OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 22, 2015

M. Jones
Microsoft
J. Bradley
Ping Identity
H. Tschofenig
ARM Limited

July 21, 2014

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

Table of Contents

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.

[[Editorial Note: This paragraph needs to be updated to provide more context and possibly also to describe the use of asymmetric keys instead. It's not clear that the symmetric case is as useful or valuable, and it is certainly more complicated.]] Envision the following use case: An OAuth 2.0 authorization server generates a JWT and places an encrypted 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 member). When the presenter needs to utilize the JWT to at 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 is interacting 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 cryptographically protected messages exchanged with the presenter.

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.

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 by this specification:

Presenter
: Party that possesses the key identified by the JWT.

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.

3.1. Proof-of-Possession of an Asymmetric Key

When the key held by the issuer is an asymmetric private key, the value of the "jwk" member is a JSON Web Key (JWK) [JWK] representing the corresponding asymmetric public key. The following example 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":{
     "kty":"EC",
     "use":"sig",
     "crv":"P-256",
     "x":"18wHLeIgW9wVN6VD1Txgpqy2LszYkMf6J8njVAibvhM",
     "y":"-V4dS4UaLMgP_4fY4j8ir7cl1TXlFdAgcx55o7TkcSA"
    }
  }
}
```

The JWK MUST contain the required key members for a JWK of that key type and MAY contain other JWK members, including the "kid" (key ID) member.

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) [JWK] encrypted to a key known 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 example illustrates a symmetric key that could subsequently be encrypted for use in the "jwk" member:

```
{
 "kty":"oct",
 "alg":"HS256",
 "k":"ZoRSOrFzN_FzUA5XKMYoVHyzff5oRJxl-IXRtztJ6uE"
}
```

The UTF-8 [RFC3629] encoding of this JWK would be used as the JWE Plaintext when encrypting the key.

The following example is a JWE Header that 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":
     "eyJhbGciOiJSU0ExXzUiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiY3R5Ijoi
      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.

3.3. 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 members 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.

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.

For an example specification that uses the mechanisms defined in this document, see [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 [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 private key 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.

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 not only require integrity protection, but also confidentiality protection.

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, that Expert should defer to the judgment of the other 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 [JWT].

5.1.1. Registry Contents

- Claim Name: "cnf"
- Claim Description: Confirmation
- Change Controller: IESG
- 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

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

6. References

6.1. Normative References

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

[JWK] Jones, M., "JSON Web Key (JWK)", draft-ietf-jose-json-web-key (work in progress), July 2014.

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

6.2. 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-02 (work in progress), 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. 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. Acknowledgements

The authors wish to thank James Manger for his review of the specification.

Appendix C. Document History

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

-02

- Used the same section structuring conventions as the JWT specification.
- Reverted some changes introduced in -01 without adequate prior review.
- Applied some editorial corrections.

-01

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

-00

- Wrote the first draft.

Authors' Addresses

Michael B. Jones
Microsoft

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

John Bradley
Ping Identity

Email: ve7jtb@ve7jtb.com
URI: http://www.thread-safe.com/

Hannes Tschofenig
ARM Limited
Austria

Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at