ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 15, 2017

M. Jones
Microsoft
E. Wahlstroem

S. Erdtman
Spotify AB
H. Tschofenig
ARM Ltd.

April 13, 2017

CBOR Web Token (CWT)
draft-ietf-ace-cbor-web-token-04

Abstract

CBOR Web Token (CWT) is a compact means of representing claims to be transferred between two parties. CWT is a profile of the JSON Web Token (JWT) that is optimized for constrained devices. The claims in a CWT are encoded in the Concise Binary Object Representation (CBOR) and CBOR Object Signing and Encryption (COSE) is used for added application layer security protection. A claim is a piece of information asserted about a subject and is represented as a name/value pair consisting of a claim name and a claim value.

Table of Contents

1. Introduction

The JSON Web Token (JWT) [RFC7519] is a standardized security token format that has found use in OAuth 2.0 and OpenID Connect deployments, among other applications. JWT uses JSON Web Signature (JWS) [RFC7515] and JSON Web Encryption (JWE) [RFC7516] to secure the contents of the JWT, which is a set of claims represented in JSON. The use of JSON for encoding information is popular for Web and native applications, but it is considered inefficient for some Internet of Things (IoT) systems that use low power radio technologies.

An alternative encoding of claims is defined in this document. Instead of using JSON, as provided by JWTs, this specification uses CBOR [RFC7049] and calls this new structure "CBOR Web Token (CWT)", which is a compact means of representing secured claims to be transferred between two parties. CWT is closely related to JWT. It references the JWT claims and both its name and pronunciation are derived from JWT. To protect the claims contained in CWTs, the CBOR Object Signing and Encryption (COSE) [I-D.ietf-cose-msg] specification is used.

The suggested pronunciation of CWT is the same as the English word "cot".

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 "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

This document reuses terminology from JWT [RFC7519] and COSE [I-D.ietf-cose-msg].

StringOrURI:
: The "StringOrURI" term has the same meaning, syntax, and processing rules as the "StringOrUri" term defined in Section 2 of

JWT [RFC7519], except that a CWT StringOrURI uses CBOR major type 3 (text string) instead of a JSON string value.

NumericDate:
The "NumericDate" term has the same meaning, syntax, and processing rules as the "NumericDate" term defined in Section 2 of JWT [RFC7519], except that a CWT NumericDate uses one of the CBOR numeric types (0, 1, or 7 with subtypes 25, 26, or 27), instead of a numeric JSON value. The numeric date values that can used for a CWT NumericDate are identical to the epoch-based date/time values that are specified to follow the tag defined in Section 2.4.1 (Date and Time) of [RFC7049], except that the tag itself need not be present.

CBOR encoded claim key:
The key used to identify a claim value.

CWT Claims Set
A CBOR map that contains the claims conveyed by the CWT.

3. Claims

The set of claims 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 claims in particular ways. However, in the absence of such requirements, all claims that are not understood by implementations MUST be ignored.

To keep CWTs as small as possible, the CBOR encoded claim keys are represented using CBOR major type 0. Section 4 summarizes all keys used to identify the claims defined in this document.

3.1. Claim Names

None of the claims defined below are intended to be mandatory to use or implement. They rather provide a starting point for a set of useful, interoperable claims. Applications using CWTs should define which specific claims they use and when they are required or optional.

3.1.1. iss (Issuer) Claim

The "iss" (issuer) claim has the same meaning, syntax, and processing rules as the "iss" claim defined in Section 4.1.1 of JWT [RFC7519], except that the format MUST be a StringOrURI. The CBOR encoded claim key 1 MUST be used to identify this claim.

3.1.2. sub (Subject) Claim

The "sub" (subject) claim has the same meaning, syntax, and processing rules as the "sub" claim defined in Section 4.1.2 of JWT [RFC7519], except that the format MUST be a StringOrURI. The CBOR encoded claim key 2 MUST be used to identify this claim.

3.1.3. aud (Audience) Claim

The "aud" (audience) claim has the same meaning, syntax, and processing rules as the "aud" claim defined in Section 4.1.3 of JWT [RFC7519], except that the format MUST be a StringOrURI. The CBOR encoded claim key 3 MUST be used to identify this claim.

