

Workgroup: Remote ATtestation Procedures

Internet-Draft: draft-ftbs-rats-msg-wrap-02

Published: 7 March 2023

Intended Status: Standards Track

Expires: 8 September 2023

Authors: H. Birkolz            N. Smith    T. Fossati    H. Tschofenig  
          Fraunhofer SIT    Intel            arm

## **RATS Conceptual Messages Wrapper**

### **Abstract**

This document defines two encapsulation formats for RATS conceptual messages (i.e., evidence, attestation results, endorsements and reference values.)

The first format uses a CBOR or JSON array with two members: one for the type, another for the value. The other format wraps the value in a CBOR byte string and prepends a CBOR tag to convey the type information.

### **Discussion Venues**

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Remote ATtestation ProcedureS Working Group mailing list (rats@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/rats/>.

Source for this draft and an issue tracker can be found at <https://github.com/thomas-fossati/draft-ftbs-rats-msg-wrap>.

### **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 8 September 2023.

## Copyright Notice

Copyright (c) 2023 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

- [1. Introduction](#)
- [2. Conventions and Definitions](#)
- [3. Conceptual Message Wrapper Encodings](#)
  - [3.1. CMW Array](#)
  - [3.2. CMW CBOR Tags](#)
    - [3.2.1. Use of Pre-existing CBOR Tags](#)
  - [3.3. Decapsulation Algorithm](#)
- [4. Examples](#)
  - [4.1. JSON Array](#)
  - [4.2. CBOR Array](#)
  - [4.3. CBOR Tag](#)
- [5. Registering a Media Type for Evidence](#)
- [6. Security Considerations](#)
- [7. IANA Considerations](#)
- [8. References](#)
  - [8.1. Normative References](#)
  - [8.2. Informative References](#)
- [Appendix A. RFC9193 ABNF](#)
- [Appendix B. Open Issues](#)
- [Acknowledgments](#)
- [Authors' Addresses](#)

## 1. Introduction

The RATS architecture defines a handful of conceptual messages (see [Section 8](#) of [\[RFC9334\]](#)), such as evidence and attestation results. Each conceptual message can have multiple claims encoding and serialization formats ([Section 9](#) of [\[RFC9334\]](#)). Such serialized messages may have to be transported via different protocols - for example, evidence using an EAT [\[I-D.ietf-rats-eat\]](#) encoding serialized as a CBOR payload in a "background check" topological arrangement, or attestation results as Attestation Results for

Secure Interactions (AR4SI) [[I-D.ietf-rats-ar4si](#)] payloads in "passport" mode.

In order to minimize the cost associated with registration and maximize interoperability, it is desirable to reuse their typing information across such boundaries.

This document defines two encapsulation formats for RATS conceptual messages that aim to achieve the goals stated above.

These encapsulation formats are designed to be:

- \*Self-describing - which removes the dependency on the framing provided by the embedding protocol (or the storage system) to convey exact typing information.

- \*Based on media types [[RFC6838](#)] - which allows amortising their registration cost across many different usage scenarios.

A protocol designer could use these formats, for example, to convey evidence, endorsements or reference values in certificates and CRLs extensions ([\[DICE-arch\]](#)), to embed attestation results or evidence as first class authentication credentials in TLS handshake messages [[I-D.fossati-tls-attestation](#)], to transport attestation-related payloads in RESTful APIs, or for stable storage of attestation results in form of file system objects.

## 2. Conventions and Definitions

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.

In this document, CDDL [[RFC8610](#)] [[RFC9165](#)] is used to describe the data formats.

The reader is assumed to be familiar with the vocabulary and concepts defined in [[RFC9334](#)].

This document reuses the terms defined in [Section 2](#) of [[RFC9193](#)] (e.g., "Content-Type").

## 3. Conceptual Message Wrapper Encodings

Two types of RATS Conceptual Message Wrapper (CMW) are specified in this document:

1. A CMW using a CBOR or JSON array ([Section 3.1](#));

2. A CMW based on CBOR tags ([Section 3.2](#)).

### 3.1. CMW Array

The CMW array format is defined in [Figure 1](#). (To improve clarity, the Content-Type ABNF is defined separately in [Appendix A](#).)

```
cmw-array = cmw-array-cbor / cmw-array-json

cmw-array-cbor = [ type, bytes ]

cmw-array-json = [ type, base64-string ]

type = coap-content-format / media-type
coap-content-format = uint .size 2
media-type = text .abnf ("Content-Type" .cat Content-Type-ABNF)

base64-string = text .regexp "[A-Za-z0-9_-]+"
```

Figure 1: CDDL definition of the Array format

It is composed of two members:

**type:**

Either a text string representing a Content-Type (e.g., an EAT media type [[I-D.ietf-rats-eat-media-type](#)]) or an unsigned Integer corresponding to a CoAP Content-Format number ([Section 12.3](#) of [[RFC7252](#)]).

