COSE Working Group
Internet-Draft
Intended status: Informational
Expires: September 22, 2016

J. Schaad
August Cellars
March 21, 2016

CBOR Encoded Message Syntax: Additional Algorithms draft-schaad-cose-alg-01

Abstract

This document defines the identifiers and usage for a set of additional cryptographic algorithms in the CBOR Encoded Message (COSE) Syntax.

The algorithms setup in this docment are: RSA-PSS, RSA-OAEP, ... !!TBD!!

Contributing to this document

The source for this draft is being maintained in GitHub. Suggested changes should be submitted as pull requests at <https://github.com/cose-wg/cose-algs>. Instructions are on that page as well. Editorial changes can be managed in GitHub, but any substantial issues need to be discussed on the COSE mailing list.

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 September 22, 2016.

Copyright Notice

Table of Contents

1. Introduction

In the process of writing RFCXXXX [I-D.ietf-cose-msg] several algorithms were removed from that document to be addressed at a later date. This document deals with a large set of the cryptographic algorithms which were removed at that time.

This document provides the necessary conventions needed to use the algorithms defined in this document. This document additionally provides the necessary registration in the appropriate IANA registry tables.

1.1. Requirements Terminology

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 [RFC2119].

When the words appear in lower case, their natural language meaning is used.

1.2. Document Terminology

In this document we use the following terminology: [CREF1]

2. Signature Algorithms

This document defines two new signature algorithms: RSA-PSS and Edwards Curve Digital Signature Algorithm (EdDSA). Both of these signature algorithms are Signature Scheme with Appendix algorithms. (For a discussion on the difference between signature scheme with appendix and signature scheme with message recovery algorithms, see [I-D.ietf-cose-msg].)

2.1. RSASSA-PSS

The RSASSA-PSS signature algorithm is defined in [RFC3447].

The RSASSA-PSS signature algorithm is parametized 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 [RFC3447]. It has been recommended that the same hash function be used for hashing the data as well as in the mask generation function, for this specification we following 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 algorithms defined in this document can be found in Table 1.

name	value	hash	salt length	description
PS256	TBD1	SHA-256	32	RSASSA-PSS w/ SHA-256
PS384	TBD2	SHA-384	48	RSASSA-PSS w/ SHA-384
PS512	TBD3	SHA-512	64	RSASSA-PSS w/ SHA-512

Table 1: RSASSA-PSS Algorithm Values

2.1.1. Security Considerations

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. This has the potential to consume resources with potentially bad keys. There are two reasonable ways to address this attack. First, a key should not be used for a cryptographic operation until it has been matched back to an authorized user. This approach means that no cryptography would be done except for authorized users. Second, applications can impose maximum as well as minimum length requirements on keys. This limits the resources consumed even if the matching is not performed until the cryptography has been done.

There is a theoretical hash substitution attack that can be mounted against RSASSA-PSS. However, the requirement that the same hash function be used consistently for all operations is an effective mitigation against it. Unlike ECDSA, hash functions 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 in order to be successful in producing a forgery based on changing the hash function. This is highly unlikely.

2.2. Edwards-curve Digital Signature Algorithms (EdDSA)

[I-D.irtf-cfrg-eddsa] describes the elliptic curve signature scheme Edwards-curve Digital Signature Algorithm (EdDSA). In that document, the signature algorithm is instantiated using parameters for edwards25519 and edwards448 curves. The document additionally

describes two variants of the EdDSA algorithm: Pure EdDSA, where no hash function is applied to the content before signing and, HashEdDSA where a hash function is applied to the content before signing and the result of that hash function is signed. For use with COSE, on the pure EdDSA version is used. This is because it is not expected that extremely large contents are going to be needed and, based on the arrangement of the message structure, the entire message is going to need to be held in memory in order to create or verify a signature. Thus, the use of an incremental update process would not be useful. Applications can provide the same features by defining the content of the message as a hash value and transporting the COSE message and the content as separate items.

The algorithms defined in this document can be found in Table 2. A single signature algorithm is defined which can be used for multiple curves.

name	value	description
EdDSA	*	EdDSA

Table 2: EdDSA Algorithm Values

[I-D.irtf-cfrg-eddsa] describes the method of encoding the signature value.