3.1.4. exp (Expiration Time) Claim

The "exp" (expiration time) claim has the same meaning, syntax, and processing rules as the "exp" claim defined in Section 4.1.4 of JWT [RFC7519], except that the format MUST be a CWT NumericDate. The CBOR encoded claim key 4 MUST be used to identify this claim.

3.1.5. nbf (Not Before) Claim

The "nbf" (not before) claim has the same meaning, syntax, and processing rules as the "nbf" claim defined in Section 4.1.5 of JWT [RFC7519], except that the format MUST be a CWT NumericDate. The CBOR encoded claim key 5 MUST be used to identify this claim.

3.1.6. iat (Issued At) Claim

The "iat" (issued at) claim has the same meaning, syntax, and processing rules as the "iat" claim defined in Section 4.1.6 of JWT [RFC7519], except that the format MUST be a CWT NumericDate. The CBOR encoded claim key 6 MUST be used to identify this claim.

3.1.7. cti (CWT ID) Claim

The "cti" (CWT ID) claim has the same meaning, syntax, and processing rules as the "jti" claim defined in Section 4.1.7 of JWT [RFC7519], except that the format MUST be of major type 2, binary string. The CBOR encoded claim key 7 MUST be used to identify this claim.

4. Summary of the values, CBOR major types and encoded claim keys

Claim	CBOR encoded claim key	CBOR major type of value
iss	1	3
sub	2	3
aud	3	3
exp	4	0, 1, or 7 with float subtype
nbf	5	0, 1, or 7 with float subtype
iat	6	0, 1, or 7 with float subtype
cti	7	2

Figure 1: Summary of the values, CBOR major types and encoded claim keys.

5. CBOR Tags and Claim Values

The use of CBOR tags to prefix any of the claim values defined in this specification is NOT RECOMMENDED. For instance, while CBOR tag 6.1 (seconds-since-the-epoch) could logically be prefixed to values of the "exp", "nbf", and "iat" claims, this is unnecessary, since the representation of the claim values is already specified by the claim definitions. Tagging claim values would only take up extra space, without adding information. However, other claims defined by other specifications can specify that a tag prefix the claim value, when appropriate.

6. CWT CBOR Tag

How to determine that a CBOR data structure is a CWT is application-dependent. In some cases, this information is known from the application context, such as from the position of the CWT in a data structure at which the value must be a CWT. One method of indicating that a CBOR object is a CWT is the use of the "application/cwt" content type by a transport protocol.

This section defines the CWT CBOR tag as another means for applications to declare that a CBOR data structure is a CWT. Its use is optional, and is intended for use in cases in which this information would not otherwise be known.

If present, the CWT tag MUST prefix a tagged object using one of the COSE CBOR tags. In this example, the COSE_Mac0 tag is used. The actual COSE_Mac0 object has been excluded from this example.

```
/ CWT CBOR tag / 61(
  / COSE_Mac0 CBOR tag / 17(
    / COSE_Mac0 object /
  )
)
```

Figure 2: Example of a CWT tag usage

7. Creating and Validating CWTs

7.1. Creating a CWT

To create a CWT, the following steps are performed. The order of the steps is not significant in cases where there are no dependencies between the inputs and outputs of the steps.

1. Create a CWT Claims Set containing the desired claims.

2. Let the Message be the binary representation of the CWT Claims Set.

3. Create a COSE Header containing the desired set of Header Parameters. The COSE Header MUST be valid per the [I-D.ietf-cose-msg] specification.

4. Depending upon whether the CWT is signed, MACed, or encrypted, there are three cases:

   * If the CWT is signed, create a COSE_Sign/COSE_Sign1 object using the Message as the COSE_Sign/COSE_Sign1 Payload; all steps specified in [I-D.ietf-cose-msg] for creating a COSE_Sign/COSE_Sign1 object MUST be followed.

   * Else, if the CWT is MACed, create a COSE_Mac/COSE_Mac0 object using the Message as the COSE_Mac/COSE_Mac0 Payload; all steps specified in [I-D.ietf-cose-msg] for creating a COSE_Mac/ COSE_Mac0 object MUST be followed.