**value:**

The RATS conceptual message serialized according to the value defined in the type member.

A CMW array can be encoded as CBOR [[STD94](#)] or JSON [[RFC8259](#)].

When using JSON, the value field is encoded as Base64 using the URL and filename safe alphabet (Section 5 of [[RFC4648](#)]) without padding.

When using CBOR, the value field is encoded as a CBOR byte string.

### 3.2. CMW CBOR Tags

CBOR Tags used as CMW are derived from CoAP Content-Format numbers. If a CoAP content format exists for a RATS conceptual message, the TN() transform defined in [Appendix B](#) of [[RFC9277](#)] can be used to derive a corresponding CBOR tag in range [1668546817, 1668612095].

The RATS conceptual message is first serialized according to the Content-Format number associated with the CBOR tag and then encoded as a CBOR byte string, to which the tag is prepended.

The CMW CBOR Tag is defined in [Figure 2](#).

```
cmw-cbor-tag<bytes> = #6.<coap-cf-tag-number>(bytes)
```

```
coap-cf-tag-number = 1668546817..1668612095
```

Figure 2: CDDL definition of the CBOR Tag format

### 3.2.1. Use of Pre-existing CBOR Tags

If a CBOR tag has been registered in association with a certain RATS conceptual message independently of a CoAP content format (i.e., it is not obtained by applying the TN() transform), it can be readily used as an encapsulation without the extra processing described in [Section 3.2](#).

A consumer can always distinguish tags that have been derived via TN(), which all fall in the [1668546817, 1668612095] range, from tags that are not, and therefore apply the right decapsulation on receive.

### 3.3. Decapsulation Algorithm

After removing any external framing (for example, the ASN.1 OCTET STRING if the CMW is carried in a certificate extension [[DICE-arch](#)]), the CMW decoder does a 1-byte lookahead, as illustrated in the following pseudo code, to decide how to decode the remainder of the byte buffer:

```
func CMWDecode(b []byte) (CMW, error) {
    if len(b) < CMWMinSize {
        return CMW{}, errors.New("CMW too short")
    }

    switch b[0] {
    case 0x82:
        return cborArrayDecode(b)
    case 0x5b:
        return jsonArrayDecode(b)
    default:
        return cborTagDecode(b)
    }
}
```

## 4. Examples

The (equivalent) examples below assume the Media-Type-Name `application/vnd.example.rats-conceptual-msg` has been registered alongside a corresponding CoAP Content-Format number 30001. The CBOR tag 1668576818 is derived applying the TN() transform as described in [Section 3.2](#).

### 4.1. JSON Array

```
[  
  "application/vnd.example.rats-conceptual-msg",  
  "q82rzQ"  
]
```

### 4.2. CBOR Array

```
[  
  30001,  
  h'abcdabcd'  
]
```

with the following wire representation:

```
82          # array(2)  
19 7531     # unsigned(30001)  
44          # bytes(4)  
  abcdabcd # "\xAB\xCD"
```

### 4.3. CBOR Tag

```
1668576818(h'abcdabcd')
```

with the following wire representation:

```
da 63747632 # tag(1668576818)  
44          # bytes(4)  
  abcdabcd # "\xAB\xCD"
```

## 5. Registering a Media Type for Evidence

Note: Not sure whether this advice should go.

When registering a new media type for evidence, in addition to its syntactical description, the author **SHOULD** provide a public and stable description of the signing and appraisal procedures associated with the data format.

## 6. Security Considerations

This document defines two encapsulation formats for RATS conceptual messages. The messages themselves and their encoding ensure security protection. For this reason there are no further security requirements raised by the introduction of this encapsulation.

Changing the encapsulation of a payload by an adversary will result in incorrect processing of the encapsulated messages and this will subsequently lead to a processing error.

## 7. IANA Considerations

This document does not make any requests to IANA.

## 8. References

### 8.1. 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, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/rfc/rfc4648>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/rfc/rfc6838>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/rfc/rfc7252>>.
- [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/rfc/rfc8174>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/rfc/rfc8259>>.
- [RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and

JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/rfc/rfc8610>>.

