Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- BCP 9 (Internet Standards Process)
- BCP 25 (Working Group processes)
- BCP 25 (Anti-Harassment Procedures)
- BCP 54 (Code of Conduct)
- BCP 78 (Copyright)
- BCP 79 (Patents, Participation)
IETF 114 Meeting Tips

In-person participants
- Make sure to sign into the session using the Meetecho (usually the “onsite tool” client) from the Datatracker agenda.
- Use Meetecho to join the mic queue
- *Keep audio and video off if not using the onsite version*

Remote participants
- Make sure your audio and video are off unless you are chairing or presenting during a session
- Use of a headset is strongly recommended
Resources for IETF 114 Philadelphia

- **Agenda**
  https://datatracker.ietf.org/meeting/agenda

- **Meetecho and other information:**
  https://www.ietf.org/how/meetings/114/preparation

- **If you need technical assistance:**
  http://www.ietf.org/how/meetings/issues/

- **RATS Materials:**
  https://datatracker.ietf.org/meeting/114/session/rats

- **RATS Minutes:**
  https://notes.ietf.org/notes-ietf-114-rats
Reminder of key points of the Code of Conduct [RFC 7154]

IETF participants:
1. Extend respect and courtesy to their colleagues at all times.
2. Have impersonal discussions.
3. Devise solutions for the global Internet that meet the needs of diverse technical and operational environments.
4. Are prepared to contribute to the ongoing work of the group.
10:00 : 10:05 **Agenda Bash & Logistics**  
(5 min) Nancy Cam-Winget, Kathleen Moriarty, Ned Smith

10:05 : 10:08 **Event Stream Subscription** (draft-ietf-rats-network-device-subscription)  
(3 min) Eric Voit

10:08 : 10:11 **AR4SI** (draft-ietf-rats-ar4si)  
(3 min) Eric Voit

10:11 : 10:14 **RATS Architecture** (draft-ietf-rats-architecture)  
(3 min) Dave Thaler, Michael Richardson

10:14 : 10:24 **EAT Update** (draft-ietf-rats-eat)  
(10 min) Laurence Lundblade

10:24 : 10:34 **EAT Media Types** (draft-lundblade-rats-eat-media-type) adoption call  
(10 min) Thomas Fossati, Laurence Lundblade, Henk Birkholz

10:34 : 10:44 **CoRIM** (draft-birkholz-rats-corim) adoption call  
(10 min) Thomas Fossati, Henk Birkholz, Yogesh Deshpande

10:44 : 10:59 **Concise TA Stores** (draft-wallace-rats-concise-ta-stores)  
(15 min) Carl Wallace

10:59 : 11:09 **EAT Collection Type** (draft-frost-rats-eat-collection)  
(10 min) Simon Frost

11:09 : 12:00 **Open Mic**  
(51 min)
Attestation Event Stream Subscription

draft-ietf-rats-network-device-subscription-01

IETF 114, July 2022, RATS WG

Henk Birkholz  Eric Voit  Wei Pan
Fraunhofer SIT  Cisco  Huawei
henk.birkholz@sit.fraunhofer.de  evoit@cisco.com  william.panwei@huawei.com
Relationship between drafts

Full WG

**draft-ietf-rats-architecture**
- Terminology
- Topological models
- Timing definitions

**draft-ietf-rats-reference-interaction-models**
- Terms
- Types of informational flows

**draft-ietf-rats-ar4si**
- Trustworthiness Claims
- Algorithm which enables Verifier to trust AR delivered via the Attester

Routers / Switches

**draft-ietf-rats-tpm-based-network-device-attest**
- Use case
- Operational prerequisites
- Evidence evaluation

**draft-ietf-rats-yang-tpm-charra**
- YANG definitions & RPCs
- TCG Algorithm registry

**draft-ietf-rats-network-device-subscription**
- Provably fresh events
- RFC-8639 based YANG subscriptions

**draft-ietf-rats-network-device-subscription**
- Specific objects and encodings for algorithm
- YANG model for provisioning

Operational prerequisites

YANG

Evidence via Telemetry

RFC Ed Queue (MISSREF)

@ AD Review

WG adopted progressing

WG adopted progressing

WG adopted progressing

WG adopted progressing

WG Consensus – awaiting writeup

simmering

Purpose & Scope

- Defines how to subscribe to a stream of attestation related Evidence on TPM-based network devices.
  - When subscribed, a Telemetry stream of verifiably fresh YANG notifications are pushed to the subscriber.
  - Notifications are generated for the Evidence going into TPM PCRs, and when the PCRs are extended.

