

OAuth Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 28, 2014

M. Jones  
Microsoft  
J. Bradley  
Ping Identity  
H. Tschofenig  
ARM Limited  
June 26, 2014

Proof-Of-Possession Semantics for JSON Web Tokens (JWTs)  
draft-jones-oauth-proof-of-possession-01.txt

## Abstract

This specification defines how to express a declaration in a JSON Web Token (JWT) that the presenter of the JWT possesses a particular key and that the recipient can cryptographically confirm proof-of-possession of the key by the presenter. This property is also sometimes described as the presenter being a holder-of-key.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 28, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Terminology</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Proof-Of-Possession Representation</a>	<a href="#">3</a>
<a href="#">3.1.</a>	<a href="#">Proof-of-Possession of an Asymmetric Key</a>	<a href="#">3</a>
<a href="#">3.2.</a>	<a href="#">Proof-of-Possession of a Symmetric Key</a>	<a href="#">4</a>
<a href="#">3.3.</a>	<a href="#">Confirmation</a>	<a href="#">5</a>
<a href="#">3.4.</a>	<a href="#">Specifics Intentionally Not Specified</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">Security Considerations</a>	<a href="#">6</a>
<a href="#">5.</a>	<a href="#">IANA Considerations</a>	<a href="#">7</a>
<a href="#">5.1.</a>	<a href="#">JSON Web Token Claims Registration</a>	<a href="#">8</a>
<a href="#">5.1.1.</a>	<a href="#">Registry Contents</a>	<a href="#">8</a>
<a href="#">5.2.</a>	<a href="#">JWT Confirmation Methods Registry</a>	<a href="#">8</a>
<a href="#">5.2.1.</a>	<a href="#">Registration Template</a>	<a href="#">8</a>
<a href="#">5.2.2.</a>	<a href="#">Initial Registry Contents</a>	<a href="#">9</a>
<a href="#">6.</a>	<a href="#">Acknowledgements</a>	<a href="#">9</a>
<a href="#">7.</a>	<a href="#">References</a>	<a href="#">9</a>
<a href="#">7.1.</a>	<a href="#">Normative References</a>	<a href="#">9</a>
<a href="#">7.2.</a>	<a href="#">Informative References</a>	<a href="#">9</a>
<a href="#">Appendix A.</a>	<a href="#">Document History</a>	<a href="#">10</a>
	<a href="#">Authors' Addresses</a>	<a href="#">10</a>

## [1.](#) Introduction

This specification defines how to express a declaration in a JSON Web Token (JWT) [[I-D.ietf-oauth-json-web-token](#)] that the presenter of the JWT possesses a particular key and that the recipient can cryptographically confirm proof-of-possession of the key by the presenter. This property is also sometimes described as the presenter being a holder-of-key.

Envision the following use case. An OAuth 2.0 authorization server generates a JWT and places a symmetric key inside the newly introduced confirmation claim. This symmetric key is encrypted with a key known only to the authorization server and the recipient. The JWT is then sent to the presenter. Since the presenter is unable to obtain the encrypted symmetric key the authorization server conveys that symmetric key separately to the presenter. Now, the presenter

is in possession of the symmetric key as well as the JWT (which includes the confirmation claim element). When the presenter needs to utilize the JWT to a recipient it also needs to demonstrate possession of the symmetric key; the presenter, for example, uses the symmetric key in a challenge / response protocol with the recipient.

The recipient is able to verify that it indeed interacts with the genuine presenter by decrypting the JWK contained inside the confirmation claim of the JWT. By doing this the recipient obtains the symmetric key, which it then uses to verify a cryptographically protected messages exchanged with the presenter.

## [2.](#) Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Unless otherwise noted, all the protocol parameter names and values are case sensitive.

This specification uses terms defined in the JSON Web Token (JWT) [[I-D.ietf-oauth-json-web-token](#)], JSON Web Key (JWK) [[I-D.ietf-jose-json-web-key](#)], and JSON Web Encryption (JWE) [[I-D.ietf-jose-json-web-encryption](#)] specifications.

## [3.](#) Proof-Of-Possession Representation

The presenter of a JWT declares that it possesses a particular key and that the recipient can cryptographically confirm proof-of-possession of the key by the issuer by including a "cnf" (confirmation) claim in the JWT whose value is a JSON object, with the JSON object containing a "jwk" (JSON Web Key) member identifying the key.

The presenter can be identified in one of two ways by the JWT, depending upon the application requirements. If the JWT contains a "sub" (subject) claim, the presenter is the subject identified by the JWT. (In some applications, the subject identifier will be relative to the issuer identified by the "iss" (issuer) claim.) If the JWT contains no "sub" (subject) claim, the presenter is the issuer identified by the JWT using the "iss" (issuer) claim. The case in

which the presenter is the subject of the JWT is analogous to SAML 2.0 [[OASIS.saml-core-2.0-os](#)] SubjectConfirmation usage. At least one of the "sub" and "iss" claims MUST be present in the JWT and in some use cases both MUST be present. In cases where the presenter shall be anonymous only the "iss" (issuer) claim may be present identifying the party that issued the JWT.

