ACE
Internet-Draft
Intended status: Standards Track
Expires: May 13, 2019

M. Jones
Microsoft
L. Seitz
RISE SICS
G. Selander
Ericsson AB
S. Erdtman
Spotify
H. Tschofenig
ARM Ltd.
November 9, 2018

Proof-of-Possession Key Semantics for CBOR Web Tokens (CWTs)
draft-ietf-ace-cwt-proof-of-possession-05

Abstract

This specification describes how to declare in a CBOR Web Token (CWT) 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 CBOR and CWTs rather than JSON and JWTs.

Table of Contents

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 CBOR [RFC7049] and CWTs [RFC8392] rather than JSON [RFC7159] and 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.
In context of OAuth this party is also called OAuth Client.

Recipient
: Party that receives the CWT containing the proof-of-possession key information from the presenter.
In context of OAuth this party is also called OAuth Resource Server.

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 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" claim to identify the presenter. In some applications, the subject identifier might be relative to the issuer identified by the "iss" (issuer) claim [RFC8392]. 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.

This specification establishes the IANA "CWT Confirmation Methods" registry for these members in Section 7.2 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 Figure 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 Claims" registry [IANA.CWT.Claims].

Name	Key	Value type
COSE_Key	1	COSE_Key
Encrypted_COSE_Key	2	COSE_Encrypt or COSE_Encrypt0
kid	3	binary string

Figure 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 (using CBOR diagnostic notation) 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 : 1361398824,
  /cnf/ 8 :{
    /COSE_Key/ 1 :{
      /kty/ 1 : /EC/ 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.

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 (using CBOR diagnostic notation, with linebreaks for readability) illustrates a symmetric key that could subsequently be encrypted for use in the "Encrypted_COSE_Key" member:

```
     {
      /kty/ 1 : /Symmetric/ 4,
      /alg/ 3 : /HMAC256/ 5,
      /k/ -1 : h'6684523ab17337f173500e5728c628547cb37df
                 e68449c65f885d1b73b49eae1A0B0C0D0E0F10'
     }
```

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

The following example CWT Claims Set of a CWT (using CBOR diagnostic notation, with linebreaks for readability) 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 : {
   /COSE_Encrypt0/ 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 by the use of 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 proof of possession of the key by the presenter by including a "cnf" claim in the CWT whose value is a CBOR map with the CBOR map containing a "kid" member identifying the key.

The following example (using CBOR diagnostic notation) 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 : 1361398824,
 /cnf/ 8 : {
   /kid/ 2 : 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-possesion key needs to be carefully circumscribed, as described below and in Section 6. Where the Key ID is not a cryptographic value derived from the key or where all of the parties involved are not validating the cryptographic derivation, it is possible to get into situations where the same Key ID is being used for multiple keys. The implication of this is that a recipient may have multiple keys known to it that have the same Key ID, and thus it might not know which proof-of-possession key is associated with the CWT.

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. These restrictions are going to protocol dependent. For example, DTLS can use a Key ID of any size. However, if the key is being used with COSE encrypted message, then the length of the key needs to be minimized and may have a limit as small as one byte.

Note that the value of a Key ID is not always 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.

For symmetric keys, the Key ID is normally going to be generated by the CWT issuer. This means that enforcing a rule that Key ID values only match if CWTs have the same issuer works for matching Key IDs between CWTs. In this case, the issuer can ensure that there are no collisions between currently active symmetric keys for all CWTs that it has issued. This allows for a recipient to use the pair of issuer and Key ID for matching keys.

For asymmetric keys, the Key ID value is normally going to be generated by the CWT recipient, thus the possibility of collisions is greater. For instance, recipients might start by assigning a Key ID

of 0, given that Key IDs are frequently only needed to be unique and meaningful to the recipient. This problem can be addressed in a couple of different ways, depending on how the Key ID value is going to be used:

- The issuer can assign a new unique Key ID the first time it sees the key. Depending on the protocol being used, the new value may then need to be transported to the presenter by the protocol used to issue CWTs. In this case, the rule of requiring that the issuer, Key ID pair be used for matching works.

- The issuer can use a different confirmation method if a collision might be unavoidable.

- A recipient can decide not to use a CWT based on a created Key ID if it does not fit the recipient's requirements.