- Result
  - Verifier is pushed new verifiably fresh Evidence whenever PCRs change.
Status

• Stable as a direct combination of RFC-8639 & Charra
• Ready to progress now that Charra is in RFC editor’s queue
• Needs Security Considerations section text
• Then request WGLC
Contents

• **Part 1:** Information Element definitions for Attestation Results (AR) generated by Verifier to support Secure Interactions between Attester and Relying Party

• **Part 2:** End-to-end implementation options: (a) Background check, (b) AR Augmented Evidence

• Implementations:
  • **Trusted Path Routing** (Proprietary – Cisco)
  • **Veraison** (Open Source, aspiration = Confidential Compute Consortium adoption)
Changes since IETF113

- Awaiting CCC definitions of various Confidential Computing environments
- Mailing list discussion on EAT ‘(endorsed-)security-level’
  - Agree new hardware environments could be added to ar4si:
- Future EAT integration (driven by “Same claim in Evidence and Results” & “EAT Profiles” threads)
  - Awaiting clarity on how to transmit the context-based meaning of claims within AR based on structured Profiles. (I.e., need to articulate the interdependence of AR asserted claims based on the namespace/profile in which they are received.)
- When clear, will add a new ar4si section showing EAT encodings:

$\text{Claims-Set-Claims} \equiv (\text{trustworthiness-claim-label} \Rightarrow \text{trustworthiness-claim-type})$

\text{trustworthiness-claim-type} = \{+ \text{trustworthiness-claim-format}\}

...
## Section 2.3.1: AR Design Principles for Trustworthiness Claims

<table>
<thead>
<tr>
<th>Design Principle</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Expose a small number of Trustworthiness Claims</td>
<td>A plethora of similar Trustworthiness Claims will result in divergent choices made on which to support between different Verifiers. This would place a lot of complexity in the Relying Party as it would be up to the Relying Party (and its policy language) to enable normalization across rich but incompatible Verifier object definitions.</td>
</tr>
<tr>
<td>(2) Each Trustworthiness Claim enumerates only the specific states that could viably result in a different outcome after the Policy for Attestation Results has been applied</td>
<td>By explicitly disallowing the standardization of enumerated states which cannot easily be connected to a use case, we avoid forcing implementers from making incompatible guesses on what these states might mean.</td>
</tr>
<tr>
<td>(3) Verifier and RP developers need explicit definitions of each state</td>
<td>Without such guidance, the Verifier will append plenty of raw supporting info. This relieves the Verifier of making the hard decisions. Of course, this raw info will be mostly non-interpretable and therefore non-actionable by the Relying Party.</td>
</tr>
<tr>
<td>(4) Support standards and non-standard extensibility</td>
<td>Standard types of Verifier generated Trustworthiness Claims should be vetted by the full RATS working group, rather than being maintained in a repository which doesn’t follow the RFC process. This will keep a tight lid on extensions which must be considered by the Relying Party’s policy language. Because this process takes time, non-standard extensions will be needed for implementation speed and flexibility</td>
</tr>
</tbody>
</table>
RATS Architecture Design Team
Status and Walkthrough

WHO:
- Henk Birholz(*)
- Thomas Fossati
- Yogesh Deshpande
- Andrew Guinn
- Thomas Hardjono
- Sarah C. Helble
- Xinxin Fan IoTeX
- Eliot Lear
- Peter Loscocco
- Laurence Lundblade
- Nicolae Paladi
- Wei (William) Pan(*)
- Michael Richardson(*)
- Paul Rowe
- Ned Smith(*)
- Dave Thaler(*)
- Akura Tsukamoto
- Eric Voit
- Monty Wiseman
- Ling (Frank) Xia
- Penglin Yang

WHEN: Tuesdays 10am EST.

four meetings since IETF113

ISSUES: 9 total since IETF113

a few issues open

Pull requests:

12 pull requests

processed AD review comments

now on round two of comments

(*)-listed author
Minor changes since IETF113

- AD review comments
- Beautiful SVG
- Adjustments to terminology
- Integrity Protection -> Conceptual Message Protection
The Attester does not consume the Attestation Result, but might cache it.

The Attester can then present the Attestation Result (and possibly additional Claims) to a Relying Party, which then compares this information against its own appraisal policy.

The Attester may also present the same Attestation Result to other Relying Parties.

Attestation Results and Evidence are signed, and can not be modified!
In this model, an Attester conveys Evidence to a Relying Party, which treats it as opaque and simply forwards it on to a Verifier.

The Verifier compares the Evidence against its appraisal policy, and returns an Attestation Result to the Relying Party.

The Relying Party then compares the Attestation Result against its own appraisal policy.

The resource access protocol between the Attester and Relying Party includes Evidence rather than an Attestation Result, but that Evidence is not processed by the Relying Party.

