| | |
|---|---|
| JOSE Working Group | M. Jones |
| Internet-Draft | Microsoft |
| Intended status: Standards Track | J. Bradley |
| Expires: April 18, 2013 | independent |
| | N. Sakimura |
| | Nomura Research Institute |
| | October 15, 2012 |


TOC

# JSON Web Signature JSON Serialization (JWS-JS) draft-jones-jose-jws-json-serialization-02


## Abstract

The JSON Web Signature JSON Serialization (JWS-JS) is a means of representing content secured with digital signatures or Message Authentication Codes (MACs) using JavaScript Object Notation (JSON) data structures. This specification describes a means of representing secured content as a JSON data object (as opposed to the JWS specification, which uses a compact serialization with a URL-safe representation). It enables multiple digital signatures and/or MACs to be applied to the same content (unlike JWS). Cryptographic algorithms and identifiers used with this specification are described in the separate JSON Web Algorithms (JWA) specification. The JSON Serialization for related encryption functionality is described in the separate JSON Web Encryption JSON Serialization (JWE-JS) specification.




---

## Table of Contents

---

TOC

## 1. Introduction

The JSON Web Signature JSON Serialization (JWS-JS) is a format for representing content secured with digital signatures or Message Authentication Codes (MACs) as a JavaScript Object Notation (JSON) **[RFC4627]** object. It enables multiple digital signatures and/or MACs to be applied to the same content (unlike JWS **[JWS]**). The digital signature and MAC mechanisms used are independent of the type of content being secured, allowing arbitrary content to be secured. Cryptographic algorithms and identifiers used with this specification are described in the separate JSON Web Algorithms (JWA) **[JWA]** specification. The JSON Serialization for related encryption functionality is described in the separate JSON Web Encryption JSON Serialization (JWE-JS) **[JWE-JS]** specification.

---

TOC

### 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]**.

---

TOC

## 2. Terminology

This specification uses the same terminology as the JSON Web Signature (JWS) **[JWS]** specification.

---

TOC

## 3. JSON Serialization

The JSON Serialization represents secured content as a JSON object with a `recipients` member containing an array of per-recipient information and a `payload` member containing a shared Encoded JWS Payload value. Each member of the `recipients` array is a JSON object with a `header` member containing an Encoded JWS Header value and a `signature` member containing an Encoded JWS Signature value.

Unlike the compact serialization used by JWSs, content using the JSON Serialization MAY be secured with more than one digital signature and/or MAC value. Each is represented as an Encoded JWS Signature value in the `signature` member of an object in the `recipients` array. For each, there is an Encoded JWS Encoded Header value in the `header` member of the same object in the `recipients` array. This specifies the digital signature or MAC applied to the Encoded JWS Header value and the shared Encoded JWS Payload value to create the JWS Signature value. Therefore, the syntax is:

```
{"recipients":[
  {"header":"<header 1 contents>",
   "signature":"<signature 1 contents>"},
  ...
  {"header":"<header N contents>",
   "signature":"<signature N contents>"}],
```

```
 "payload":"<payload contents>"
}
```

The contents of the Encoded JWS Header, Encoded JWS Payload, and Encoded JWS Signature values are exactly as specified in JSON Web Signature (JWS) **[JWS]**. They are interpreted and validated in the same manner, with each corresponding `header` and `signature` value being created and validated together.

Each JWS Signature value is computed on the JWS Secured Input corresponding to the concatenation of the Encoded JWS Header, a period ('.') character, and the Encoded JWS Payload in the same manner described in the JWS specification. This has the desirable result that each Encoded JWS signature value in the `recipients` array is identical to the value that would be used for the same parameters in a JWS.

---

TOC

## 4. Example JWS-JS

This section contains an example using the JWS JSON Serialization. This example demonstrates the capability for conveying multiple digital signatures and/or MACs for the same payload.

