Workgroup: Network Working Group
Internet-Draft:
draft-rundgren-deterministic-cbor-17
Published: 16 August 2023
Intended Status: Informational
Expires: 17 February 2024
Authors: A. Rundgren, Ed.
Independent

Deterministically Encoded CBOR (D-CBOR)

Abstract

This document describes a deterministic encoding scheme for CBOR intended for usage in high-end computing platforms like mobile phones, Web browsers, and Web servers. In addition to enhancing interoperability, deterministic encoding also enables performing cryptographic operations like signing "raw" CBOR data items, something which otherwise would require wrapping such data in byte strings, or introduce dependencies on non-standard canonicalization procedures.

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 https://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 17 February 2024.

Copyright Notice

Table of Contents

1. Introduction

This specification introduces a deterministic encoding scheme for data expressed in the CBOR [RFC8949] format. This scheme is subsequently referred to as D-CBOR.

Note that this document is not on the IETF standards track. However, a conformant implementation is supposed to adhere to the specified behavior for security and interoperability reasons.

1.1. Background

[RFC8949] supports a number of deterministic encoding options. Some of these options are not necessarily interoperable, like Rule 1-3 in Section 4.2.2. This could in turn hamper large scale rollout of applications depending on deterministically encoded CBOR.

1.2. Objectives

The main objective of D-CBOR is providing an interoperable CBOR encoding profile, *primarily* targeting high-end computing platforms like mobile phones, Web browsers, and Web servers. In addition, due to the underpinning deterministic representation of data, D-CBOR also enables performing cryptographic operations like signatures over "raw" (unwrapped) CBOR data items since signatures depend on a

unified representation of the data to be signed. Furthermore, building on the same foundation, D-CBOR also permits decoded CBOR data to be subjected to simple and secure *transformation* and *reencoding* operations.

The deterministic encoding scheme described in this document is characterized by being *bidirectional* also when CBOR is provided in *diagnostic notation* (Section 8 of [RFC8949]), making D-CBOR comparatively easy to understand, debug, and implement.

Although this document specifies a *deterministic* encoding scheme, the intent is that the encoding scheme should be equally useful for applications that do not depend on this particular feature.

In spite of the enhanced functionality, this specification retains full compatibility with [RFC8949].

See also [I-D.mcnally-deterministic-cbor] which represents an alternative approach to deterministic encoding.

1.3. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Specification

The deterministic encoding scheme used by D-CBOR builds on Section 4.2 of [RFC8949].

The following sections contain some additional clarifications and explicit choices, in order to facilitate an interoperable encoding scheme.

2.1. General Considerations

Occurrences of unknown or malformed CBOR data items MUST be rejected.

Map keys MUST only be compared and sorted based on their bytewise lexicographic order of their deterministic encoding.

For applications that depend on *deterministic reencoding* of CBOR data items, compliant decoder implementations MUST be able to recreate such data in its original form. This means for example that the string component of date items (tag 0) MUST be preserved "as is" in order to maintain consistency.

The *optional* numerical extensions described in Section 3.4.4 of [RFC8949] MUST be treated as *distinct* data items as well as not be subjected to any transformations at the encoding level.

2.2. CBOR Data Items

A compliant D-CBOR implementation SHOULD as a *minimum* support the following CBOR data items:

Data Item	Encoding
integer	Major type 0 and 1
bignum	0xc2 and 0xc3
floating point	0xf9, 0xfa and 0xfb
byte string	Major type 2
text string	Major type 3
false	0xf4
true	0xf5
null	0xf6
array	Major type 4
map	Major type 5
tag	Major type 6

Table 1: CBOR Data Items

See also Appendix A.

2.3. Encoding of Numbers

To achieve a *fixed and bidirectional* representation of numbers, Rule 2 in Section 4.2.2 of [RFC8949] MUST be adhered to.

Note that integer and floating point data items MUST use preferred serialization as described in Section 4.2.1 of [RFC8949].

The following sub sections hold examples of numeric values expressed in *diagnostic notation* (Section 8 of [RFC8949]) and their D-CBOR encoded counterpart (expressed in hexadecimal).

Note that the values and encodings are supposed to work in *both* directions.

2.3.1. Integer Numbers

The following table holds a set of integers. Note that bignum data items MUST use preferred serialization as described in Section 3.4.3 of [RFC8949].

Value	Encoding
0	00
-1	20
23	17
24	1818
-24	37
-25	3818
255	18ff
256	190100
-256	38ff
-257	390100
65535	19ffff
65536	1a00010000
1099511627775	1b000000ffffffffff
18446744073709551615	1bffffffffffffffff
18446744073709551616	c249010000000000000000
-18446744073709551616	3bffffffffffffffff
-18446744073709551617	c349010000000000000000

Table 2: Integer Numbers

2.3.2. Special Floating Point Numbers

The following table holds the set of special IEEE 754 [IEEE754] values. Note that NaN "signaling" MUST be flagged as an error.

