the value of a Key ID for a specific key is not necessarily the same for different parties. When sending a COSE encrypted message with a shared key, the Key ID may be different on both sides of the conversation, with the appropriate one being included in the message based on the recipient of the message.

3.5. Specifics Intentionally Not Specified

Proof of possession is often demonstrated by having the presenter sign a value determined by the recipient using the key possessed by the presenter. This value is sometimes called a "nonce" or a "challenge". There are, however, also other means to demonstrate freshness of the exchange and to link the proof-of-possession key to the participating parties, as demonstrated by various authentication and key exchange protocols.

The means of communicating the nonce and the nature of its contents are intentionally not described in this specification, as different protocols will communicate this information in different ways. Likewise, the means of communicating the signed nonce is also not specified, as this is also protocol specific.

Note that other means of proving possession of the key exist, which could be used in conjunction with a CWT's confirmation key. Applications making use of such alternate means are encouraged to register them in the IANA "CBOR Web Token (CWT) Confirmation Methods" registry established in Section 7.2.

4. Security Considerations

All the security considerations that are discussed in [RFC8392] also apply here. In addition, proof of possession introduces its own unique security issues. Possessing a key is only valuable if it is kept secret. Appropriate means must be used to ensure that unintended parties do not learn private key or symmetric key values.

Applications utilizing proof of possession **SHOULD** also utilize audience restriction, as described in Section 3.1.3 of [RFC8392], because it provides additional protections. Audience restriction can be used by recipients to reject messages intended for different recipients. (Of course, applications not using proof of possession can also benefit from using audience restriction to reject messages intended for different recipients.)

CBOR Web Tokens with proof-of-possession keys are used in context of an architecture, such as the ACE OAuth Framework [ACE-OAUTH], in which protocols are used by a presenter to request these tokens and to subsequently use them with recipients. Proof of possession only provides the intended security gains when the proof is known to be current and not subject to replay attacks; security protocols using mechanisms such as nonces and timestamps can be used to avoid the risk of replay when performing proof of possession for a token. Note that a discussion of the architecture or specific protocols that CWTs with proof-of-possession keys are used with is beyond the scope of this specification.

As is the case with other information included in a CWT, it is necessary to apply data origin authentication and integrity protection (via a keyed message digest or a digital signature). Data origin authentication ensures that the recipient of the CWT learns about the entity that created the CWT, since this will be important for any policy decisions. Integrity protection prevents an adversary from changing any elements conveyed within the CWT payload. Special care has to be applied when carrying symmetric keys inside the CWT since those not only require integrity protection but also confidentiality protection.

As described in Section 6 (Key Identification) and Appendix D (Notes on Key Selection) of [JWS], it is important to make explicit trust decisions about the keys. Proof-of-possession signatures made with keys not meeting the application's trust criteria **MUST NOT** be relied upon.

5. Privacy Considerations

A proof-of-possession key can be used as a correlation handle if the same key is used on multiple occasions. Thus, for privacy reasons, it is recommended that different proof-of-possession keys be used when interacting with different parties.

6. Operational Considerations

The use of CWTs with proof-of-possession keys requires additional information to be shared between the involved parties in order to ensure correct processing. The recipient needs to be able to use credentials to verify the authenticity and integrity of the CWT. Furthermore, the recipient may need to be able to decrypt either the whole CWT or the encrypted parts thereof (see Section 3.3). This requires the recipient to know information about the issuer. Likewise, there needs to be agreement between the issuer and the recipient about the claims being used (which is also true of CWTs in general).

When an issuer creates a CWT containing a Key ID claim, it needs to make sure that it does not issue another CWT with different claims containing the same Key ID within the lifetime of the CWTs, unless intentionally desired. Failure to do so may allow one party to impersonate another party, with the potential to gain additional privileges. A case where such reuse of a Key ID would be intentional is when a presenter obtains a CWT with different claims (e.g., extended scope) for the same recipient but wants to continue using an existing security association (e.g., a DTLS session) bound to the key identified by the Key ID. Likewise, if PoP keys are used for multiple different kinds of CWTs in an application and the PoP keys are identified by Key IDs, care must be taken to keep the keys for the different kinds of CWTs segregated so that an attacker cannot cause the wrong PoP key to be used by using a valid Key ID for the wrong kind of CWT. Using an audience restriction for the CWT would be one strategy to mitigate this risk.

7. IANA Considerations

The following registration procedure is used for all the registries established by this specification.

Values are registered on a Specification Required [RFC8126] basis after a three-week review period on the <cwt-reg-review@ietf.org> mailing list, on the advice of one or more designated experts. However, to allow for the allocation of values prior to publication, the designated experts may approve registration once they are satisfied that such a specification will be published.

Registration requests sent to the mailing list for review should use an appropriate subject (e.g., "Request to Register CWT Confirmation Method: example"). Registration requests that are undetermined for a period longer than 21 days can be brought directly to IANA's attention (using the iana@iana.org mailing list) for resolution.

Designated experts should determine whether a registration request contains enough information for the registry to be populated with the new values and whether the proposed new functionality already exists. In the case of an incomplete registration or an attempt to register already existing functionality, the designated experts should ask for corrections or reject the registration.