- [RFC9165] Bormann, C., "Additional Control Operators for the Concise Data Definition Language (CDDL)", RFC 9165, DOI 10.17487/RFC9165, December 2021, <<https://www.rfc-editor.org/rfc/rfc9165>>.
- [RFC9277] Richardson, M. and C. Bormann, "On Stable Storage for Items in Concise Binary Object Representation (CBOR)", RFC 9277, DOI 10.17487/RFC9277, August 2022, <<https://www.rfc-editor.org/rfc/rfc9277>>.
- [STD94] 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/rfc/rfc8949>>.

## 8.2. Informative References

- [DICE-arch] Trusted Computing Group, "DICE Attestation Architecture", March 2021, <<https://trustedcomputinggroup.org/wp-content/uploads/DICE-Attestation-Architecture-r23-final.pdf>>.
- [I-D.fossati-tls-attestation] Tschofenig, H., Fossati, T., Howard, P., Mihalcea, I., and Y. Deshpande, "Using Attestation in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", Work in Progress, Internet-Draft, draft-fossati-tls-attestation-02, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-fossati-tls-attestation-02>>.
- [I-D.ietf-rats-ar4si] Voit, E., Birkholz, H., Hardjono, T., Fossati, T., and V. Scarlata, "Attestation Results for Secure Interactions", Work in Progress, Internet-Draft, draft-ietf-rats-ar4si-04, 2 March 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-ar4si-04>>.
- [I-D.ietf-rats-eat] Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", Work in Progress, Internet-Draft, draft-ietf-rats-eat-19, 19 December 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat-19>>.
- [I-D.ietf-rats-eat-media-type] Lundblade, L., Birkholz, H., and T. Fossati, "EAT Media Types", Work in Progress, Internet-Draft, draft-ietf-rats-eat-media-type-01, 19 October



2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat-media-type-01>>.

[RFC9193] Keränen, A. and C. Bormann, "Sensor Measurement Lists (SenML) Fields for Indicating Data Value Content-Format", RFC 9193, DOI 10.17487/RFC9193, June 2022, <<https://www.rfc-editor.org/rfc/rfc9193>>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedures (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <<https://www.rfc-editor.org/rfc/rfc9334>>.

## Appendix A. RFC9193 ABNF

; from RFC9193

Content-Type-ABNF = '

Content-Type = Media-Type-Name \*( \*SP ";" \*SP parameter )  
parameter = token "=" ( token / quoted-string )

token = 1\*tchar  
tchar = "!" / "#" / "\$" / "%" / "&" / "\'" / "\*" / "+" / "-" / "." / "^" / "\_" / "`" / "|" / "~" / DIGIT / ALPHA

quoted-string = %x22 \*( qdtext / quoted-pair ) %x22  
qdtext = SP / %x21 / %x23-5B / %x5D-7E  
quoted-pair = "\" ( SP / VCHAR )

Media-Type-Name = type-name "/" subtype-name

type-name = restricted-name  
subtype-name = restricted-name

restricted-name = restricted-name-first \*126restricted-name-chars  
restricted-name-first = ALPHA / DIGIT  
restricted-name-chars = ALPHA / DIGIT / "!" / "#" / "\$" / "&" / "-" / "^" / "\_"  
restricted-name-chars =/ "." ; Characters before first dot always  
; specify a facet name  
restricted-name-chars =/ "+" ; Characters after last plus always  
; specify a structured syntax suffix

DIGIT = %x30-39 ; 0 - 9  
POS-DIGIT = %x31-39 ; 1 - 9  
ALPHA = %x41-5A / %x61-7A ; A - Z / a - z  
SP = %x20  
VCHAR = %x21-7E ; printable ASCII (no SP)  
,

## **Appendix B. Open Issues**

Note to RFC Editor: please remove before publication.

The list of currently open issues for this documents can be found at <https://github.com/thomas-fossati/draft-ftbs-rats-msg-wrap/issues>.

## **Acknowledgments**

The authors would like to thank Carl Wallace and Carsten Bormann for their reviews and suggestions.

## **Authors' Addresses**

Henk Birkolz  
Fraunhofer SIT

Email: [henk.birkholz@sit.fraunhofer.de](mailto:henk.birkholz@sit.fraunhofer.de)

Ned Smith  
Intel

Email: [ned.smith@intel.com](mailto:ned.smith@intel.com)

Thomas Fossati  
arm

Email: [thomas.fossati@arm.com](mailto:thomas.fossati@arm.com)

Hannes Tschofenig

Email: [hannes.tschofenig@gmx.net](mailto:hannes.tschofenig@gmx.net)