RATS Message Wrappers

draft-ftbs-rats-msg-wrap-01

RATS WG, IETF 115, London

What is it?

A uniform encapsulation format for RATS "conceptual" messages based on media types

Example Use Cases

- Stashing evidence, endorsements/ref-vals and attestation results in certificates and CRLs extensions [DICE]
- Embedding attestation results or evidence as first class authentication credentials in TLS handshake messages [TLS-A]
- Transporting attestation-related in RESTful APIs payloads [Veraison]
- Archival of attestation results as file system objects

Advantages

Converging on a common format:

- Allows multiple different protocols to tunnel attestation data in a homogeneous way
  - Easier consumption by RPs and Verifiers, as well as composition across different protocols (no need to encap-decap-encap).
- (by-product) interfaces / API to Attesting Environments can become more uniform

Design phases (A)

$$\zeta(s) = \frac{1}{\Gamma(s)} \int_0^\infty \frac{x^{s-1}}{e^x - 1} dx$$

Design phases (B)

```
[ type, value ]
```

Using Media Types as Type Discriminators

This allows us to build a variety of generic "RATS conceptual message" wrapping formats, including using CBOR tagging based on the RFC9277's `TN()` transform.

For example, a type-value wrapper build using a CDDL array:

```
rats-conceptual-message-wrapper = [ type, value ]
```

Type

"type" is either a CoAP C-F code-point or a media type string:

```
type = coap-content-format / media-type

coap-content-format = uint .size 2
media-type = text .abnf ("media-type" .det RFC6838)
```

Value

"value" is a CBOR byte string for the CBOR encoding (or a Base64 URL-safe string w/o padding for JSON serialisations):

```
value = cbor-bytes /   ; CBOR
        base64-string ; JSON

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

Example

Suppose you go ahead and register "application/vnd.intel.sgx" and then you also register the compressed CoAP C-F equivalent - let's say 30001.

IANA considerations

The first registration is an email to the IANA expert (Alexey or Murray); the second (since >10000 == FCFS) would be another email to IANA, this time bypassing expert review altogether.

Encoding

- As CBOR type-val array

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

- As JSON type-val array

```
[
    "application/vnd.intel.sgx",
    "q82rzQ"
]
```

Grab a CBOR tag automatically using RFC9277's TN()

Since `TN(30001)=1668576818`

→ CBOR tag

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

IANA considerations (cont.)

- FCFS allocation
- The bureaucracy is three emails in total: the first one with a possibly longer RTT due to human expert processing

Overhead considerations

The overhead of the two (CBOR) wrappers is essentially the same:

- CBOR tag:

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

- CBOR type-value array (one byte less):

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

Summary

- Simple (trivial) format
- Useful in a number of different scenarios
- Adopt?