   * Else, if the CWT is a COSE_Encrypt/COSE_Encrypt0 object, create a COSE_Encrypt/COSE_Encrypt0 using the Message as the plaintext for the COSE_Encrypt/COSE_Encrypt0 object; all steps specified in [I-D.ietf-cose-msg] for creating a COSE_Encrypt/ COSE_Encrypt0 object MUST be followed.

5. If a nested signing, MACing or encryption operation will be performed, let the Message be the COSE_Sign/COSE_Sign1, COSE_Mac/ COSE_Mac0 or COSE_Encrypt/COSE_Encrypt0, and return to Step 3, using a "content type" header value corresponding to the media

type "application/cwt" in the new COSE Header created in that step.

6. If needed by the application, add the appropriate COSE CBOR tag to the COSE object to indicate type of COSE object. If also needed by the application, add the CWT CBOR tag to indicate that the COSE object is a CWT.

7.2. Validating a CWT

When validating a CWT, the following steps are performed. The order of the steps is not significant in cases where there are no dependencies between the inputs and outputs of the steps. If any of the listed steps fail, then the CWT MUST be rejected -- that is, treated by the application as invalid input.

1. Verify that the CWT is a valid CBOR object.

2. If the object begins with the CWT CBOR tag, remove it and verify that one of the COSE CBOR tags follows it.

3. If the object is tagged with one of the COSE CBOR tags, remove it and verify that it corresponds to the structure of the following COSE object.

4. Verify that the resulting COSE Header includes only parameters and values whose syntax and semantics are both understood and supported or that are specified as being ignored when not understood.

5. Use the CBOR tag to determine the type of the CWT, COSE_Sign/ COSE_Sign1, COSE_Mac/COSE_Mac0, or COSE_Encrypt/COSE_Encrypt0.

6. Depending upon whether the CWT is a signed, MACed, or encrypted, there are three cases:

   * If the CWT is a COSE_Sign/COSE_Sign1, follow the steps specified in [I-D.ietf-cose-msg] Section 4 (Signing Objects) for validating a COSE_Sign/COSE_Sign1 object. Let the Message be the COSE_Sign/COSE_Sign1 payload.

   * Else, if the CWT is a COSE_Mac/COSE_Mac0, follow the steps specified in [I-D.ietf-cose-msg] Section 6 (MAC Objects) for validating a COSE_Mac/COSE_Mac0 object. Let the Message be the COSE_Mac/COSE_Mac0 payload.

   * Else, if the CWT is a COSE_Encrypt/COSE_Encrypt0 object, follow the steps specified in [I-D.ietf-cose-msg] Section 5

(Encryption Objects) for validating a COSE_Encrypt/ COSE_Encrypt0 object. Let the Message be the resulting plaintext.

7. If the COSE Header contains a "content type" header value corresponding to the media type "application/cwt", then the Message is a CWT that was the subject of nested signing or encryption operations. In this case, return to Step 1, using the Message as the CWT.

8. Verify that the Message is a valid CBOR object; let the CWT Claims Set be this CBOR object.

8. Security Considerations

The security of the CWT is dependent on the protections offered by COSE. Unless the claims in a CWT are protected, an adversary can modify, add, or remove claims. Since the claims conveyed in a CWT may be used to make authorization decisions, it is not only important to protect the CWT in transit but also to ensure that the recipient can authenticate the party that assembled the claims and created the CWT. Without trust of the recipient in the party that created the CWT, no sensible authorization decision can be made. Furthermore, the creator of the CWT needs to carefully evaluate each claim value prior to including it in the CWT so that the recipient can be assured of the validity of the information provided.

9. IANA Considerations

9.1. CBOR Web Token (CWT) Claims Registry

This section establishes the IANA "CBOR Web Token (CWT) Claims" registry.

Values are registered on a Specification Required [RFC5226] basis, 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.

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

9.1.1. Registration Template

Claim Name:
The human-readable name requested (e.g., "iss").

Claim Description:
Brief description of the claim (e.g., "Issuer").

