JOSE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

M. Jones
Microsoft

July 16, 2012

JSON Web Key (JWK)
draft-ietf-jose-json-web-key-04

Abstract

A JSON Web Key (JWK) is a JavaScript Object Notation (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 for use with this specification are described in the separate JSON Web Algorithms (JWA) specification.

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 January 17, 2013.

Copyright Notice

Copyright (c) 2012 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

1. Introduction 3
1.1. Notational Conventions 3
2. Terminology 3
3. Example JSON Web Key Set 4
4. JSON Web Key (JWK) Format 4
4.1. "alg" (Algorithm Family) Parameter 5
4.2. "use" (Key Use) Parameter 5
4.3. "kid" (Key ID) Parameter 5
5. JSON Web Key Set (JWK Set) Format 6
5.1. "keys" (JSON Web Key Set) Parameter 6
6. IANA Considerations 6
6.1. JSON Web Key Parameters Registry 7
6.1.1. Registration Template 7
6.1.2. Initial Registry Contents 7
6.2. JSON Web Key Set Parameters Registry 8
6.2.1. Registration Template 8
6.2.2. Initial Registry Contents 8
7. Security Considerations 9
8. Open Issues 9
9. References 9
9.1. Normative References 9
9.2. Informative References 10
Appendix A. Acknowledgements 10
Appendix B. Document History 10
Author's Address 12

1. Introduction

A JSON Web Key (JWK) is a JavaScript Object Notation (JSON) [RFC4627] 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 for use with this specification are described in the separate JSON Web Algorithms (JWA) [JWA] specification.

Goals for this specification do not include representing private keys, representing symmetric keys, representing certificate chains, representing certified keys, and replacing X.509 certificates.

JWKs and JWK Sets are used in the JSON Web Signature (JWS) [JWS] and JSON Web Encryption (JWE) [JWE] specifications.

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

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 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 C of [JWS] for notes on implementing base64url encoding without padding.)

Collision Resistant Namespace A namespace that allows names to be allocated in a manner such that they are highly unlikely to collide with other names. For instance, collision resistance can be achieved through administrative delegation of portions of the namespace or through use of collision-resistant name allocation functions. Examples of Collision Resistant Namespaces include: Domain Names, Object Identifiers (OIDs) as defined in the ITU-T X.660 and X.670 Recommendation series, and Universally Unique IDentifiers (UUIDs) [RFC4122]. When using an administratively delegated namespace, the definer of a name needs to take

reasonable precautions to ensure they are in control of the portion of the namespace they use to define the name.

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.

In addition to the common parameters, each JWK will have members that are specific to the key being represented. These members represent the parameters of the key. Section 5 of the JSON Web Algorithms (JWA) [JWA] specification defines multiple kinds of public keys and their associated members.

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. Member names used for representing key parameters for different kinds of keys need not be distinct. Member names SHOULD either be registered in the IANA JSON Web Key Parameters registry Section 6.1 or be URIs that contain a Collision Resistant Namespace.

4.1. "alg" (Algorithm Family) Parameter

The "alg" (algorithm family) member identifies the cryptographic algorithm family used with the key. "alg" values SHOULD either be registered in the IANA JSON Web Key Algorithm Families registry [JWA] or be a URI that contains a Collision Resistant Namespace. The "alg" value is a case sensitive string.

A list of defined "alg" values can be found in the IANA JSON Web Key Algorithm Families registry [JWA]; the initial contents of this registry is the values defined in Section 5.1 of the JSON Web Algorithms (JWA) [JWA] specification.

Additional members used with these "alg" values can be found in the IANA JSON Web Key Parameters registry Section 6.1; the initial contents of this registry is the values defined in Sections 5.2 and 5.3 of the JSON Web Algorithms (JWA) [JWA] specification.

4.2. "use" (Key Use) Parameter

The "use" (key use) member identifies the intended use of the key. Values defined by this specification are:

o "sig" (signature)

o "enc" (encryption)

Other values MAY be used. The "use" value is a case sensitive 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. The interpretation of the "kid" value is unspecified. Key ID values within a JWK Set need not be unique. The "kid" value is a case sensitive string. This member is OPTIONAL.

When used with JWS or JWE, the "kid" value MAY be used to match a JWS

or JWE "kid" header parameter value.

In some contexts, different keys using the same Key ID value might be present, with the keys being disambiguated using other information, such as the "alg" or "use" values. For example, imagine "kid" values like "Current", "Upcoming", and "Deprecated", used for key rollover guidance. One could apply a label to all keys where the classification fits. If there are multiple "Current" keys, then in this example, they might be differentiated either by having different "alg" or "use" values, or some combination of both. As one example, there might only be one current RSA signing key and one current Elliptic Curve signing key, but both would be "Current".

