JSON Web Key (JWK)
draft-ietf-jose-json-web-key-02

Abstract

   A JSON Web Key (JWK) is a JSON data structure that represents a
   public key. This specification also defines a JSON Web Key Set (JWK
   Set) JSON data structure for representing a set of JWKs.
   Cryptographic algorithms and identifiers used with this specification
   are enumerated in the separate JSON Web Algorithms (JWA)
   specification.

Requirements Language

   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].

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 November 13, 2012.

Table of Contents

1. Introduction

A JSON Web Key (JWK) is a JSON data structure that represents a
public key as a JSON object [RFC4627]. This specification also
defines a JSON Web Key Set (JWK Set) JSON data structure for
representing a set of JWKs. Cryptographic algorithms and identifiers
used with this specification are enumerated in the separate JSON Web
Algorithms (JWA) [JWA] specification.

Non-goals for this specification include representing private keys,
representing symmetric keys, representing certificate chains,
representing certified keys, and replacing X.509 certificates.

JWKs are used in the JSON Web Signature (JWS) [JWS] "jwk" (JSON Web
Key) header parameter and the JSON Web Encryption (JWE) [JWE] "jwk"
(JSON Web Key) and "epk" (Ephemeral Public Key) header parameters.
The resources referenced by the JWS "jku" (JWK Set URL) and JWE "jku"
(JWK Set URL) header parameters contain JWK Sets.

2. Terminology

JSON Web Key (JWK) A JSON data structure that represents a public
   key.

JSON Web Key Set (JWK Set) A JSON object that contains an array of
   JWKs as a member.

Base64url Encoding For the purposes of this specification, this term
   always refers to the URL- and filename-safe Base64 encoding
   described in RFC 4648 [RFC4648], Section 5, with the (non URL-
   safe) '=' padding characters omitted, as permitted by Section 3.2.
   (See Appendix B of [JWS] for notes on implementing base64url
   encoding without padding.)

3. Example JSON Web Key Set

The following example JWK Set contains two public keys represented as
JWKs: one using an Elliptic Curve algorithm and a second one using an
RSA algorithm. The first specifies that the key is to be used for
encryption. Both provide a Key ID for key matching purposes. In
both cases, integers are represented using the base64url encoding of
their big endian representations. (Long lines are broken are for
display purposes only.)

```
{"keys":
  [
    {"alg":"EC",
     "crv":"P-256",
     "x":"MKBCTNIcKUSDii11ySs3526iDZ8AiTo7Tu6KPAqv7D4",
     "y":"4Etl6SRW2YiLUrN5vfvVHuhp7x8PxltmWWlbbM4IFyM",
     "use":"enc",
     "kid":"1"},

    {"alg":"RSA",
     "mod": "0vx7agoebGcQSuuPiLJXZptN9nndrQmbXEps2aiAFbWhM78LhWx
4cbbfAAtVT86zwu1RK7aPFFxuhDR1L6tSoc_BJECPebWKRXjBZCiFV4n3oknjhMs
tn64tZ_2W-5JsGY4Hc5n9yBXArwl93lqt7_RN5w6Cf0h4QyQ5v-65YGjQR0_FDW2
QvzqY368QQMicAtaSqzs8KJZgnYb9c7d0zgdAZHzu6qMQvRL5hajrn1n91CbOpbI
SD08qNLyrdkt-bFTWhAI4vMQFh6WeZu0fM4lFd2NcRwr3XPksINHaQ-G_xBniIqb
w0Ls1jF44-csFCur-kEgU8awapJzKnqDKgw",
     "exp":"AQAB",
     "kid":"2011-04-29"}
  ]
}
```

4. JSON Web Key (JWK) Format

   A JSON Web Key (JWK) is a JSON object containing specific members, as
   specified below. Those members that are common to all key types are
   defined below.

   JWKs also require members that are specific to the particular key
   algorithm family to represent the key parameters. These algorithm
   specific members are defined in Section 5 of the JSON Web Algorithms
   (JWA) [JWA] specification.

   The member names within a JWK MUST be unique; objects with duplicate
   member names MUST be rejected.

   Additional members MAY be present in the JWK. If present, they MUST
   be understood by implementations using them. Parameters for
   representing keys for additional algorithm families or additional key
   properties SHOULD either be defined in the IANA JSON Web Key
   Parameters registry [JWA] or be a URI that contains a collision
   resistant namespace.

4.1. "alg" (Algorithm Family) Parameter

   The "alg" (algorithm family) member identifies the cryptographic
   algorithm family used with the key. A list of defined "alg" values
   is presented in Section 5.1 of the JSON Web Algorithms (JWA) [JWA]

specification. Specific additional members are required to represent
the key, depending upon the algorithm family. These members are
specified in Section 5 of the JSON Web Algorithms (JWA) [JWA]
specification. The "alg" value is case sensitive. Its value MUST be
a string.