- If an issuer is going to use the Key ID confirmation method and is not going to guarantee that serial number uniqueness is going to be preserved, the recipient needs to have that information configured into it so that appropriate actions can be taken.

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".

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 another means of proving possession of the key when it is a symmetric key is to encrypt the key to the recipient. The means of obtaining a key for the recipient is likewise protocol specific.

4. Security Considerations

All of 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 4.1.3 of [JWT], as it provides additional protections. Audience restriction can be used by recipients to reject messages intended for different recipients.

A recipient might not understand the "cnf" claim. Applications that require the proof-of-possession keys communicated with it to be understood and processed MUST ensure that the parts of this specification that they use are implemented.

CBOR Web Tokens with proof-of-possession keys are used in context of an architecture, such as the ACE OAuth Framework [I-D.ietf-ace-oauth-authz], in which protocols are used by a presenter to request these tokens and to subsequently use them with recipients. To avoid replay attacks when the proof-of-possession tokens are sent to presenters, a security protocol, which uses mechansims such as nonces or timestamps, has to be utilized. Note that a discussion of the architecture or specific protocols that CWT proof-of-possession tokens 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 with multiple parties. 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, integrity, and potentially the confidentiality of the CWT and its content. 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 containing the same Key ID with a different content, or for a different subject, 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. 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.

7. IANA Considerations

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

Values are registered on a Specification Required [RFC5226] 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. [[Note to the RFC Editor: The name of the mailing list should be determined in consultation with the IESG and IANA. Suggested name: cwt-reg-review@ietf.org.]]

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 to the IESG's attention (using the iesg@ietf.org mailing list) for resolution.

Criteria that should be applied by the Designated Experts include determining whether the proposed registration duplicates existing functionality, determining whether it is likely to be of general applicability or whether it is useful only for a single application,

and evaluating the security properties of the item being registered and whether the registration makes sense.

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 Claims" registry [IANA.CWT.Claims] established by [RFC8392].

7.1.1. Registry Contents

- o Claim Name: "cnf"
- o Claim Description: Confirmation
- o JWT Claim Name: "cnf"
- o Claim Key: TBD (maybe 8)
- o Claim Value Type(s): map
- o Change Controller: IESG
- o Specification Document(s): Section 3.1 of [[this document]]

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 [IANA.JWT.Claims]. 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. Other details (e.g., postal address, email address, home page URI) may also be included.

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.

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 [[this document]]

- 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 [[this document]]

- 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 [[this document]]

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>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[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>.

[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

[I-D.ietf-ace-oauth-authz]
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)", draft-ietf-ace-oauth-authz-16 (work in progress), October 2018.

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

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

[JWT]   Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7159, May 2015, <http://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/>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[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>.

Acknowledgements

Thanks to the following people for their reviews of the specification: Roman Danyliw, Michael Richardson, and Jim Schaad.

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

Document History

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

-05

- Added text suggested by Jim Schaad describing considerations when using the Key ID confirmation method.

-04

- Addressed additional WGLC comments by Jim Schaad and Roman Danyliw.

-03

- Addressed review comments by Jim Schaad, see https://www.ietf.org/mail-archive/web/ace/current/msg02798.html

- Removed unnecessary sentence in the introduction regarding the use any strings that could be case-sensitive.

- Clarified the terms Presenter and Recipient.

- Clarified text about the confirmation claim.

-02

- Changed "typically" to "often" when describing ways of performing proof of possession.

- Changed b64 to hex encoding in an example.

- Changed to using the RFC 8174 boilerplate instead of the RFC 2119 boilerplate.

-01

- Now uses CBOR diagnostic notation for the examples.

- Added a table summarizing the "cnf" names, keys, and value types.

- Addressed some of Jim Schaad's feedback on -00.

-00

- Created the initial working group draft from draft-jones-ace-cwt-proof-of-possession-01.

Authors' Addresses

Michael B. Jones
Microsoft

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

Ludwig Seitz
RISE SICS
Scheelevaegen 17
Lund 223 70
Sweden

Email: ludwig@ri.se

Goeran Selander
Ericsson AB
Faeroegatan 6
Kista 164 80
Sweden

Email: goran.selander@ericsson.com

Samuel Erdtman
Spotify

Email: erdtman@spotify.com

Hannes Tschofenig
ARM Ltd.
Hall in Tirol 6060
Austria

Email: Hannes.Tschofenig@arm.com