Internet Engineering Task Force (IETF)
Request for Comments: 9782
Category: Standards Track
ISSN: 2070-1721

L. Lundblade
Security Theory LLC
H. Birkholz
Fraunhofer SIT
T. Fossati
Linaro

May 2025

Entity Attestation Token (EAT) Media Types

Abstract

The payloads used in Remote ATtestation procedureS (RATS) may require an associated media type for their conveyance, for example, when the payloads are used in RESTful APIs.

This memo defines media types to be used for Entity Attestation Tokens (EATs).

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 https://www.rfc-editor.org/info/rfc9782.

Table of Contents

1. Introduction

Payloads used in Remote ATtestation procedureS (RATS) [RATS-ARCH] may require an associated media type for their conveyance, for example, when used in RESTful APIs (Figure 1).

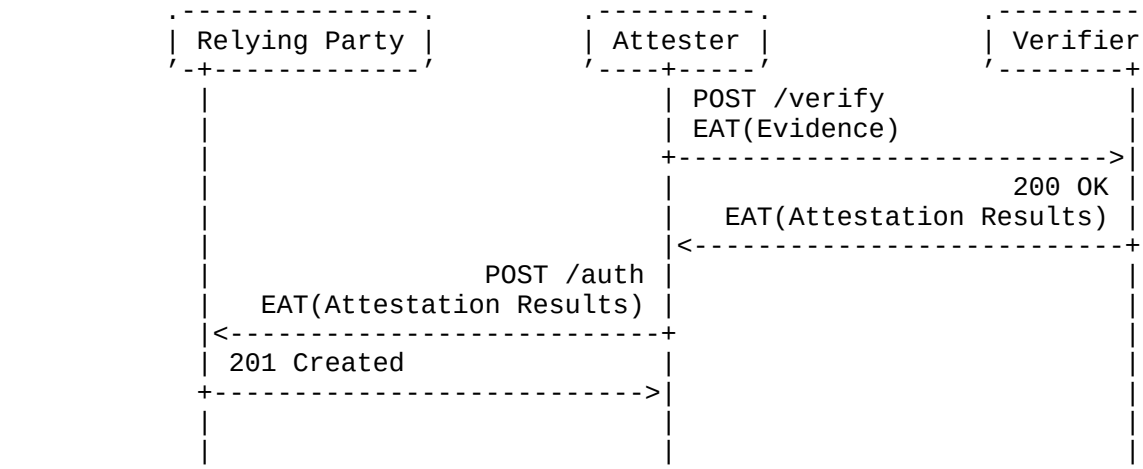


Figure 1: Conveying RATS Conceptual Messages in REST APIs Using EATs

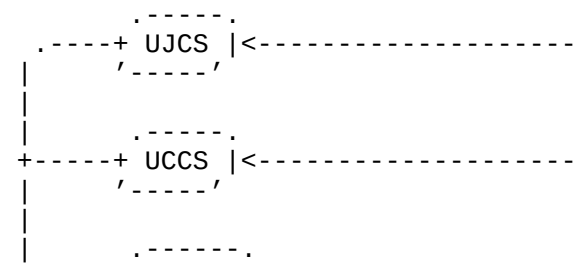
This memo defines media types to be used for EAT payloads [EAT] independently of the RATS Conceptual Message in which they manifest themselves. The objective is to give protocol, API, and application designers a number of readily available and reusable media types for integrating EAT-based messages in their flows, e.g., when using HTTP [BUILD-W-HTTP] or the Constrained Application Protocol (CoAP) [REST-IoT].

1.1. Terminology

This document uses the terms and concepts defined in [RATS-ARCH].

2. EAT Types

Figure 2 illustrates the six EAT wire formats and how they relate to each other. [EAT] defines four of them (CBOR Web Token (CWT), JSON Web Token (JWT), and the detached EAT bundle in its JSON and CBOR flavours), while [UCCS] defines the Unprotected CWT Claims Set (UCCS) and Unprotected JWT Claims Sets (UJCS).