The Encoded JWS Payload used in this example is the same as used in the examples in Appendix A of JWS (with line breaks for display purposes only):

```
eyJpc3MiOiJqb2UiLA0KICJleHAiOjEzMDA4MTkzODAsDQogImh0dHA6Ly9leGFt
cGxlLmNvbS9pc19yb290Ijp0cnVlfQ
```

Two digital signatures are used in this example: an RSA SHA-256 signature, for which the header and signature values are the same as in Appendix A.2 of JWS, and an ECDSA P-256 SHA-256 signature, for which the header and signature values are the same as in Appendix A.3 of JWS. The two Decoded JWS Header Segments used are:

```
{"alg":"RS256"}
```

and:

```
{"alg":"ES256"}
```

Since the computations of the JWS Header and JWS Signature values are the same as in Appendix A.2 and Appendix A.3 of JWS, they are not repeated here.

The complete JSON Web Signature JSON Serialization (JWS-JS) for these values is as follows (with line breaks for display purposes only):

```
{"recipients":[
  {"header":"eyJhbGciOiJSUzI1NiJ9",
   "signature":
    "cC4hiUPoj9Eetdgtv3hF80EGrhuB__dzERat0XF9g2VtQgr9PJbu3XOiZj5RZ
     mh7AAuHIm4Bh-0Qc_lF5YKt_O8W2Fp5jujGbds9uJdbF9CUAr7t1dnZcAcQjb
     KBYNX4BAynRFdiuB--f_nZLgrnbyTyWzO75vRK5h6xBArLIARNPvkSjtQBMHl
     b1L07Qe7K0GarZRmB_eSN9383LcOLn6_dO--xi12jzDwusC-eOkHWEsqtFZES
     c6BfI7noOPqvhJ1phCnvWh6IeYI2w9QOYEUipUTI8np6LbgGY9Fs98rqVt5AX
     LIhWkWywlVmtVrBp0igcN_IoypGlUPQGe77Rw"},
  {"header":"eyJhbGciOiJFUzI1NiJ9",
   "signature":
    "DtEhU3ljbEg8L38VWAfUAqOyKAM6-Xx-F4GawxaepmXFCgfTjDxw5djxLa8IS
     lSApmWQxfKTUJqPP3-Kg6NU1Q"}],
 "payload":
```

```
    "eyJpc3MiOiJqb2UiLA0KICJleHAiOjEzMDA4MTkzODAsDQogImh0dHA6Ly9leGF
     tcGxlLmNvbS9pc19yb290Ijp0cnVlfQ"
  }
```

## 5. IANA Considerations

TOC

This specification makes no requests of IANA.

## 6. Security Considerations

TOC

The security considerations for this specification are the same as those for the JSON Web Signature (JWS) **[JWS]** specification.

## 7. References

TOC

### 7.1. Normative References

TOC


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

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

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

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


### 7.2. Informative References

TOC


**[JSS]** Bradley, J. and N. Sakimura (editor), "JSON Simple Sign," September 2010.

**[JWE-JS]** Jones, M., "JSON Web Encryption JSON Serialization (JWE-JS)," October 2012.

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


## Appendix A. Acknowledgements

TOC


JSON serializations for secured content were previously explored by **Magic Signatures** [MagicSignatures] and **JSON Simple Sign** [JSS].


## Appendix B. Open Issues

TOC

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

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

- Track changes that occur in the JWS spec.

## Appendix C. Document History

TOC

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

-02

- Changed to use an array of structures for per-recipient values, rather than a set of parallel arrays.

-01

- Generalized language to refer to Message Authentication Codes (MACs) rather than Hash-based Message Authentication Codes (HMACs).

-00

- Renamed draft-jones-json-web-signature-json-serialization to draft-jones-jose-jws-json-serialization to have "jose" be in the document name so it can be included in the Related Documents list at http://datatracker.ietf.org/wg/jose/. No normative changes.

draft-jones-json-web-signature-json-serialization-02

- Tracked editorial changes made to the JWS spec.

draft-jones-json-web-signature-json-serialization-01

- Corrected the Magic Signatures reference.

draft-jones-json-web-signature-json-serialization-00

- Created the initial version incorporating JOSE working group input and drawing from the JSON Serialization previously proposed in draft-jones-json-web-token-01.

---

## Authors' Addresses

TOC

Michael B. Jones
Microsoft
**Email:** mbj@microsoft.com
**URI:** http://self-issued.info/

John Bradley
independent
**Email:** ve7jtb@ve7jtb.com

Nat Sakimura
Nomura Research Institute
**Email:** n-sakimura@nri.co.jp