

Internet Engineering Task Force (IETF)  
Request for Comments: 8230  
Category: Standards Track  
ISSN: 2070-1721

M. Jones  
Microsoft  
September 2017

Using RSA Algorithms with  
CBOR Object Signing and Encryption (COSE) Messages

Abstract

The CBOR Object Signing and Encryption (COSE) specification defines cryptographic message encodings using Concise Binary Object Representation (CBOR). This specification defines algorithm encodings and representations enabling RSA algorithms to be used for COSE messages. Encodings are specified for the use of RSA Probabilistic Signature Scheme (RSASSA-PSS) signatures, RSA Encryption Scheme - Optimal Asymmetric Encryption Padding (RSAES-OAEP) encryption, and RSA keys.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc8230>.

Copyright Notice

Copyright (c) 2017 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. Requirements Notation and Conventions . . . . .	3
2. RSASSA-PSS Signature Algorithm . . . . .	3
3. RSAES-OAEP Key Encryption Algorithm . . . . .	4
4. RSA Keys . . . . .	4
5. IANA Considerations . . . . .	6
5.1. COSE Algorithms Registrations . . . . .	6
5.2. COSE Key Type Registrations . . . . .	7
5.3. COSE Key Type Parameters Registrations . . . . .	7
6. Security Considerations . . . . .	9
6.1. Key Size Security Considerations . . . . .	9
6.2. RSASSA-PSS Security Considerations . . . . .	10
6.3. RSAES-OAEP Security Considerations . . . . .	10
7. References . . . . .	10
7.1. Normative References . . . . .	10
7.2. Informative References . . . . .	11
Acknowledgements . . . . .	12
Author's Address . . . . .	12

## 1. Introduction

The CBOR Object Signing and Encryption (COSE) [RFC8152] specification defines cryptographic message encodings using Concise Binary Object Representation (CBOR) [RFC7049]. This specification defines algorithm encodings and representations enabling RSA algorithms to be used for COSE messages.

### 1.1. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2. RSASSA-PSS Signature Algorithm

The RSASSA-PSS signature algorithm is defined in [RFC8017].

The RSASSA-PSS signature algorithm is parameterized with a hash function ( $h$ ), a mask generation function ( $mgf$ ), and a salt length ( $sLen$ ). For this specification, the mask generation function is fixed to be MGF1 as defined in [RFC8017]. It has been recommended that the same hash function be used for hashing the data as well as in the mask generation function. This specification follows this recommendation. The salt length is the same length as the hash function output.

Implementations need to check that the key type is 'RSA' when creating or verifying a signature.

The RSASSA-PSS algorithms specified in this document are in the following table.

Name	Value	Hash	Salt Length	Description
PS256	-37	SHA-256	32	RSASSA-PSS w/ SHA-256
PS384	-38	SHA-384	48	RSASSA-PSS w/ SHA-384
PS512	-39	SHA-512	64	RSASSA-PSS w/ SHA-512

Table 1: RSASSA-PSS Algorithm Values

### 3. RSAES-OAEP Key Encryption Algorithm

RSAES-OAEP is an asymmetric key encryption algorithm. The definition of RSAEA-OAEP can be found in Section 7.1 of [RFC8017]. The algorithm is parameterized using a mask generation function (mgf), a hash function (h), and encoding parameters (P). For the algorithm identifiers defined in this section:

- o mgf is always set to MGF1 as defined in [RFC8017] and uses the same hash function as h.
- o P is always set to the empty octet string.

The following table summarizes the rest of the values.

Name	Value	Hash	Description
RSAES-OAEP w/ RFC 8017 default parameters	-40	SHA-1	RSAES-OAEP w/ SHA-1
RSAES-OAEP w/ SHA-256	-41	SHA-256	RSAES-OAEP w/ SHA-256
RSAES-OAEP w/ SHA-512	-42	SHA-512	RSAES-OAEP w/ SHA-512

Table 2: RSAES-OAEP Algorithm Values

The key type MUST be 'RSA'.

### 4. RSA Keys

Key types are identified by the 'kty' member of the COSE\_Key object. This specification defines one value for this member in the following table.

Name	Value	Description
RSA	3	RSA Key

Table 3: Key Type Values

This document defines a key structure for both the public and private parts of RSA keys. Together, an RSA public key and an RSA private key form an RSA key pair.

The document also provides support for the so-called "multi-prime" RSA keys, in which the modulus may have more than two prime factors. The benefit of multi-prime RSA is lower computational cost for the decryption and signature primitives. For a discussion on how multi-prime affects the security of RSA cryptosystems, the reader is referred to [MultiPrimeRSA].