When using a COSE key for this algorithm the following checks are made:

- The 'kty' field MUST be present and it MUST be 'OKP'.
- The 'crv' field MUST be present, and it MUST be a curve defined for this signature algorithm.
- If the 'alg' field is present, it MUST match 'EdDSA'.
- If the 'key_ops' field is present, it MUST include 'sign' when creating an EdDSA signature.
- If the 'key_ops' field is present, it MUST include 'verify' when verifying an EdDSA signature.

3. Message Authentication (MAC) Algorithms

This document defines no new Message Authentication Code algorithms.

4. Content Encryption Algorithms

This document defines no new content inception algorithms.

5. Key Derivation Functions (KDF)

This document defines new new key derivation functions.

6. Recipient Algorithms

6.1. RSAES-OAEP

RSAES-OAEP is an asymmetric key encryption algorithm. The defintion of RSAEA-OAEP can be find in Section 7.1 of [RFC3447]. 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 MFG1 from [RFC3447] and uses the same hash function as h.

- P is always set to the empty octet string.

Table 3 summarizes the rest of the values.

name	value	hash	description
RSAES-OAEP w/SHA-256	-25	SHA-256	RSAES OAEP w/ SHA-256
RSAES-OAEP w/SHA-512	-26	SHA-512	RSAES OAEP w/ SHA-512

Table 3: RSAES-OAEP Algorithm Values

The key type MUST be 'RSA'.

6.1.1. Security Considerations for RSAES-OAEP

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.

It is highly recommended that checks on the key length be done before starting a decryption operation. One potential denial of service operation is to provide encrypted objects using either abnormally long or oddly sized RSA modulus values. 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.

6.2. ECDH

The algorithm ECDH is defined for use in COSE in [I-D.ietf-cose-msg]. In this document the algorithm is extended to be used with the two curves defined in [I-D.irtf-cfrg-curves].

The following updates [I-D.ietf-cose-msg] sections 12.4.1 and 12.5.1.

- OLD: The 'kty' field MUST be present and it MUST be 'EC2'.

- NEW: The 'kty' field MUST be present and it MUST be 'EC2' or 'OKP'.

All the rest of the checks remain the same.

7. Keys

The COSE_Key object defines a way to hold a single key object, it is still required that the members of individual key types be defined. This section of the document is where we define an initial set of members for specific key types.

For each of the key types, we define both public and private members. The public members are what is transmitted to others for their usage. We define private members mainly for the purpose of archival of keys by individuals. However, there are some circumstances where private keys may be distributed by various entities in a protocol. Examples include: Entities which have poor random number generation. Centralized key creation for multi-cast type operations. Protocols where a shared secret is used as a bearer token for authorization purposes.

Key types are identified by the 'kty' member of the COSE_Key object. In this document we define four values for the member.

name	value	description
OPK	TBDXX	Octet Key Pair
RSA	TBDXX1	RSA Keys

Table 4: Key Type Values

7.1. Octet Key Pair

A new key type is defined for Octet Key Pairs (OKP). Do not assume that keys using this type are elliptic curves. This key type could be used for other curve types (for example mathematics based on hyper-elliptic surfaces).

The key parameters defined in this section are summarized in Table 5. The members that are defined for this key type are:

crv contains an identifier of the curve to be used with the key. [CREF2] The curves defined in this document for this key type can be found in Table 6. Other curves may be registered in the future and private curves can be used as well.

x contains the x coordinate for the EC point. The octet string represents a little-endian encoding of x.

d contains the private key.

For public keys, it is REQUIRED that 'crv' and 'x' be present in the structure. For private keys, it is REQUIRED that 'crv' and 'd' be present in the structure. For private keys, it is RECOMMENDED that 'x' also be present, but it can be recomputed from the required elements and omitting it saves on space.

name	key type	value	type	description
crv	1	-1	int / tstr	EC Curve identifier - Taken from the COSE General Registry
x	1	-2	bstr	X Coordinate
d	1	-4	bstr	Private key

Table 5: EC Key Parameters

name	key type	value	description
Curve25519	EC1	TBDYY1	Curve 25519
Curve448	EC1	TBDYY2	Curve 448

Table 6: EC Curves

7.2. RSA Keys

This document defines a key structure for both the public and private halves of RSA keys. Together, an RSA public key and an RSA private key form an RSA key pair. [CREF3]