JWT Claim Name:
Claim Name of the equivalent JWT claim, 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".

CBOR Key Value:
Key value for the claim. The key value MUST be an integer in the range of 1 to 65536.

CBOR Major Type:
CBOR major type and optional tag for the claim.

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.

9.1.2. Initial Registry Contents

- Claim Name: "iss"
- Claim Description: Issuer
- JWT Claim Name: "iss"
- CBOR Key Value: 1
- CBOR Major Type: 3
- Change Controller: IESG
- Specification Document(s): Section 3.1.1 of [[this specification]]

- Claim Name: "sub"
- Claim Description: Subject
- JWT Claim Name: "sub"
- CBOR Key Value: 2

- CBOR Major Type: 3
- Change Controller: IESG
- Specification Document(s): Section 3.1.2 of [[this specification]]

- Claim Name: "aud"
- Claim Description: Audience
- JWT Claim Name: "aud"
- CBOR Key Value: 3
- CBOR Major Type: 3
- Change Controller: IESG
- Specification Document(s): Section 3.1.3 of [[this specification]]

- Claim Name: "exp"
- Claim Description: Expiration Time
- JWT Claim Name: "exp"
- CBOR Key Value: 4
- CBOR Major Type: 0, 1, or 7 with subtypes 25, 26, or 27
- Change Controller: IESG
- Specification Document(s): Section 3.1.4 of [[this specification]]

- Claim Name: "nbf"
- Claim Description: Not Before
- JWT Claim Name: "nbf"
- CBOR Key Value: 5
- CBOR Major Type: 0, 1, or 7 with subtypes 25, 26, or 27
- Change Controller: IESG
- Specification Document(s): Section 3.1.5 of [[this specification]]

- Claim Name: "iat"
- Claim Description: Issued At
- JWT Claim Name: "iat"
- CBOR Key Value: 6
- CBOR Major Type: 0, 1, or 7 with subtypes 25, 26, or 27
- Change Controller: IESG
- Specification Document(s): Section 3.1.6 of [[this specification]]

- Claim Name: "cti"
- Claim Description: CWT ID
- JWT Claim Name: "jti"
- CBOR Key Value: 7
- CBOR Major Type: 2
- Change Controller: IESG

- Specification Document(s): Section 3.1.7 of [[this specification]]

9.2. Media Type Registration

This section registers the "application/cwt" media type in the "Media Types" registry [IANA.MediaTypes] in the manner described in RFC 6838 [RFC6838], which can be used to indicate that the content is a CWT.

9.2.1. Registry Contents

- Type name: application
- Subtype name: cwt
- Required parameters: N/A
- Optional parameters: N/A
- Encoding considerations: binary
- Security considerations: See the Security Considerations section of [[this specification]]
- Interoperability considerations: N/A
- Published specification: [[this specification]]
- Applications that use this media type: IoT applications sending security tokens over HTTP(S) and other transports.
- Fragment identifier considerations: N/A
- Additional information:

  Magic number(s): N/A
  File extension(s): N/A
  Macintosh file type code(s): N/A

- Person & email address to contact for further information: IESG, iesg@ietf.org
- Intended usage: COMMON
- Restrictions on usage: none
- Author: Michael B. Jones, mbj@microsoft.com
- Change controller: IESG
- Provisional registration? No

9.3. CoAP Content-Formats Registration

This section registers the CoAP Content-Format ID for the "application/cwt" media type in the "CoAP Content-Formats" registry [IANA.CoAP.Content-Formats].

9.3.1. Registry Contents

- Media Type: application/cwt
- Encoding: -
- Id: TBD (maybe 61)

- Reference: [[this specification]]

9.4. CBOR Tag registration

This section registers the CWT CBOR tag in the "CBOR Tags" registry [IANA.CBOR.Tags].

9.4.1. Registry Contents

- CBOR Tag: TBD (maybe 61 to use the same value as the Content-Format)
- Data Item: CBOR Web Token (CWT)
- Semantics: CBOR Web Token (CWT), as defined in [[this specification]]
- Reference: [[this specification]]
- Point of Contact: Michael B. Jones, mbj@microsoft.com

10. References

10.1. Normative References

[I-D.ietf-cose-msg]
Schaad, J., "CBOR Object Signing and Encryption (COSE)", draft-ietf-cose-msg-24 (work in progress), November 2016.