It is suggested that multiple designated experts be appointed who are able to represent the perspectives of different applications using this specification in order to enable broadly informed review of registration decisions. In cases where a registration decision could be perceived as creating a conflict of interest for a particular expert, that expert should defer to the judgment of the other experts.

7.1. CBOR Web Token Claims Registration

This specification registers the `cnf` claim in the IANA "CBOR Web Token (CWT) Claims" registry [IANA.CWT.Claims], established by [RFC8392].

7.1.1. Registry Contents

- Claim Name: `cnf`
- Claim Description: Confirmation
- JWT Claim Name: `cnf`
- Claim Key: 8
- Claim Value Type(s): map
- Change Controller: IESG
- Specification Document(s): Section 3.1 of RFC 8747

7.2. CWT Confirmation Methods Registry

This specification establishes the IANA "CWT Confirmation Methods" registry for CWT `cnf` member values. The registry records the confirmation method member and a reference to the specification that defines it.

7.2.1. Registration Template

Confirmation Method Name:
  The human-readable name requested (e.g., "kid").

Confirmation Method Description:
  Brief description of the confirmation method (e.g., "Key Identifier").

JWT Confirmation Method Name:
  Claim Name of the equivalent JWT confirmation method value, as registered in the "JSON Web Token Claims" subregistry in the "JSON Web Token (JWT)" registry [IANA.JWT]. CWT claims should normally have a corresponding JWT claim. If a corresponding JWT claim would not make sense, the designated experts can choose to accept registrations for which the JWT Claim Name is listed as "N/A".

Confirmation Key:
  CBOR map key value for the confirmation method.

Confirmation Value Type(s):
  CBOR types that can be used for the confirmation method value.

Change Controller:
  For Standards Track RFCs, list the "IESG". For others, give the name of the responsible party.

Specification Document(s):

Reference to the document or documents that specify the parameter, preferably including URIs that can be used to retrieve copies of the documents. An indication of the relevant sections may also be included but is not required. Note that the designated experts and IANA must be able to obtain copies of the specification document(s) to perform their work.

7.2.2. Initial Registry Contents

- Confirmation Method Name: `COSE_Key`
- Confirmation Method Description: COSE_Key Representing Public Key
- JWT Confirmation Method Name: `jwk`
- Confirmation Key: 1
- Confirmation Value Type(s): COSE_Key structure
- Change Controller: IESG
- Specification Document(s): Section 3.2 of RFC 8747

- Confirmation Method Name: `Encrypted_COSE_Key`
- Confirmation Method Description: Encrypted COSE_Key
- JWT Confirmation Method Name: `jwe`
- Confirmation Key: 2
- Confirmation Value Type(s): COSE_Encrypt or COSE_Encrypt0 structure (with an optional corresponding COSE_Encrypt or COSE_Encrypt0 tag)
- Change Controller: IESG
- Specification Document(s): Section 3.3 of RFC 8747

- Confirmation Method Name: `kid`
- Confirmation Method Description: Key Identifier
- JWT Confirmation Method Name: `kid`
- Confirmation Key: 3
- Confirmation Value Type(s): binary string
- Change Controller: IESG
- Specification Document(s): Section 3.4 of RFC 8747

8. References

8.1. Normative References

[IANA.CWT.Claims] IANA, "CBOR Web Token Claims", , <http://www.iana.org/assignments/cwt>.

[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>.

[RFC7049]

Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[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>.

[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>.

8.2. Informative References

[ACE-OAUTH] Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", Work in Progress, Internet-Draft, draft-ietf-ace-oauth-authz-21, 14 February 2019, <https://tools.ietf.org/html/draft-ietf-ace-oauth-authz-21>.

[IANA.JWT] IANA, "JSON Web Token (JWT)", , <http://www.iana.org/assignments/jwt>.

[JWS] 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>.

[JWT] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[OASIS.saml-core-2.0-os] Cantor, S., Kemp, J., Philpott, R., and E. Maler, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-core-2.0-os, March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object

Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/info/rfc8610>.

Acknowledgements

Thanks to the following people for their reviews of the specification: Roman Danyliw, Christer Holmberg, Benjamin Kaduk, Mirja Kühlewind, Yoav Nir, Michael Richardson, Adam Roach, Éric Vyncke, and Jim Schaad.

Ludwig Seitz and Göran Selander worked on this document as part of the CelticPlus projects CyberWI and CRITISEC, with funding from Vinnova.

Authors' Addresses

Michael B. Jones
Microsoft
Email: mbj@microsoft.com
URI: https://self-issued.info/

Ludwig Seitz
Combitech
Djaeknegatan 31
211 35 Malmö
Sweden
Email: ludwig.seitz@combitech.se

Göran Selander
Ericsson AB
164 80 Kista
Sweden
Email: goran.selander@ericsson.com

Samuel Erdtman
Spotify
Email: erdtman@spotify.com

Hannes Tschofenig
Arm Ltd.
6060 Hall in Tirol
Austria
Email: Hannes.Tschofenig@arm.com