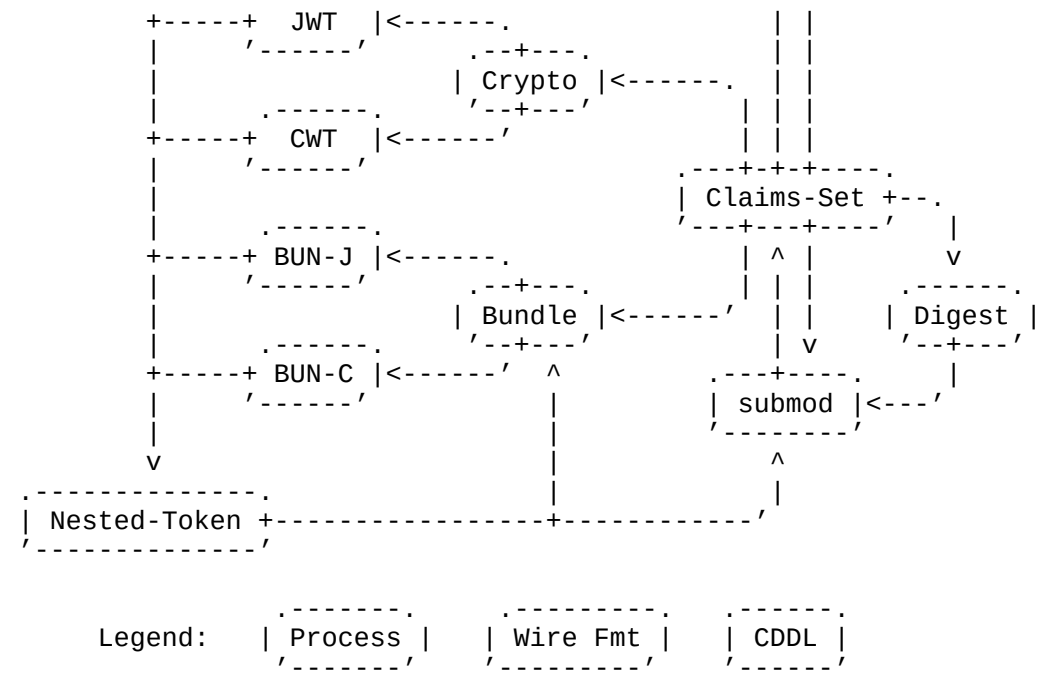


Figure 2: EAT Types

3. A Media Type Parameter for EAT Profiles

EAT is an open and flexible format. To improve interoperability, Section 6 of [EAT] defines the concept of EAT profiles. Profiles are used to constrain the parameters that producers and consumers of a specific EAT profile need to understand in order to interoperate, e.g., the number and type of claims, which serialisation format, the supported signature schemes, etc. EATs carry an in-band profile identifier using the "eat_profile" claim (see Section 4.3.2 of [EAT]). The value of the "eat_profile" claim is either an OID or a URI.

The media types defined in this document include an optional "eat_profile" parameter that can be used to mirror the "eat_profile" claim of the transported EAT. Exposing the EAT profile at the API layer allows API routers to dispatch payloads directly to the profile-specific processor without having to snoop into the request bodies. This design also provides a finer-grained and scalable type system that matches the inherent extensibility of EAT. The expectation being that a certain EAT profile automatically obtains a media type derived from the base (e.g., application/eat+cwt) by populating the "eat_profile" parameter with the corresponding OID or URL.

When the parameterised version of the EAT media type is used in HTTP (for example, with the "Content-Type" and "Accept" headers) and the value is an absolute URI (Section 4.3 of [URI]), the parameter-value (Appendix A of [HTTP]) uses the quoted-string encoding, for example:

```
application/eat+jwt; eat_profile="tag:evidence.example,2022"
```

Instead, when the EAT profile is an OID, the token encoding (i.e., without quotes) can be used. For example:

```
application/eat+cwt; eat_profile=2.999.1.
```

4. Examples

The example in Figure 3 illustrates the usage of EAT media types for transporting attestation evidence as well as negotiating the

acceptable format of the attestation result.

NOTE: '\' line wrapping per RFC 8792

```
POST /challenge-response/v1/session/1234567890 HTTP/1.1
Host: verifier.example
Accept: application/eat+cwt; eat_profile="tag:ar4si.example,2021"
Content-Type: application/eat+cwt; \
              eat_profile="tag:evidence.example,2022"

[ CBOR-encoded EAT w/ eat_profile="tag:evidence.example,2022" ]
```

Figure 3: Example REST Verification API (request)

The example in Figure 4 illustrates the usage of EAT media types for transporting attestation results.

NOTE: '\' line wrapping per RFC 8792

```
HTTP/1.1 200 OK
Content-Type: application/eat+cwt; \
              eat_profile="tag:ar4si.example,2021"

[ CBOR-encoded EAT w/ eat_profile="tag:ar4si.example,2021" ]
```

Figure 4: Example REST Verification API (response)

In both cases, a tag URI [TAG] identifying the profile is carried as an explicit parameter.

5. Security Considerations

Media types only provide clues to the processing application. The application must verify that the received data matches the expected format, regardless of the advertised media type, and stop further processing on failure. Failing to do so could expose the user to security risks, such as privilege escalation and cross-protocol attacks.

The security considerations of [EAT] and [UCCS] apply in full.

When using application/eat-ucs+json and application/eat-ucs+cbor in particular, the reader should review Section 3 of [UCCS], which contains a detailed discussion about the characteristics of a "Secure Channel" for conveyance of such messages.