The document also provides support for the so-called "multi-prime" RSA where 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 [RFC3447] for the naming of the fields of an RSA public or private key. The table Table 7 provides a summary of the label values and the types associated with each of those labels. 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 Table 7 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.

name	key type	value	type	description
n	3	-1	bstr	Modulus Parameter
e	3	-2	int	Exponent Parameter
d	3	-3	bstr	Private Exponent Parameter
p	3	-4	bstr	First Prime Factor
q	3	-5	bstr	Second Prime Factor
dP	3	-6	bstr	First Factor CRT Exponent
dQ	3	-7	bstr	Second Factor CRT Exponent
qInv	3	-8	bstr	First CRT Coefficient
other	3	-9	array	Other Primes Info
r_i	3	-10	bstr	i-th factor, Prime Factor
d_i	3	-11	bstr	i-th factor, Factor CRT Exponent
t_i	3	-12	bstr	i-th factor, Factor CRT Coefficient

Table 7: RSA Key Parameters

8. IANA Considerations

8.1. COSE Header Parameter Registry

There are currently no registration requests here

8.2. COSE Header Algorithm Label Table

TBD

8.3. COSE Algorithm Registry

TBD

8.4. COSE Key Common Parameter Registry

There are currently no registration tasks inthis section.

8.5. COSE Key Type Parameter Registry

It is requested that IANA create a new registry "COSE Key Type Parameters".

The columns of the table are:

key type This field contains a descriptive string of a key type. This should be a value that is in the COSE General Values table and is placed in the 'kty' field of a COSE Key structure.

name This is a descriptive name that enables easier reference to the item. It is not used in the encoding.

label The label is to be unique for every value of key type. The range of values is from -256 to -1. Labels are expected to be reused for different keys.

CBOR type This field contains the CBOR type for the field

description This field contains a brief description for the field

specification This contains a pointer to the public specification for the field if one exists

This registry will be initially populated by the values in Table 5, and Table 7. The specification column for all of these entries will be this document.

8.6. COSE Elliptic Curve Registry

It is requested that IANA create a new registry "COSE Elliptic Curve Parameters".

The columns of the table are:

name This is a descriptive name that enables easier reference to the item. It is not used in the encoding.

value This is the value used to identify the curve. These values MUST be unique. The integer values from -256 to 255 are designated as Standards Track Document Required. The the integer values from 256 to 65535 and -65536 to -257 are designated as Specification Required. Integer values over 65535 are designated as first come first serve. Integer values less than -65536 are marked as private use.

key type This designates the key type(s) that can be used with this curve.

description This field contains a brief description of the curve.

specification This contains a pointer to the public specification for the curve if one exists.

This registry will be initially populated by the values in Table 4. The specification column for all of these entries will be this document.

9. Security Considerations

There are security considerations:

1. Protect private keys

2. MAC messages with more than one recipient means one cannot figure out who sent the message

3. Use of direct key with other recipient structures hands the key to other recipients.

4. Use of direct ECDH direct encryption is easy for people to leak information on if there are other recipients in the message.

5. Considerations about protected vs unprotected header fields.

6. Need to verify that: 1) the kty field of the key matches the key and algorithm being used. 2) that the kty field needs to be included in the trust decision as well as the other key fields. 3) that the algorithm be included in the trust decision.

10. References

10.1. Normative References

[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>.

10.2. Informative References

[AES-GCM] Dworkin, M., "NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.", Nov 2007.

[DSS] U.S. National Institute of Standards and Technology, "Digital Signature Standard (DSS)", July 2013.

[I-D.greevenbosch-appsawg-cbor-cddl] Vigano, C. and H. Birkholz, "CBOR data definition language (CDDL): a notational convention to express CBOR data structures", draft-greevenbosch-appsawg-cbor-cddl-07 (work in progress), October 2015.

[I-D.ietf-cose-msg] Schaad, J., "CBOR Encoded Message Syntax", draft-ietf-cose-msg-10 (work in progress), February 2016.

[I-D.irtf-cfrg-curves] Langley, A. and M. Hamburg, "Elliptic Curves for Security", draft-irtf-cfrg-curves-11 (work in progress), October 2015.