[IANA.CBOR.Tags]
IANA, "Concise Binary Object Representation (CBOR) Tags", <http://www.iana.org/assignments/cbor-tags/cbor-tags.xhtml>.

[IANA.CoAP.Content-Formats]
IANA, "CoAP Content-Formats", <http://www.iana.org/assignments/core-parameters/core-parameters.xhtml#content-formats>.

[IANA.MediaTypes]
IANA, "Media Types", <http://www.iana.org/assignments/media-types>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://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, <http://www.rfc-editor.org/info/rfc7049>.

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

10.2. Informative References

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

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

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

Appendix A. Examples

This appendix includes a set of CWT examples that show how the CWT Claims Set can be protected. There are examples that are signed, MACed, encrypted, and that use nested signing and encryption. To make the examples easier to read, they are presented both as hex strings and in the extended CBOR diagnostic notation described in Section 6 of [RFC7049].

A.1. Example CWT Claims Set

The CWT Claims Set used for the different examples displays usage of all the defined claims. For signed and MACed examples, the CWT Claims Set is the CBOR encoding as a binary string.

```
a70175636f61703a2f2f61732e6578616d706c652e636f6d02656572696b7703
7818636f61703a2f2f6c696768742e6578616d706c652e636f6d041a5612aeb0
051a5610d9f0061a5610d9f007420b71
```

Figure 3: Example CWT Claims Set as hex string

```
{
  / iss / 1: "coap://as.example.com",
  / sub / 2: "erikw",
  / aud / 3: "coap://light.example.com",
  / exp / 4: 1444064944,
  / nbf / 5: 1443944944,
  / iat / 6: 1443944944,
  / cti / 7: h'0b71'
}
```

Figure 4: Example CWT Claims Set in CBOR diagnostic notation

A.2. Example keys

This section contains the keys used to sign, MAC, and encrypt the messages in this appendix. Line breaks are for display purposes only.

A.2.1. 128-bit Symmetric Key as Hex Encoded String

```
8e82e68e61654ecb5a369fe8be7572dd
```

A.2.2. 256-bit Symmetric Key as Hex Encoded String

```
403697de87af64611c1d32a05dab0fe1fcb715a86ab435f1ec99192d79569388
```

A.2.3. ECDSA P-256 256-bit COSE Key

```
a622582060f7f1a780d8a783bfb7a2dd6b2796e8128dbbcef9d3d168db952997
1a36e7b92358206c1382765aec5358f117733d281c1c7bdc39884d04a45a1e6c
67c858bc206c1903260102215820143329cce7868e416927599cf65a34f3ce2f
fda55a7eca69ed8919a394d42f0f2001
```

Figure 5: ECDSA 256-bit COSE Key as hex string

```
{
  / d /   -4: h'6c1382765aec5358f117733d281c1c7bdc39884d04a45a1e
               6c67c858bc206c19',
  / y /   -3: h'60f7f1a780d8a783bfb7a2dd6b2796e8128dbbcef9d3d168
               db9529971a36e7b9',
  / x /   -2: h'143329cce7868e416927599cf65a34f3ce2ffda55a7eca69
               ed8919a394d42f0f',
  / crv / -1: 1 / P-256 / ,
  / kty /  1: 2 / EC2 / ,
  / alg /  3: -7 / ECDSA 256 /
}
```

Figure 6: ECDSA 256-bit COSE Key in CBOR diagnostic notation

A.3. Example Signed CWT

This section shows a signed CWT with a single recipient and a full CWT Claims Set.