Attestation Results and Evidence are signed, and can not modified!
Questions
Discussion
EAT Update

Laurence Lundblade

IETF 114 July 2022
Important changes in the -13 and -14 drafts (since IETF 113)

- **Document Organization**
  - Claims section divided into 4: nonce, claims about entity, claims about token, cryptographic keys
  - Moved several sections to appendices; core of document is shorter now

- **Specification changes**
  - Use CoAP for content type of manifest and swevidence claims rather than CBOR tags
  - Added SPDX and CycloneDX manifest types
  - Measurement results claim reworked
  - Added a standard EAT CBOR profile for constrained devices.

- **CDDL Improvements**
  - Claims-Set now replicated in the document
  - No definition of UCCS, CDDL socket for where UCCS plugs in
  - Lots of improvements CDDL; validating for JSON and CBOR examples
  - Lots and lots of wording improvements

- **Profiles section**
  - Resynch with RATS Architecture terminology
  - For the following claims: UEID, SUEID, DLOAs Boot Count, cti & jti, nonce, SW Name,
  - Relation of Evidence to Attestation Results
Work in the EAT queue

- Security Considerations – some improvements needed
- Introduction and Abstract – some comments to address
- Various other comments to address – mostly clarifications and small inconsistencies

- Possible minor improvements:
  - Optional nonce? – accommodate timestamp-based freshness in RATS architecture?
  - Add standard profile for JSON? – we have one for CBOR now
EAT Media Types
Carrying EAT around

https://datatracker.ietf.org/doc/draft-lundblade-rats-eat-media-type/

RATS WG, IETF 114
# Base Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAT CWT</td>
<td>application/eat-cwt</td>
</tr>
<tr>
<td>EAT JWT</td>
<td>application/eat-jwt</td>
</tr>
<tr>
<td>EAT CBOR DEB</td>
<td>application/eat-deb+cbor</td>
</tr>
<tr>
<td>EAT JSON DEB</td>
<td>application/eat-deb+json</td>
</tr>
<tr>
<td>EAT UCS</td>
<td>application/eat-ucs+cbor</td>
</tr>
<tr>
<td>EAT UJCS</td>
<td>application/eat-ucs+json</td>
</tr>
</tbody>
</table>
The "profile" parameter

→ An optional profile parameter mirrors the `eat_profile` claim value of the transported EAT

→ dotted-decimal absolute OID or URI

→ To help API routing middleware, application servers and the like

→ A fine-grained and scalable type system that matches the inherent extensibility of EAT
Example (Successful negotiation)

POST /challenge-response/v1/session/02a4fe12-1887-4a24-9d61-4b76b0f3c488 HTTP/1.1
Host: verifier.example
Content-Type: application/eat-cwt; profile=tag:evidence.example,2022
Accept: application/eat-jwt; profile=tag:ar4si.example,2021

[ CWT-protected EAT w/ profile=tag:evidence.example,2022 ]

HTTP/1.1 200 OK
Content-Type: application/eat-jwt; profile=tag:ar4si.example,2021

[ JWT-protected EAT w/ profile=tag:ar4si.example,2021 ]
Example (Failed negotiation 1)

POST /verify/my-evidence HTTP/1.1
Host: verifier.example
Content-Type: application/eat-cwt; profile=tag:evidence.example,3022
Accept: application/eat-jwt; profile=tag:ar4si.example,2021

[ CWT-protected EAT w/ profile=tag:evidence.example,3022 ]

HTTP/1.1 415 Unsupported Media Type
Accept: application/eat-cwt; profile=tag:evidence.example,2022,
application/eat-cwt; profile=tag:evidence.example,2021
Example (Failed negotiation II)

POST /verify/my-evidence HTTP/1.1
Host: verifier.example
Content-Type: application/eat-cwt; profile=tag:evidence.example,2022
Accept: application/eat-jwt; profile=tag:ar4si.example,3021

[ CWT-protected EAT w/ profile=tag:evidence.example,2022 ]

HTTP/1.1 406 Not Acceptable
Accept: application/eat-jwt; profile=tag:ar4si.example,2021
Adopt?
CoRIM
Explaining Attesters to Verifiers

https://datatracker.ietf.org/doc/draft-birkholz-rats-corim/

RATS WG, IETF 114
History

- WG indicated strong support for adopting CoRIM as a starting point for conveyance of reference values and endorsements
  - see ML thread
- Unfortunately CoRIM was out-of-scope of charter-01
History

Recharter happened (charter-02)

- Main deliverable 6: “Standardize interoperable data formats to securely declare and convey endorsements and reference values.”
- Milestone: “Jul 2022 Call for adoption on Concise Reference Integrity and Endorsement Manifests”
In the meantime ...