[I-D.irtf-cfrg-eddsa] Josefsson, S. and I. Liusvaara, "Edwards-curve Digital Signature Algorithm (EdDSA)", draft-irtf-cfrg-eddsa-05 (work in progress), March 2016.

[MAC] NiST, N., "FIPS PUB 113: Computer Data Authentication", May 1985.

[MultiPrimeRSA] Hinek, M. and D. Cheriton, "On the Security of Multi-prime RSA", June 2006.

[PVSig] Brown, D. and D. Johnson, "Formal Security Proofs for a Signature Scheme with Partial Message Recover", February 2000.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <http://www.rfc-editor.org/info/rfc2104>.

[RFC2633] Ramsdell, B., Ed., "S/MIME Version 3 Message Specification", RFC 2633, DOI 10.17487/RFC2633, June 1999, <http://www.rfc-editor.org/info/rfc2633>.

[RFC2898] Kaliski, B., "PKCS #5: Password-Based Cryptography Specification Version 2.0", RFC 2898, DOI 10.17487/RFC2898, September 2000, <http://www.rfc-editor.org/info/rfc2898>.

[RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, DOI 10.17487/RFC3394, September 2002, <http://www.rfc-editor.org/info/rfc3394>.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, DOI 10.17487/RFC3447, February 2003, <http://www.rfc-editor.org/info/rfc3447>.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, DOI 10.17487/RFC3610, September 2003, <http://www.rfc-editor.org/info/rfc3610>.

[RFC4231] Nystrom, M., "Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512", RFC 4231, DOI 10.17487/RFC4231, December 2005, <http://www.rfc-editor.org/info/rfc4231>.

[RFC4262] Santesson, S., "X.509 Certificate Extension for Secure/Multipurpose Internet Mail Extensions (S/MIME) Capabilities", RFC 4262, DOI 10.17487/RFC4262, December 2005, <http://www.rfc-editor.org/info/rfc4262>.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <http://www.rfc-editor.org/info/rfc5480>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <http://www.rfc-editor.org/info/rfc5652>.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, DOI 10.17487/RFC5751, January 2010, <http://www.rfc-editor.org/info/rfc5751>.

[RFC5752] Turner, S. and J. Schaad, "Multiple Signatures in Cryptographic Message Syntax (CMS)", RFC 5752, DOI 10.17487/RFC5752, January 2010, <http://www.rfc-editor.org/info/rfc5752>.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <http://www.rfc-editor.org/info/rfc5869>.

[RFC5990] Randall, J., Kaliski, B., Brainard, J., and S. Turner, "Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax (CMS)", RFC 5990, DOI 10.17487/RFC5990, September 2010, <http://www.rfc-editor.org/info/rfc5990>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI 10.17487/RFC6090, February 2011, <http://www.rfc-editor.org/info/rfc6090>.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <http://www.rfc-editor.org/info/rfc6151>.

[RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August 2013, <http://www.rfc-editor.org/info/rfc6979>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

[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>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <http://www.rfc-editor.org/info/rfc7516>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <http://www.rfc-editor.org/info/rfc7517>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <http://www.rfc-editor.org/info/rfc7518>.

[RFC7539] Nir, Y. and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols", RFC 7539, DOI 10.17487/RFC7539, May 2015, <http://www.rfc-editor.org/info/rfc7539>.

[SEC1] Standards for Efficient Cryptography Group, "SEC 1: Elliptic Curve Cryptography", May 2009.

[SP800-56A] Barker, E., Chen, L., Roginsky, A., and M. Smid, "NIST Special Publication 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography", May 2013.

Appendix A. Document Updates

A.1. Version -00

- TBD

Editorial Comments

[CREF1] JLS: I have not gone through the document to determine what needs to be here yet. We mostly want to grab terms which are used in unusual ways or are not generally understood.

[CREF2] JLS: Is is the same registry for both OKP and EC2?

[CREF3] JLS: Looking at the CBOR specification, the bstr that we are looking in our table below should most likely be specified as big numbers rather than as binary strings. This means that we would use the tag 6.2 instead. From my reading of the specification, there is no difference in the encoded size of the

resulting output. The specification of bignum does explicitly allow for integers encoded with leading zeros.

Author's Address

Jim Schaad
August Cellars

Email: ietf@augustcellars.com