Value	Encoding
0.0	f90000
-0.0	f98000
Infinity	f97c00
-Infinity	f9fc00
NaN	f97e00

Table 3: Special Floating Point Numbers

2.3.3. "Ordinary" Floating Point Numbers

The following table holds a set of "ordinary" IEEE 754 [IEEE754] values including some edge cases. Note that subnormal floating point values MUST be supported.

Value	Encoding
-5.960464477539062e-8	fbbe6fffffffffffff
-5.9604644775390625e-8	f98001
-5.960464477539064e-8	fbbe70000000000001
-5.960465188081798e-8	fab3800001

Value	Encoding
0.00006097555160522461	f903ff
65504.0	f97bff
65504.00390625	fa477fe001
65536.0	fa47800000
10.559998512268066	fa4128f5c1
10.559998512268068	fb40251eb820000001
3.4028234663852886e+38	fa7f7fffff
3.402823466385289e+38	fb47effffe0000001
1.401298464324817e-45	fa00000001
1.1754942106924411e-38	fa007fffff
5.0e-324	fb0000000000000001
-1.7976931348623157e+308	fbffeffffffffffff

Table 4: "Ordinary" Floating Point Numbers

3. IANA Considerations

This document has no IANA actions.

4. Security Considerations

This specification inherits all the security considerations of CBOR [RFC8949].

Applications that exploit the uniqueness of deterministic encoding should verify that the used decoder actually flags incorrectly formatted CBOR data items.

5. References

5.1. Normative References

[IEEE754] IEEE, "IEEE Standard for Floating-Point Arithmetic", IEEE 754-2019, DOI 10.1109/IEEESTD.2019.8766229, <https://ieeexplore.ieee.org/document/8766229>.

[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>.

[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>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

5.2. Informative References

[I-D.mcnally-deterministic-cbor] McNally, W. and C. Allen, "Gordian dCBOR: Deterministic CBOR Implementation Practices", Work in Progress, Internet-Draft, draft-mcnally-deterministic-cbor-01, 4 May 2023, <https://datatracker.ietf.org/doc/html/draft-mcnally-deterministic-cbor-01>.

Appendix A. Implementation Constraints

This section is informative.

Note that even if an application does not support (or need) bignum or floating point data items, you can still use D-CBOR since a strict subset is upwardly compatible with full-blown implementations. Low-end platforms typically also restrict CBOR map keys to integer and text string data items. Since these issues are application specific, they are out of scope for this specification.

Appendix B. Reference Implementations

This section is informative.

Reference implementations that conform to this specification include:

- JavaScript: <https://github.com/cyberphone/CBOR.js#cborjs>
- JDK 17+: <https://github.com/cyberphone/openkeystore#cbor-support>
- Android/Java: <https://github.com/cyberphone/android-cbor#cbor-for-android>

Appendix C. Online Tools

This section is informative.

The following online tools enable testing D-CBOR without installing any software:

- <https://cyberphone.github.io/CBOR.js/doc/playground.html>
- <https://test.webpki.org/csf-lab/convert>

Acknowledgements

This document incorporates much appreciated suggestions and feedback by Eliot Lear, and Carsten Bormann.

Document History

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

Version 00:

*Initial publication.

Version 01:

*Added Table 1: Supported CBOR Data Types

Version 02:

*Added bidirectional + reencoding to 2

Version 03:

*Added ref to 3.4.4. Decimal Fractions and Bigfloats.

*Type => Data Item (throughout the spec).

Version 04-00:

*Document name spelling error.

Version 01:

*Minor tweaks.

Version 02:

*ISE submission and associated changes.

Version 03:

*Number table clarifications.

Version 04:

*Word-smithing.

Version 05:

*ISE input resulted in Background section.

Version 06:

*Word-smithing.

Version 07:

*Word-smithing.

Version 08:

*Explained universality.

Version 09:

*Stream added.

Version 10:

*External "Section" refs made into links.

Version 11:

*IEEE 754 ref.

Version 12:

*Language nit.

Version 13:

*Major restructuring of "Specification".

Version 14:

*Word-smithing.

Version 15:

*Word-smithing.

Version 16:

*Added section references to RFC8949 for numbers.

Version 17:

*Acknowledgements.

Author's Address

Anders Rundgren (editor)
Independent
Montpellier
France

Email: anders.rundgren.net@gmail.com
URI: https://www.linkedin.com/in/andersrundgren/