"alg" values SHOULD either be defined in the IANA JSON Web Key
Algorithm Families registry [JWA] or be a URI that contains a
collision resistant namespace.

4.2. "use" (Key Use) Parameter

The "use" (key use) member identifies the intended use of the key.
Values defined by this specification are "sig" (signature) and "enc"
(encryption). Other values MAY be used. The "use" value is case
sensitive. Its value MUST be a string. This member is OPTIONAL.

4.3. "kid" (Key ID) Parameter

The "kid" (key ID) member can be used to match a specific key. This
can be used, for instance, to choose among a set of keys within the
JWK during key rollover. When used with JWS or JWE, the "kid" value
MAY be used to match a JWS or JWE "kid" header parameter value. The
interpretation of the "kid" value is unspecified. Its value MUST be
a string. This member is OPTIONAL.

5. JSON Web Key Set (JWK Set) Format

A JSON Web Key Set (JWK Set) is a JSON object that contains an array
of JSON Web Key values as the value of its "keys" member.

The member names within a JWK Set MUST be unique; objects with
duplicate member names MUST be rejected.

Additional members MAY be present in the JWK Set. If present, they
MUST be understood by implementations using them. Parameters for
representing additional properties of JWK Sets SHOULD either be
defined in the IANA JSON Web Key Set Parameters registry Section 6.1
or be a URI that contains a collision resistant namespace.

5.1. "keys" (JSON Web Key Set) Parameter

The value of the "keys" (JSON Web Key Set) member is an array of JSON
Web Key (JWK) values. This member is REQUIRED.

6. IANA Considerations

6.1. JSON Web Key Set Parameters Registry

   This specification establishes the IANA JSON Web Key Set Parameters
   registry for reserved JWK Set parameter names. Inclusion in the
   registry is RFC Required in the RFC 5226 [RFC5226] sense. The
   registry records the reserved parameter name and a reference to the
   RFC that defines it. This specification registers the parameter
   names defined in Section 5.

7. Security Considerations

   A key is no more trustworthy than the method by which it was
   received.

   The security considerations in XML DSIG 2.0
   [W3C.CR-xmldsig-core2-20120124], about public key representations
   also apply to this specification, other than those that are XML
   specific.

8. Open Issues and Things To Be Done (TBD)

   The following items remain to be done in this draft:

   o (None at present)

9. References

9.1. Normative References

   [JWA]   Jones, M., "JSON Web Algorithms (JWA)", May 2012.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC4627] Crockford, D., "The application/json Media Type for
              JavaScript Object Notation (JSON)", RFC 4627, July 2006.

   [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data
              Encodings", RFC 4648, October 2006.

   [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
              IANA Considerations Section in RFCs", BCP 26, RFC 5226,
              May 2008.

9.2. Informative References

[JWE]   Jones, M., Rescorla, E., and J. Hildebrand, "JSON Web
             Encryption (JWE)", May 2012.

[JWS]   Jones, M., Bradley, J., and N. Sakimura, "JSON Web
             Signature (JWS)", May 2012.

[MagicSignatures]
             Panzer (editor), J., Laurie, B., and D. Balfanz, "Magic
             Signatures", January 2011.

[W3C.CR-xmldsig-core2-20120124]
             Eastlake, D., Reagle, J., Yiu, K., Solo, D., Datta, P.,
             Hirsch, F., Cantor, S., and T. Roessler, "XML Signature
             Syntax and Processing Version 2.0", World Wide Web
             Consortium CR CR-xmldsig-core2-20120124, January 2012,
             <http://www.w3.org/TR/2012/CR-xmldsig-core2-20120124>.

Appendix A. Acknowledgements

A JSON representation for RSA public keys was previously introduced
in Magic Signatures [MagicSignatures].

Appendix B. Document History

-02

o Simplified JWK terminology to get replace the "JWK Key Object" and
   "JWK Container Object" terms with simply "JSON Web Key (JWK)" and
   "JSON Web Key Set (JWK Set)" and to eliminate potential confusion
   between single keys and sets of keys. As part of this change, the
   top-level member name for a set of keys was changed from "jwk" to
   "keys".

o Clarified that values with duplicate member names MUST be
   rejected.

o Established JSON Web Key Set Parameters registry.

o Explicitly listed non-goals in the introduction.

o Moved algorithm-specific definitions from JWK to JWA.

o Reformatted to give each member definition its own section
   heading.

   -01

   o Corrected the Magic Signatures reference.

   -00

   o Created the initial IETF draft based upon
      draft-jones-json-web-key-03 with no normative changes.

Author's Address

   Michael B. Jones
   Microsoft

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