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 registered in the IANA JSON Web Key Set Parameters registry Section 6.2 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

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 RFC-EDITOR: The name of the mailing list should be determined in consultation with the IESG and IANA. Suggested name: jose-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.

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

6.1. JSON Web Key Parameters Registry

This specification establishes the IANA JSON Web Key Parameters registry for reserved JWK parameter names. The registry records the reserved parameter name and a reference to the specification that defines it. This specification registers the parameter names defined in Section 4.

6.1.1. Registration Template

Parameter Name:
: The name requested (e.g., "example"). This name is case sensitive. Names that match other registered names in a case insensitive manner SHOULD NOT be accepted.

Change Controller:
: For standards-track RFCs, state "IETF". For others, give the name of the responsible party. Other details (e.g., postal address, e-mail address, home page URI) may also be included.

Specification Document(s):
: Reference to the document that specifies the parameter, preferably including a URI that can be used to retrieve a copy of the document. An indication of the relevant sections may also be included, but is not required.

6.1.2. Initial Registry Contents

- Parameter Name: "alg"
- Change Controller: IETF

- Specification Document(s): Section 4.1 of [[this document]]
- Parameter Name: "use"
- Change Controller: IETF
- Specification Document(s): Section 4.2 of [[this document]]
- Parameter Name: "kid"
- Change Controller: IETF
- Specification Document(s): Section 4.3 of [[this document]]

6.2. JSON Web Key Set Parameters Registry

This specification establishes the IANA JSON Web Key Set Parameters registry for reserved JWK Set parameter names. The registry records the reserved parameter name and a reference to the specification that defines it. This specification registers the parameter names defined in Section 5.

6.2.1. Registration Template

Parameter Name:
: The name requested (e.g., "example"). This name is case sensitive. Names that match other registered names in a case insensitive manner SHOULD NOT be accepted.

Change Controller:
: For standards-track RFCs, state "IETF". For others, give the name of the responsible party. Other details (e.g., postal address, e-mail address, home page URI) may also be included.

Specification Document(s):
: Reference to the document that specifies the parameter, preferably including a URI that can be used to retrieve a copy of the document. An indication of the relevant sections may also be included, but is not required.

6.2.2. Initial Registry Contents

- Parameter Name: "keys"
- Change Controller: IETF
- Specification Document(s): Section 5.1 of [[this document]]

7. Security Considerations

All of the security issues faced by any cryptographic application must be faced by a JWS/JWE/JWK agent. Among these issues are protecting the user's private key, preventing various attacks, and helping the user avoid mistakes such as inadvertently encrypting a message for the wrong recipient. The entire list of security considerations is beyond the scope of this document, but some significant concerns are listed here.

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

Per Section 4.3, applications should not assume that "kid" values are unique within a JWK Set.

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

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

The following items remain to be considered or done in this draft:

- There was a request to define the key use value "both". This would seem to be semantically redundant, since omitting a key use value effectively allows unconstrained use of the key. For what it's worth, omitting the use parameter is how XMLDSIG expresses the same thing, so we're currently parallel to XMLDSIG. Furthermore, legitimizing the use of a single key for both signing and encryption seems like it may be a bad idea, since there's a potential vulnerability with using the same key for both signing and encryption.

9. References

9.1. Normative References

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

[W3C.CR-xmldsig-core2-20120124] Reagle, J., Hirsch, F., Cantor, S., Roessler, T., Eastlake, D., Yiu, K., Solo, D., and P. Datta, "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>.

9.2. Informative References

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

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

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

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

Appendix A. Acknowledgements

A JSON representation for RSA public keys was previously introduced by John Panzer, Ben Laurie, and Dirk Balfanz in Magic Signatures [MagicSignatures].

Appendix B. Document History

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

-04

- Refer to the registries as the primary sources of defined values and then secondarily reference the sections defining the initial contents of the registries.

- Normatively reference XML DSIG 2.0 [W3C.CR-xmldsig-core2-20120124] for its security considerations.

- Added this language to Registration Templates: "This name is case sensitive. Names that match other registered names in a case insensitive manner SHOULD NOT be accepted."

- Described additional open issues.

- Applied editorial suggestions.

-03

- Clarified that "kid" values need not be unique within a JWK Set.

- Moved JSON Web Key Parameters registry to the JWK specification.

- Added "Collision Resistant Namespace" to the terminology section.

- Changed registration requirements from RFC Required to Specification Required with Expert Review.

- Added Registration Template sections for defined registries.

- Added Registry Contents sections to populate registry values.

- Numerous editorial improvements.

-02

- 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".

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

- Established JSON Web Key Set Parameters registry.

- Explicitly listed non-goals in the introduction.

- Moved algorithm-specific definitions from JWK to JWA.
- Reformatted to give each member definition its own section heading.

-01

- Corrected the Magic Signatures reference.

-00

- 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/