This document follows the naming convention of [RFC8017] for the naming of the fields of an RSA public or private key, and the corresponding fields have identical semantics. The requirements for fields for RSA keys are as follows:

- o For all keys, 'kty' MUST be present and MUST have a value of 3.
- o For public keys, the fields 'n' and 'e' MUST be present. All other fields defined in the following table below MUST be absent.
- o For private keys with two primes, the fields 'other', 'r\_i', 'd\_i', and 't\_i' MUST be absent; all other fields MUST be present.
- o For private keys with more than two primes, all fields MUST be present. For the third to nth primes, each of the primes is represented as a map containing the fields 'r\_i', 'd\_i', and 't\_i'. The field 'other' is an array of those maps.
- o All numeric key parameters are encoded in an unsigned big-endian representation as an octet sequence using the CBOR byte string type (major type 2). The octet sequence MUST utilize the minimum number of octets needed to represent the value. For instance, the value 32,768 is represented as the CBOR byte sequence 0b010\_00010, 0x80 0x00 (major type 2, additional information 2 for the length).

The following table provides a summary of the label values and the types associated with each of those labels.

Key Type	Name	Label	CBOR Type	Description
3	n	-1	bstr	the RSA modulus n
3	e	-2	bstr	the RSA public exponent e
3	d	-3	bstr	the RSA private exponent d
3	p	-4	bstr	the prime factor p of n
3	q	-5	bstr	the prime factor q of n
3	dP	-6	bstr	dP is $d \bmod (p - 1)$
3	dQ	-7	bstr	dQ is $d \bmod (q - 1)$
3	qInv	-8	bstr	qInv is the CRT coefficient $q^{-1} \bmod p$
3	other	-9	array	other prime infos, an array
3	r <sub>i</sub>	-10	bstr	a prime factor r <sub>i</sub> of n, where $i \geq 3$
3	d <sub>i</sub>	-11	bstr	$d_i = d \bmod (r_i - 1)$
3	t <sub>i</sub>	-12	bstr	the CRT coefficient $t_i = (r_1 * r_2 * \dots * r_{(i-1)})^{-1} \bmod r_i$

Table 4: RSA Key Parameters

## 5. IANA Considerations

### 5.1. COSE Algorithms Registrations

IANA has registered the following values in the IANA "COSE Algorithms" registry [IANA.COSE].

- o Name: PS256
- o Value: -37
- o Description: RSASSA-PSS w/ SHA-256
- o Reference: Section 2 of this document
- o Recommended: Yes
  
- o Name: PS384
- o Value: -38
- o Description: RSASSA-PSS w/ SHA-384
- o Reference: Section 2 of this document
- o Recommended: Yes

- o Name: PS512
- o Value: -39
- o Description: RSASSA-PSS w/ SHA-512
- o Reference: Section 2 of this document
- o Recommended: Yes
  
- o Name: RSAES-OAEP w/ RFC 8017 default parameters
- o Value: -40
- o Description: RSAES-OAEP w/ SHA-1
- o Reference: Section 3 of this document
- o Recommended: Yes
  
- o Name: RSAES-OAEP w/ SHA-256
- o Value: -41
- o Description: RSAES-OAEP w/ SHA-256
- o Reference: Section 3 of this document
- o Recommended: Yes
  
- o Name: RSAES-OAEP w/ SHA-512
- o Value: -42
- o Description: RSAES-OAEP w/ SHA-512
- o Reference: Section 3 of this document
- o Recommended: Yes

#### 5.2. COSE Key Type Registrations

IANA has registered the following value in the IANA "COSE Key Types" registry [IANA.COSE].

- o Name: RSA
- o Value: 3
- o Description: RSA Key
- o Reference: Section 4 of this document

#### 5.3. COSE Key Type Parameters Registrations

IANA has registered the following values in the IANA "COSE Key Type Parameters" registry [IANA.COSE].

- o Key Type: 3
- o Name: n
- o Label: -1
- o CBOR Type: bstr
- o Description: the RSA modulus n
- o Reference: Section 4 of this document

- o Key Type: 3
- o Name: e
- o Label: -2
- o CBOR Type: bstr
- o Description: the RSA public exponent e
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name: d
- o Label: -3
- o CBOR Type: bstr
- o Description: the RSA private exponent d
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name: p
- o Label: -4
- o CBOR Type: bstr
- o Description: the prime factor p of n
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name: q
- o Label: -5
- o CBOR Type: bstr
- o Description: the prime factor q of n
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name: dP
- o Label: -6
- o CBOR Type: bstr
- o Description:  $dP$  is  $d \bmod (p - 1)$
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name: dQ
- o Label: -7
- o CBOR Type: bstr
- o Description:  $dQ$  is  $d \bmod (q - 1)$
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name: qInv
- o Label: -8
- o CBOR Type: bstr
- o Description: qInv is the CRT coefficient  $q^{-1} \bmod p$
- o Reference: Section 4 of this document