The signature is generated using the private key listed in Appendix A.2.3 and it can be validated using the public key from Appendix A.2.3. Line breaks are for display purposes only.

```
d28443a10126a05850a70175636f61703a2f2f61732e6578616d706c652e636f6
d02656572696b77037818636f61703a2f2f6c696768742e6578616d706c652e63
6f6d041a5612aeb0051a5610d9f0061a5610d9f007420b7158401fe410abce650
effed497f05d7f9462de67d571384097de0d96f1e2514d284cdd85634f269af6c
36c64f22e7691abb464bed2ff23176cdba9fd9e213f637d082
```

Figure 7: Signed CWT as hex string

```
18(
  [
    / protected / h'a10126' / {
      / alg / 1: -7 / ECDSA 256 /
    } / ,
    / unprotected / {},
    / payload / h'a70175636f61703a2f2f61732e6578616d706c652e63
                  6f6d02656572696b77037818636f61703a2f2f6c6967
                  68742e6578616d706c652e636f6d041a5612aeb0051a
                  5610d9f0061a5610d9f007420b71' / {
        / iss / 1: "coap://as.example.com",
        / sub / 2: "erikw",
        / aud / 3: "coap://light.example.com",
        / exp / 4: 1444064944,
        / nbf / 5: 1443944944,
        / iat / 6: 1443944944,
        / cti / 7: h'0b71'
      } / ,
    / signature / h'1fe410abce650effed497f05d7f9462de67d571384
                    097de0d96f1e2514d284cdd85634f269af6c36c64f
                    22e7691abb464bed2ff23176cdba9fd9e213f637d0
                    82'
  ]
)
```

Figure 8: Signed CWT in CBOR diagnostic notation

A.4. Example MACed CWT

This section shows a MACed CWT with a single recipient and a full CWT Claims Set.

The MAC is generated using the 256-bit symmetric key from Appendix A.2.2 with a 64-bit truncation. Line breaks are for display purposes only.

```
d83dd18443a10104a05850a70175636f61703a2f2f61732e6578616d706c652e
636f6d02656572696b77037818636f61703a2f2f6c696768742e6578616d706c
652e636f6d041a5612aeb0051a5610d9f0061a5610d9f007420b7148093101ef
6d789200
```

Figure 9: MACed CWT with CWT tag as hex string

```
61(
  17(
    [
      / protected / h'a10104' / {
        / alg / 1: 4 / HMAC 256/64 /
      } / ,
      / unprotected / {},
      / payload / h'a70175636f61703a2f2f61732e6578616d706c652e636f
                    6d02656572696b77037818636f61703a2f2f6c69676874
                    2e6578616d706c652e636f6d041a5612aeb0051a5610d9
                    f0061a5610d9f007420b71' / {
          / iss / 1: "coap://as.example.com",
          / sub / 2: "erikw",
          / aud / 3: "coap://light.example.com",
          / exp / 4: 1444064944,
          / nbf / 5: 1443944944,
          / iat / 6: 1443944944,
          / cti / 7: h'0b71'
        } / ,
      / tag / h'093101ef6d789200'
    ]
  )
)
```

Figure 10: MACed CWT with CWT tag in CBOR diagnostic notation

A.5. Example Encrypted CWT

This section shows an encrypted CWT with a single recipient and a full CWT Claims Set.

The encryption is done with AES-CCM mode using the 128-bit symmetric key from Appendix A.2.1 with a 64-bit tag and 13-byte nonce, i.e., COSE AES-CCM-16-64-128. Line breaks are for display purposes only.

```
d08343a1010aa1054d3a869e378e72b77d077c29be025858d275ad9cd7df1b10
ba8cde785c74b1e1e6ada287e2baf1451b06862529b784d230b0111773b6c369
1319aec4dcc379fe47115a5d62632727c05f4567fc84dd79554db86676a14978
42de805d8be93180af4d6ff3043886a0
```

Figure 11: Encrypted CWT as hex string

```
16(
  [
    / protected / h'a1010a' / {
      / alg / 1: 10 / AES-CCM-16-64-128 /
    } /,
    / unprotected / {
      / iv / 5: h'3a869e378e72b77d077c29be02'
    },
    / ciphertext / h'd275ad9cd7df1b10ba8cde785c74b1e1e6ada287e2b
                     af1451b06862529b784d230b0111773b6c3691319ae
                     c4dcc379fe47115a5d62632727c05f4567fc84dd795
                     54db86676a1497842de805d8be93180af4d6ff30438
                     86a0'
  ]
)
```

Figure 12: Encrypted CWT in CBOR diagnostic notation

A.6. Example Nested CWT

This section shows a Nested CWT, signed and then encrypted, with a single recipient and a full CWT Claims Set.