### [3.1.](#) Proof-of-Possession of an Asymmetric Key

When the key held by the issuer is an asymmetric key pair, the value of the "jwk" member is a JSON Web Key (JWK) [[I-D.ietf-jose-json-web-key](#)] representing the public key. The

Jones, et al.

Expires December 28, 2014

[Page 3]

---

Internet-Draft

proof-of-possession for JWTs

June 2014

example shown in Figure 1 demonstrates such a declaration in the JWT Claims Set of a JWT:

```
{
  "iss": "xas.example.com",
  "aud": "http://auth.example.com",
  "exp": "1361398824",
  "nbf": "1360189224",
  "cnf": {
    "jwk": {
      "kid": "pk1"
      "kty": "EC",
      "use": "sig",
      "crv": "P-256",
      "x": "18wHLeIgW9wVN6VD1Txgpqy2LszYkMf6J8njVAibvhM",
      "y": "-V4dS4UaLMgP_4fY4j8ir7cl1TXlFdAgcx55o7TkcSA"
    }
  }
}
```

Figure 1: JWT with a Confirmation Claim containing a Public Key

The JWK MUST contain the required elements of a JWK (as needed for the particular type) and MAY contain other JWK elements. It is highly recommended to include the "kid" (key ID) element in the JWK as well since it allows the presenter and the recipient to later use this key id to refer to the exchanged key.

### [3.2.](#) Proof-of-Possession of a Symmetric Key

When the key held by the issuer is a symmetric key, the value of the "jwk" member is an encrypted JSON Web Key (JWK) [[I-D.ietf-jose-json-web-key](#)] encrypted with a key known to the recipient. The rules for encrypting a JWK are found in [Section 6](#) of the JSON Web Key [[I-D.ietf-jose-json-web-key](#)] specification. Note that the JWE compact serialization is used.

The example shown in Figure 2 illustrates a symmetric key that is subsequently encrypted for use in the "jwk" member:

```
{
  "kid": "sessionkey-1"
  "kty": "oct",
  "alg": "HS256",
  "k": "ZoRSOrFzN_FzUA5XKMYoVHyzzff5oRJxl-IXRtztJ6uE"
}
```

Figure 2: JWK with a Symmetric Key

The UTF-8 [[RFC3629](#)] encoding of the JWK shown in Figure 2 would be used as the JWE Plaintext.

An example JWE header is shown in Figure 3. It provides the necessary information for encrypting the JWK.

```
{
  "kid": "longterm-12345"
  "alg": "RSA1_5",
  "enc": "A128CBC-HS256",
  "cty": "jwk+json",
}
```

Figure 3: Header of the JWE with Information about the Encryption of the Symmetric Key

The example in Figure 4 illustrates a JWT with the encrypted symmetric key as the "jwk" claim value:

```
{
  "iss": "https://server.example.com",
  "sub": "24400320",
```

```

    "aud": "s6BhdRkqt3",
    "nonce": "n-0S6_WzA2Mj",
    "exp": 1311281970,
    "iat": 1311280970,
    "cnf": {
      "jwk":
        "eyJhbGciOiJSU0ExXzUiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaY3R5IjoiaWoiandrk2pzb24ifQ. ... (remainder of JWE omitted for brevity)"
    }
  }
}

```

Figure 4: JWT with a Confirmation Claim containing the Encrypted Symmetric Key

Note that the case in which the "jwk" claim contains an unencoded JWK value and the case in which it contains an encrypted JWK value can be distinguished by the type of the member value. In the first case, the value is a JSON object containing the JWK and in the second case, the value is a string containing the JWE JSON Serialization of the encrypted JWK representation.

### [3.3.](#) Confirmation

The "cnf" (confirmation) claim is used in the JWT to contain the "jwk" element because a proof-of-possession key may not be the only means of confirming the authenticity of the token. This is analogous

---

Jones, et al. Expires December 28, 2014 [Page 5]

---

Internet-Draft proof-of-possession for JWTs June 2014

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. When a recipient receives a "cnf" claim with a member that it does not understand, it MUST ignore that member.

This specification defines a registry for these elements in [Section 5.2](#) and registers the "jwk" member within the registry.

### [3.4.](#) Specifics Intentionally Not Specified

Proof-of-possession is typically demonstrated by having the issuer sign a value determined by the recipient using the key possessed by the issuer. 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.

For a protocol that applies the mechanisms described in this document to the OAuth 2.0 context please take a look at [\[I-D.hunt-oauth-pop-architecture\]](#).

#### 4. Security Considerations

All of the normal security issues, especially in relationship to comparing URIs and dealing with unrecognized values, that are discussed in JWT [\[I-D.ietf-oauth-json-web-token\]](#) also apply here.

Similarly to other information included in a JWT 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 JWT learns about the entity that created the JWT since this will be important for any policy decisions. Integrity protection prevents an adversary from changing any elements conveyed within the JWT payload. Special care has to be applied when carrying symmetric keys inside the JWT since those do not only require integrity protection but also confidentiality protection.

