

OAuth Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 3, 2014

M. Jones  
Microsoft  
J. Bradley  
Ping Identity  
H. Tschofenig  
April 1, 2014

Proof-Of-Possession Semantics for JSON Web Tokens (JWTs)  
draft-jones-oauth-proof-of-possession-00

## 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 October 3, 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="#">3</a>
<a href="#">1.1.</a>	<a href="#">Notational Conventions</a>	<a href="#">3</a>
<a href="#">1.2.</a>	<a href="#">Terminology</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Proof-Of-Possession Representation</a>	<a href="#">3</a>
<a href="#">2.1.</a>	<a href="#">Proof-of-Possession of a Private Key</a>	<a href="#">4</a>
<a href="#">2.2.</a>	<a href="#">Proof-of-Possession of a Symmetric Key</a>	<a href="#">4</a>
<a href="#">2.3.</a>	<a href="#">Confirmation</a>	<a href="#">5</a>
<a href="#">2.4.</a>	<a href="#">Specifics Intentionally Not Specified</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">Security Considerations</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">IANA Considerations</a>	<a href="#">6</a>
<a href="#">4.1.</a>	<a href="#">JSON Web Token Claims Registration</a>	<a href="#">7</a>
<a href="#">4.1.1.</a>	<a href="#">Registry Contents</a>	<a href="#">7</a>
<a href="#">4.2.</a>	<a href="#">JWT Confirmation Methods Registry</a>	<a href="#">7</a>
<a href="#">4.2.1.</a>	<a href="#">Registration Template</a>	<a href="#">8</a>
<a href="#">4.2.2.</a>	<a href="#">Initial Registry Contents</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">References</a>	<a href="#">8</a>
<a href="#">5.1.</a>	<a href="#">Normative References</a>	<a href="#">8</a>
<a href="#">5.2.</a>	<a href="#">Informative References</a>	<a href="#">9</a>
<a href="#">Appendix A.</a>	<a href="#">Open Issues</a>	<a href="#">9</a>
<a href="#">Appendix B.</a>	<a href="#">Document History</a>	<a href="#">9</a>
	<a href="#">Authors' Addresses</a>	<a href="#">9</a>

## [1.](#) Introduction

This specification defines how to express a declaration in a JSON Web Token (JWT) [[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.

### [1.1.](#) Notational Conventions

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.

### [1.2.](#) Terminology

This specification uses terms defined in the JSON Web Token (JWT) [[JWT](#)], JSON Web Key (JWK) [[JWK](#)], and JSON Web Encryption (JWE) [[JWE](#)] specifications.

These terms are defined for use by this specification:

#### Presenter

Party that possesses the key identified by the JWT.

## [2.](#) 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.

### [2.1](#). Proof-of-Possession of a Private Key

When the key held by the issuer is a private key, the value of the "jwk" member is a JSON Web Key (JWK) [[JWK](#)] representing the corresponding public key. The following example demonstrates such a declaration in the JWT Claims Set of a JWT:

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

The JWK MUST contain the REQUIRED key elements for a JWK of that key type and MAY contain other JWK elements, including the "kid" (key ID) element.

## [2.2.](#) 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) [[JWK](#)] encrypted to the recipient using the JWE Compact Serialization containing the symmetric key. The rules for encrypting a JWK are found in [Section 6](#) of the JSON Web Key [[JWK](#)] specification.

The following is an example symmetric key that could be encrypted for use in the "jwk" member:

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

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

The following is an example JWE Header could be used when encrypting this key:

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

The following example JWT Claims Set of a JWT illustrates the use of an 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":
      "eyJhbGciOiJSU0ExXzUiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2Iiwia3R5Ijoia256Iiwia2VudCI6ImZzUA5XKMYoVHyzzff5oRJxl-IXRtztJ6uE"
  }
}
```

```
        andrK2pzb24ifQ. ... (remainder of JWE omitted for brevity)"
    }
}
```

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.

### [2.3.](#) Confirmation

The "cnf" (confirmation) claim is used in the JWT to contain the "jwk" member because a proof-of-possession key may not be the only means of confirming the authenticity of the token. This 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. 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 4.2](#) and registers the "jwk" member within the registry.

### [2.4.](#) 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.

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.

### [3.](#) Security Considerations

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

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 secret or symmetric key value.

Proof-of-possession via encrypted symmetric secrets is subject to replay attacks. This attack can be avoided when a signed nonce or challenge is used, since the recipient can use a distinct nonce or challenged for each interaction.

### [4.](#) 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](mailto: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, that Expert should defer to the judgment of the other Expert(s).

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

This specification registers the "cnf" claim in the IANA JSON Web Token Claims registry defined in [\[JWT\]](#).

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

- o Claim Name: "cnf"
- o Claim Description: Confirmation
- o Change Controller: IESG
- o Specification Document(s): [Section 2.3](#) of this document

#### [4.2.](#) 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



#### [4.2.1.](#) 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.

#### [4.2.2.](#) 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 2](#) of [[ this document ]]

### [5.](#) References

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

- [JWE] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", [draft-ietf-jose-json-web-encryption](#) (work in progress), March 2014.
- [JWK] Jones, M., "JSON Web Key (JWK)", [draft-ietf-jose-json-web-key](#) (work in progress),

March 2014.

- [JWT] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", [draft-ietf-oauth-json-web-token](#) (work in progress), March 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.

## [5.2.](#) Informative References

- [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.](#) Open Issues

In some conversations, we have said that it is the issuer of the JWT that possesses the key, and in some conversations, we have said that it is the presenter of the JWT that possesses the key. Which description should we use?

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

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

-00

- o Wrote the first draft.

Internet-Draft

proof-of-possession for JWTs

April 2014

#### 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

Email: [Hannes.Tschofenig@gmx.net](mailto:Hannes.Tschofenig@gmx.net)

URI: <http://www.tschofenig.priv.at>

Jones, et al.

Expires October 3, 2014

[Page 10]