6. IANA Considerations

6.1. +cwt Structured Syntax Suffix

IANA has registered +cwt in the "Structured Syntax Suffixes" registry [STRUCT-SYNTAX] in the manner described in [MEDIATYPES]. +cwt can be used to indicate that the media type is encoded as a CWT.

6.1.1. Registry Contents

Name: CBOR Web Token (CWT)

+suffix: +cwt

References: [CWT]

Encoding Considerations: binary

Interoperability Considerations: N/A

Fragment Identifier Considerations: The syntax and semantics of fragment identifiers specified for +cwt SHOULD be as specified for application/cwt. (At the time of publication, there is no fragment identification syntax defined for application/cwt.)

Security Considerations: See Section 8 of [CWT]

Contact: RATS WG mailing list (rats@ietf.org), or IETF Security Area (saag@ietf.org)

Author/Change Controller: Remote ATtestation ProcedureS (RATS) Working Group. The IETF has change control over this registration.

6.2. Media Types

IANA has registered the following media types in the "Media Types" registry [MEDIA-TYPES].

Name	Template	Reference
EAT CWT	application/eat+cwt	RFC 9782, Section 6.3
EAT JWT	application/eat+jwt	RFC 9782, Section 6.4
Detached EAT Bundle CBOR	application/eat-bun+cbor	RFC 9782, Section 6.5
Detached EAT Bundle JSON	application/eat-bun+json	RFC 9782, Section 6.6
EAT UCCS	application/eat-ucs+cbor	RFC 9782, Section 6.7
EAT UJCS	application/eat-ucs+json	RFC 9782, Section 6.8

Table 1: New Media Types

6.3. application/eat+cwt Registration

Type name: application

Subtype name: eat+cwt

Required parameters: N/A

Optional parameters: "eat_profile" (EAT profile in string format. OIDs must use the dotted-decimal notation. The parameter value is case insensitive.)

Encoding considerations: binary

Security considerations: Section 9 of [EAT]

Interoperability considerations: N/A

Published specification: RFC 9782

Applications that use this media type: Attesters, Verifiers, Endorsers and Reference-Value providers, and Relying Parties that need to transfer EAT payloads over HTTP(S), CoAP(S), and other transports.

Fragment identifier considerations: N/A

Person & email address to contact for further information: RATS WG mailing list (rats@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

6.4. application/eat+jwt Registration

Type name: application

Subtype name: eat+jwt

Required parameters: N/A

Optional parameters: "eat_profile" (EAT profile in string format. OIDs must use the dotted-decimal notation. The parameter value is case insensitive.)

Encoding considerations: 8bit

Security considerations: Section 9 of [EAT] and [BCP225]

Interoperability considerations: N/A

Published specification: RFC 9782

Applications that use this media type: Attesters, Verifiers, Endorsers and Reference-Value providers, and Relying Parties that need to transfer EAT payloads over HTTP(S), CoAP(S), and other transports.

Fragment identifier considerations: N/A

Person & email address to contact for further information: RATS WG mailing list (rats@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

6.5. application/eat-bun+cbor Registration

Type name: application

Subtype name: eat-bun+cbor

Required parameters: N/A

Optional parameters: "eat_profile" (EAT profile in string format. OIDs must use the dotted-decimal notation. The parameter value is case insensitive.)

Encoding considerations: binary

Security considerations: Section 9 of [EAT]

Interoperability considerations: N/A

Published specification: RFC 9782

Applications that use this media type: Attesters, Verifiers, Endorsers and Reference-Value providers, and Relying Parties that need to transfer EAT payloads over HTTP(S), CoAP(S), and other transports.

Fragment identifier considerations: N/A

Person & email address to contact for further information: RATS WG mailing list (rats@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

6.6. application/eat-bun+json Registration

Type name: application

Subtype name: eat-bun+json

Required parameters: N/A

Optional parameters: "eat_profile" (EAT profile in string format. OIDs must use the dotted-decimal notation. The parameter value is case insensitive.)

Encoding considerations: Same as [JSON]

Security considerations: Section 9 of [EAT]

Interoperability considerations: N/A

Published specification: RFC 9782

Applications that use this media type: Attesters, Verifiers, Endorsers and Reference-Value providers, and Relying Parties that need to transfer EAT payloads over HTTP(S), CoAP(S), and other transports.

Fragment identifier considerations: N/A

Person & email address to contact for further information: RATS WG mailing list (rats@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

6.7. application/eat-ucs+cbor Registration

Type name: application

Subtype name: eat-ucs+cbor

Required parameters: N/A

Optional parameters: "eat_profile" (EAT profile in string format. OIDs must use the dotted-decimal notation. The parameter value is case insensitive.)