In addition, proof-of-possession introduces its own unique security issues. Possessing the key is only valuable if it is kept secret. Appropriate means must be used to ensure that unintended parties do not learn the symmetric key or the private key (in case of an asymmetric crypto-system).

A recipient may not understand the newly introduced "cnf" claim and may consequently treat it as a bearer token. While this is a legitimate concern it is outside the scope of this specification since demonstration the possession of the key associated with the "cnf" claim is not covered by this specification. For more details please consult [\[I-D.hunt-oauth-pop-architecture\]](#).

#### 5. IANA Considerations

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

Values are registered with a Specification Required [[RFC5226](#)] after a two-week review period on the [TBD]@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 Expert(s) may approve registration once they are satisfied that such a specification will be published.

Registration requests must be sent to the [TBD]@ietf.org mailing list for review and comment, with an appropriate subject (e.g., "Request for access token type: example"). [[ Note to the RFC Editor: The name of the mailing list should be determined in consultation with the IESG and IANA. Suggested name: jwt-reg-review. ]]

Within the review period, the Designated Expert(s) will either approve or deny the registration request, communicating this decision to the review list and IANA. Denials should include an explanation and, if applicable, suggestions as to how to make the request successful. Registration requests that are undetermined for a period longer than 21 days can be brought to the IESG's attention (using the iesg@iesg.org mailing list) for resolution.

Criteria that should be applied by the Designated Expert(s) includes 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 whether the registration makes sense.

IANA must only accept registry updates from the Designated Expert(s) and should direct all requests for registration to the review mailing list.

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(s).

### [5.1.](#) JSON Web Token Claims Registration

This specification registers the "cnf" claim in the IANA JSON Web Token Claims registry defined in [[I-D.ietf-oauth-json-web-token](#)].

#### [5.1.1.](#) Registry Contents

- o Claim Name: "cnf"
- o Claim Description: Confirmation
- o Change Controller: IESG
- o Specification Document(s): [Section 3.3](#) of this document

### [5.2.](#) JWT Confirmation Methods Registry

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

#### [5.2.1.](#) Registration Template

Confirmation Method Value:

The name requested (e.g., "example"). Because a core goal of this specification is for the resulting representations to be compact, it is RECOMMENDED that the name be short -- not to exceed 8 characters without a compelling reason to do so. This name is case-sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Expert(s) state that there is a compelling reason to allow an exception in this particular case.

Confirmation Method Description:

Brief description of the confirmation method (e.g., "Example description").

Change Controller:

For Standards Track RFCs, state "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(s) that specify the parameter, preferably including URI(s) that can be used to retrieve copies of the document(s). An indication of the relevant sections may also be included but is not required.

### [5.2.2.](#) Initial Registry Contents

- o Confirmation Method Value: "jwk"
- o Confirmation Method Description: JSON Web Key or Encrypted JSON Web Key
- o Change Controller: IESG
- o Specification Document(s): [Section 3](#) of [[ this document ]]

## [6.](#) Acknowledgements

We would like to thanks James Manger for his review feedback.

## [7.](#) References

### [7.1.](#) Normative References

- [I-D.ietf-jose-json-web-encryption]  
Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", [draft-ietf-jose-json-web-encryption-29](#) (work in progress), June 2014.
- [I-D.ietf-jose-json-web-key]  
Jones, M., "JSON Web Key (JWK)", [draft-ietf-jose-json-web-key-29](#) (work in progress), June 2014.
- [I-D.ietf-oauth-json-web-token]  
Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", [draft-ietf-oauth-json-web-token-23](#) (work in progress), June 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.

### [7.2.](#) Informative References

- [I-D.hunt-oauth-pop-architecture]  
Hunt, P., Richer, J., Mills, W., Mishra, P., and H. Tschofenig, "OAuth 2.0 Proof-of-Possession (PoP) Security Architecture", [draft-hunt-oauth-pop-architecture-01](#) (work

in progress), April 2014.

Jones, et al.

Expires December 28, 2014

[Page 9]

---

Internet-Draft

proof-of-possession for JWTs

June 2014

[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.

## [Appendix A](#). Document History

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

-01

- o Updated affiliation.
- o Various editorial changes.
- o Updates to the security consideration section based on review feedback by James Manger.
- o Included the kid element in the examples (as requested by James Manger).
- o Expanded the introduction section.
- o Moved the terminology/RFC2119 boilerplate text from the introduction to a separate terminology section.

-00

- o Wrote the first draft.

## Authors' Addresses

Michael B. Jones  
Microsoft

Email: [mbj@microsoft.com](mailto:mbj@microsoft.com)  
URI: <http://self-issued.info/>

John Bradley  
Ping Identity

Email: [ve7jtb@ve7jtb.com](mailto:ve7jtb@ve7jtb.com)

URI: <http://www.thread-safe.com/>

Hannes Tschofenig  
ARM Limited  
Austria

Email: [Hannes.Tschofenig@gmx.net](mailto:Hannes.Tschofenig@gmx.net)  
URI: <http://www.tschofenig.priv.at>

Jones, et al.

Expires December 28, 2014

[Page 10]