The signature is generated using the private ECDSA key from Appendix A.2.3 and it can be validated using the public ECDSA parts from Appendix A.2.3. The encryption is done with AES-CCM mode using the 128-bit symmetric key from Appendix A.2.1 with a 64-bit tag and 13-byte nonce, i.e., COSE AES-CCM-16-64-128. The content type is set to CWT to indicate that there are multiple layers of COSE protection before finding the CWT Claims Set. The decrypted ciphertext will be a COSE_sign1 structure. In this example, it is the same one as in Appendix A.3, i.e., a Signed CWT Claims Set. Note that there is no limitation to the number of layers; this is an example with two layers. Line breaks are for display purposes only.

```
d08346a203183d010aa1054d9120e5dc42c9f9aec05ebe8a4858a538be026c02
4a40b19d6dbea3ddb18b31021f874a097a05ff3cdaa4665bafc8e46a3d7f37ad
f002fe57eee267f8f62a9c1621af75e1ecd742a3d801c2cc82358cf104a8d902
4d38a599ea6027d482dc2948a88fe83f9734804299c832401029e2d32a984789
c8e9563e8d2a751323bb7e4462b549e0fa89ef93f78bf6425635fba76b4aa804
7908e89b3b7c3d59d8a80e22f70a1b6ee8c162c564341c2f15cec252d3da038c
```

Figure 13: Signed and Encrypted CWT as hex string

```
16(
  [
    / protected / h'a203183d010a' / {
      / content type / 3: 61, / CWT /
      / alg / 1: 10 / AES-CCM-16-64-128 /
    } / ,
    / unprotected / {
      / iv / 5: h'9120e5dc42c9f9aec05ebe8a48'
    },
    / ciphertext / h'38be026c024a40b19d6dbea3ddb18b31021f874a097
                     a05ff3cdaa4665bafc8e46a3d7f37adf002fe57eee2
                     67f8f62a9c1621af75e1ecd742a3d801c2cc82358cf
                     104a8d9024d38a599ea6027d482dc2948a88fe83f97
                     34804299c832401029e2d32a984789c8e9563e8d2a7
                     51323bb7e4462b549e0fa89ef93f78bf6425635fba7
                     6b4aa8047908e89b3b7c3d59d8a80e22f70a1b6ee8c
                     162c564341c2f15cec252d3da038c'
  ]
)
```

Figure 14: Signed and Encrypted CWT in CBOR diagnostic notation

Appendix B. Acknowledgements

This specification is based on JSON Web Token (JWT) [RFC7519], the authors of which also include Nat Sakimura and John Bradley. Ludwig Seitz and Goeran Selander have made contributions the specification.

Appendix C. Document History

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

-04

- Specified that the use of CBOR tags to prefix any of the claim values defined in this specification is NOT RECOMMENDED.

-03

- Reworked the examples to include signed, MACed, encrypted, and nested CWTs.
- Defined the CWT CBOR tag and explained its usage.

-02

- Added IANA registration for the application/cwt media type.
- Clarified the nested CWT language.
- Corrected nits identified by Ludwig Seitz.

-01

- Added IANA registration for CWT Claims.
- Added IANA registration for the application/cwt CoAP content-format type.
- Added Samuel Erdtman as an editor.
- Changed Erik's e-mail address.

-00

- Created the initial working group version based on draft-wahlstroem-ace-cbor-web-token-00.

Authors' Addresses

Michael B. Jones
Microsoft

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

Erik Wahlstroem
Sweden

Email: erik@wahlstromstekniska.se

Samuel Erdtman
Spotify AB
Birger Jarlsgatan 61, 4tr
Stockholm 113 56
Sweden

Phone: +46702691499
Email: erdtman@spotify.com

Hannes Tschofenig
ARM Ltd.
Hall in Tirol 6060
Austria

Email: Hannes.Tschofenig@arm.com