Encoding considerations: binary

Security considerations: Sections 3 and 7 of [UCCS]

Interoperability considerations: N/A

Published specification: RFC 9782

Applications that use this media type: Attesters, Verifiers, Endorsers and Reference-Value providers, and Relying Parties that need to transfer EAT payloads over HTTP(S), CoAP(S), and other transports.

Fragment identifier considerations: N/A

Person & email address to contact for further information: RATS WG mailing list (rats@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

6.8. application/eat-ucs+json Registration

Type name: application

Subtype name: eat-ucs+json

Required parameters: N/A

Optional parameters: "eat_profile" (EAT profile in string format. OIDs must use the dotted-decimal notation. The parameter value is case insensitive.)

Encoding considerations: Same as [JSON]

Security considerations: Sections 3 and 7 of [UCCS]

Interoperability considerations: N/A

Published specification: RFC 9782

Applications that use this media type: Attesters, Verifiers, Endorsers and Reference-Value providers, and Relying Parties that need to transfer EAT payloads over HTTP(S), CoAP(S), and other transports.

Fragment identifier considerations: N/A

Person & email address to contact for further information: RATS WG mailing list (rats@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

6.9. CoAP Content-Format Registrations

IANA has registered the following Content-Format numbers in the "CoAP Content-Formats" registry, within the "Constrained RESTful Environments (CoRE) Parameters" registry group [CORE-PARAMS]:

Content Type	Content Coding	ID	Reference
application/eat+cwt	-	263	RFC 9782
application/eat+jwt	-	264	RFC 9782
application/eat-bun+cbor	-	265	RFC 9782
application/eat-bun+json	-	266	RFC 9782
application/eat-ucs+cbor	-	267	RFC 9781
application/eat-ucs+json	-	268	RFC 9782

Table 2: New Content-Formats

7. References

7.1. Normative References

[BCP225] Best Current Practice 225, <https://www.rfc-editor.org/info/bcp225>. At the time of writing, this BCP comprises the following:

Sheffer, Y., Hardt, D., and M. Jones, "JSON Web Token Best Current Practices", BCP 225, RFC 8725, DOI 10.17487/RFC8725, February 2020, <https://www.rfc-editor.org/info/rfc8725>.

[CORE-PARAMS] IANA, "CoAP Content-Formats", <https://www.iana.org/assignments/core-parameters>.

[CWT] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[EAT] Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", RFC 9711, DOI 10.17487/RFC9711, April 2025, <https://www.rfc-editor.org/info/rfc9711>.

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[JSON] 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/info/rfc8259>.

[MEDIA-TYPES]

IANA, "Media Types",
<https://www.iana.org/assignments/media-types>.

[MEDIATYPES]
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/info/rfc6838>.

[STRUCT-SYNTAX]
IANA, "Structured Syntax Suffixes", <https://www.iana.org/assignments/media-type-structured-suffix>.

[UCCS] Birkholz, H., O'Donoghue, J., Cam-Winget, N., and C. Bormann, "A Concise Binary Object Representation (CBOR) Tag for Unprotected CBOR Web Token Claims Sets (UCCS)", RFC 9781, DOI 10.17487/RFC9781, April 2025, <https://www.rfc-editor.org/info/rfc9781>.

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

7.2. Informative References

[BUILD-W-HTTP]
Best Current Practice 56, <https://www.rfc-editor.org/info/bcp56>.
At the time of writing, this BCP comprises the following:

Nottingham, M., "Building Protocols with HTTP", BCP 56, RFC 9205, DOI 10.17487/RFC9205, June 2022, <https://www.rfc-editor.org/info/rfc9205>.

[RATS-ARCH]
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/info/rfc9334>.

[REST-IoT] Kernen, A., Kovatsch, M., and K. Hartke, "Guidance on RESTful Design for Internet of Things Systems", Work in Progress, Internet-Draft, draft-irtf-t2trg-rest-iot-16, 23 April 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-rest-iot-16>.

[TAG] Kindberg, T. and S. Hawke, "The 'tag' URI Scheme", RFC 4151, DOI 10.17487/RFC4151, October 2005, <https://www.rfc-editor.org/info/rfc4151>.

Acknowledgments

Thank you Carl Wallace, Carsten Bormann, Dave Thaler, Deb Cooley, ric Vyncke, Francesca Palombini, Jouni Korhonen, Kathleen Moriarty, Michael Richardson, Murray Kucherawy, Orie Steele, Paul Howard, Roman Danyliw, and Tim Hollebeek for your comments and suggestions.

Authors' Addresses

Laurence Lundblade
Security Theory LLC
Email: lgl@securitytheory.com

Henk Birkholz
Fraunhofer Institute for Secure Information Technology
Rheinstrasse 75
64295 Darmstadt
Germany
Email: henk.birkholz@ietf.contact

Thomas Fossati
Linaro
Email: thomas.fossati@linaro.org