- o Key Type: 3
- o Name: other
- o Label: -9
- o CBOR Type: array
- o Description: other prime infos, an array
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name:  $r_i$
- o Label: -10
- o CBOR Type: bstr
- o Description: a prime factor  $r_i$  of  $n$ , where  $i \geq 3$
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name:  $d_i$
- o Label: -11
- o CBOR Type: bstr
- o Description:  $d_i = d \bmod (r_i - 1)$
- o Reference: Section 4 of this document
  
- o Key Type: 3
- o Name:  $t_i$
- o Label: -12
- o CBOR Type: bstr
- o Description: the CRT coefficient  $t_i = (r_1 * r_2 * \dots * r_{(i-1)})^{-1} \bmod r_i$
- o Reference: Section 4 of this document

## 6. Security Considerations

### 6.1. Key Size Security Considerations

A key size of 2048 bits or larger MUST be used with these algorithms. This key size corresponds roughly to the same strength as provided by a 128-bit symmetric encryption algorithm. Implementations SHOULD be able to encrypt and decrypt with modulus between 2048 and 16K bits in length. Applications can impose additional restrictions on the length of the modulus.

In addition to needing to worry about keys that are too small to provide the required security, there are issues with keys that are too large. Denial-of-service attacks have been mounted with overly large keys or oddly sized keys. This has the potential to consume resources with these keys. It is highly recommended that checks on the key length be done before starting a cryptographic operation.

There are two reasonable ways to address this attack. First, a key should not be used for a cryptographic operation until it has been verified that it is controlled by a party trusted by the recipient. This approach means that no cryptography will be done until a trust decision about the key has been made, a process described in Appendix D, Item 4 of [RFC7515]. Second, applications can impose maximum- as well as minimum-length requirements on keys. This limits the resources that would otherwise be consumed by the use of overly large keys.

#### 6.2. RSASSA-PSS Security Considerations

There is a theoretical hash substitution attack that can be mounted against RSASSA-PSS [HASHID]. However, the requirement that the same hash function be used consistently for all operations is an effective mitigation against it. Unlike an Elliptic Curve Digital Signature Algorithm (ECDSA), hash function outputs are not truncated so that the full hash value is always signed. The internal padding structure of RSASSA-PSS means that one needs to have multiple collisions between the two hash functions to be successful in producing a forgery based on changing the hash function. This is highly unlikely.

#### 6.3. RSAES-OAEP Security Considerations

A version of RSAES-OAEP using the default parameters specified in Appendix A.2.1 of [RFC8017] is included because this is the most widely implemented set of OAEP parameter choices. (Those default parameters are the SHA-1 hash function and the MGF1 with SHA-1 mask generation function.)

Keys used with RSAES-OAEP MUST follow the constraints in Section 7.1 of [RFC8017]. Also, keys with a low private key exponent value, as described in Section 3 of "Twenty Years of Attacks on the RSA Cryptosystem" [Boneh99], MUST NOT be used.

### 7. References

#### 7.1. Normative References

- [Boneh99] Boneh, D., "Twenty Years of Attacks on the RSA Cryptosystem", Notices of the American Mathematical Society (AMS), Vol. 46, No. 2, pp. 203-213, 1999, <<http://www.ams.org/notices/199902/boneh.pdf>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <<https://www.rfc-editor.org/info/rfc7049>>.
- [RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <<https://www.rfc-editor.org/info/rfc7515>>.
- [RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <<https://www.rfc-editor.org/info/rfc8017>>.
- [RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <<https://www.rfc-editor.org/info/rfc8152>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 7.2. Informative References

- [HASHID] Kaliski, B., "On Hash Function Firewalls in Signature Schemes", Lecture Notes in Computer Science (LNCS), Volume 2271, pp. 1-16, DOI 10.1007/3-540-45760-7\_1, February 2002, <[https://rd.springer.com/chapter/10.1007/3-540-45760-7\\_1](https://rd.springer.com/chapter/10.1007/3-540-45760-7_1)>.
- [IANA.COSE] IANA, "CBOR Object Signing and Encryption (COSE)", <<http://www.iana.org/assignments/cose>>.
- [MultiPrimeRSA] Hinek, M. and D. Cheriton, "On the Security of Multi-prime RSA", June 2006, <<http://cacr.uwaterloo.ca/techreports/2006/cacr2006-16.pdf>>.

#### Acknowledgements

This specification incorporates text from "CBOR Encoded Message Syntax" (September 2015) authored by Jim Schaad and Brian Campbell. Thanks are due to Ben Campbell, Roni Even, Steve Kent, Kathleen Moriarty, Eric Rescorla, Adam Roach, Rich Salz, and Jim Schaad for their reviews of the specification.

#### Author's Address

Michael B. Jones  
Microsoft

Email: [mbj@microsoft.com](mailto:mbj@microsoft.com)  
URI: <http://self-issued.info/>