COSE Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 24, 2017

M. Jones
Microsoft

June 22, 2017

Using RSA Algorithms with COSE Messages
draft-jones-cose-rsa-05

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 for the use of RSASSA-PSS signatures, RSAES-OAEP encryption, and RSA keys are specified.

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 December 24, 2017.

Copyright Notice

Table of Contents

1. Introduction

The CBOR Object Signing and Encryption (COSE) [I-D.ietf-cose-msg] 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 RFC 2119 [RFC2119].

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 masking generation function (mgf), a hash function (h) and encoding parameters (P). For the algorithm identifiers defined in this section:

- mgf is always set to MGF1 from [RFC8017] and uses the same hash function as h.

- 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 crypto-systems, 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:

- For all keys, 'kty' MUST be present and MUST have a value of 3.
- For public keys, the fields 'n' and 'e' MUST be present. All other fields defined in the following table below MUST be absent.
- 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.
- 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.
- 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 mod (p - 1)
3	dQ	-7	bstr	dQ is d mod (q - 1)
3	qInv	-8	bstr	qInv is the CRT coefficient q^(-1) mod p
3	other	-9	array	other prime infos, an array
3	r_i	-10	bstr	a prime factor r_i of n, where i

				>= 3
3	d_i	-11	bstr	d_i = d mod (r_i - 1)
3	t_i	-12	bstr	the CRT coefficient t_i = (r_1 * r_2 * ... * r_(i-1))^(-1) mod r_i

Table 4: RSA Key Parameters

5. IANA Considerations

5.1. COSE Algorithms Registrations

This section registers the following values in the IANA "COSE Algorithms" registry [IANA.COSE].

- Name: PS256
- Value: -37
- Description: RSASSA-PSS w/ SHA-256
- Reference: Section 2 of [[this specification]]
- Recommended: Yes

- Name: PS384
- Value: -38
- Description: RSASSA-PSS w/ SHA-384
- Reference: Section 2 of [[this specification]]
- Recommended: Yes

- Name: PS512
- Value: -39
- Description: RSASSA-PSS w/ SHA-512
- Reference: Section 2 of [[this specification]]
- Recommended: Yes

- Name: RSAES-OAEP w/ RFC 8017 default parameters
- Value: -40
- Description: RSAES-OAEP w/ SHA-1
- Reference: Section 3 of [[this specification]]
- Recommended: Yes

- Name: RSAES-OAEP w/ SHA-256
- Value: -41
- Description: RSAES-OAEP w/ SHA-256

- Reference: Section 3 of [[this specification]]
- Recommended: Yes

- Name: RSAES-OAEP w/ SHA-512
- Value: -42
- Description: RSAES-OAEP w/ SHA-512
- Reference: Section 3 of [[this specification]]
- Recommended: Yes

5.2. COSE Key Type Registrations

This section registers the following values in the IANA "COSE Key Type" registry [IANA.COSE].

- Name: RSA
- Value: 3
- Description: RSA Key
- Reference: Section 4 of [[this specification]]

5.3. COSE Key Type Parameters Registrations

This section registers the following values in the IANA "COSE Key Type Parameters" registry [IANA.COSE].

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

- Key Type: 3
- Name: e

- Label: -2
- CBOR Type: bstr
- Description: the RSA public exponent e
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: d
- Label: -3

- CBOR Type: bstr
- Description: the RSA private exponent d
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: p
- Label: -4
- CBOR Type: bstr
- Description: the prime factor p of n
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: q
- Label: -5
- CBOR Type: bstr
- Description: the prime factor q of n
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: dP
- Label: -6
- CBOR Type: bstr
- Description: dP is d mod (p - 1)
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: dQ
- Label: -7
- CBOR Type: bstr
- Description: dQ is d mod (q - 1)
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: qInv
- Label: -8
- CBOR Type: bstr
- Description: qInv is the CRT coefficient q^(-1) mod p
- Reference: Section 4 of [[this specification]]

- Key Type: 3

- Name: other

- Label: -9
- CBOR Type: array
- Description: other prime infos, an array
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: r_i
- Label: -10
- CBOR Type: bstr
- Description: a prime factor r_i of n, where i >= 3
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: d_i
- Label: -11
- CBOR Type: bstr
- Description: d_i = d mod (r_i - 1)
- Reference: Section 4 of [[this specification]]

- Key Type: 3
- Name: t_i
- Label: -12
- CBOR Type: bstr
- Description: the CRT coefficient t_i = (r_1 * r_2 * ... * r_(i-1))^(-1) mod r_i
- Reference: Section 4 of [[this specification]]

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 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 RFC 8017 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 RFC 8017. 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://crypto.stanford.edu/~dabo/pubs/papers/RSA-survey.pdf>.

[I-D.ietf-cose-msg] Schaad, J., "CBOR Object Signing and Encryption (COSE)", draft-ietf-cose-msg-24 (work in progress), November 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc8017>.

7.2. Informative References

[HASHID] Kaliski, B., "On Hash Function Firewalls in Signature Schemes", Lecture Notes in Computer Science, 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>.

[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.

Appendix A. Acknowledgements

This specification incorporates text from draft-ietf-cose-msg-05 by Jim Schaad. 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.

Appendix B. Document History

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

-05

- Addressed IESG review comments.

- Updated the RFC 3447 reference to RFC 8017.
- Updated the field descriptions to use the wording from Section A.1.2 of RFC 8017.
- Corrected an error in the RSAES-OAEP security considerations.

-04

- Addressed SecDir review comments by Steve Kent and Gen-ART review comments by Roni Even.

-03

- Clarified the Security Considerations in ways suggested by Kathleen Moriarty.
- Acknowledged reviewers.

-02

- Reorganized the security considerations.
- Flattened the section structure.
- Applied wording improvements suggested by Jim Schaad.

-01

- Completed the sets of IANA registration requests.
- Revised the algorithm assignments based on those in draft-ietf-

cose-msg-24.

-00

- This specification addresses COSE issue #21: Restore RSA-PSS and the "RSA" key type. The initial version of this specification incorporates text from draft-ietf-cose-msg-05 -- the last COSE message specification version before the RSA algorithms were removed.

Author's Address

Michael B. Jones
Microsoft

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