- version -03 published
- (open source) tooling available
- CoRIM protocol extensions
- application profiles built on top of CoRIM
Adopt?
Background

If you want to catch up with the nitty gritty:

→ Previously on this YT channel
→ Slides
concise-ta-stores (CoTS)

IETF 114 – Philadelphia – July 2022 – RATS Working Group
Russ Housley
Carl Wallace
Draft and example implementation

  • https://github.com/carl-wallace/draft-wallace-rats-concise-ta-stores

• https://github.com/carl-wallace/corim
  • Fork of https://github.com/veraison/corim
Desired path forward

• Accept as working group draft and proceed on standards track
Why define concise-ta-stores?

• Current RATS work implies use of trust anchors for many different purposes, including verification of evidence, endorsements, reference values, digital letters of approval, public key certificates, revocation information, etc.

• The concise-ta-stores spec provides a means of representing trust anchors with limitations on the contexts in which a trust anchor store may be used

• Support various combinations of TAs and CAs, i.e., single vendor TA/CA, multiple vendor TA/single vendor CA, multiple vendor TA/multi-vendor CA
Why define as an extension of CoRIM?

• Similar general purpose of conveying information to verifiers and relying parties

• Why not define as a profile of CoRIM?
  • The lifecycle of TAs and CAs is different than the lifecycle of reference data
  • The use cases for trust anchors in RATS are broader than CoRIM
  • The verification-map in CoRIM is tied to CoMIDs, leaving no easy path to support non-CoMID-centric use cases
Basic structure

```javascript
concise-ta-store-map = {
    ? tastore.language => language-type
    ? tastore.store-identity => tag-identity-map
    tastore.environments => environment-group-list
    ? tastore.purposes => [+ $tas-list-purpose]
    ? tastore.perm_claims => [+ $$claims-set-claims]
    ? tastore.excl_claims => [+ $$claims-set-claims]
    tastore.keys => cas-and-tas-map
}
```

- concise-ta-stores are arrays of the concise-ta-store-map, which defines a trust anchor (TA) store
- Each TA store may be defined with optional constraints
- Optional store-identity facilitates linking from other artifacts
- Each TA store contains at least one TA, which may also optionally constrained
Basic structure: store identity

tag-identity-map = {
    &(tag-id: 0) => $tag-id-type-choice
    ? &(tag-version: 1) => tag-version-type
}

• Defined in CoRIM.
• Allows for identifying a store using a UUID or textual identifier with an optional version value
Basic structure: environments

environment-group-list-map = {
    ? tastore.environment_map => environment-map,
    ? tastore.concise_swid_tag => abbreviated-swid-tag,
    ? tastore.named_ta_store => named-ta-store,
}

• environment-map is from CoRIM. Features class, instance, and group.
• abbreviated-swid-tag is modified from CoSWID to allow all fields except entity to be optional.
• named-ta-store is freeform text name
Basic structure: constraints

```
$tas-list-purpose /= "cots"
$tas-list-purpose /= "corim"
$tas-list-purpose /= "coswid"
$tas-list-purpose /= "eat"
$tas-list-purpose /= "key-attestation"
$tas-list-purpose /= "certificate"
$tas-list-purpose /= "dloa"
```

• The purpose field is similar to the PKIX extended key usage extension
• Represents constraints as abstract names, i.e., corim, eat, dloa, etc.
  • Corresponding EKU values will be defined for use in certificates
  • Will propose a registry for purpose values
Basic structure: constraints (cont.)

? tastore.perm_claims => [+ $$claims-set-claims]
? tastore.excl_claims => [+ $$claims-set-claims]

• The perm_claims and excl_claims fields can carry EAT claims to represent acceptable or unacceptable values for associated TA(s)
• $$claims-set-claims is a group socket defined in EAT
  • Claims are registered in the CBOR Web Token (CWT) Claims registry: [http://www.iana.org/assignments/cwt](http://www.iana.org/assignments/cwt)
Basic structure: keys

```plaintext
trust-anchor = {
  format => $pkix-ta-type
  data => bstr
}
cas-and-tas-map = {
  tastore.tas => [ + trust-anchor ]
  ? tastore.cas => [ + pkix-cert-data ]
}
```

• Provides means to convey trust anchors and, optionally, intermediate CA certificates
• TAs can be represented as bare public key (i.e., SubjectPublicKeyInfo), a Certificate, or a TrustAnchorInfo
  • TrustAnchorInfo allows for per-trust anchor constraints, which would be in addition to any TA store constraints
Security mechanisms

• Inherits signed structure from CoRIM, which uses from COSE
• Recommend verification to a trust anchor with the CoTS purpose
Things left to other specifications

• Use of constraints represented in a TA store definition or TA definition is not covered in this specification
Questions

1. Is CoRIM extension the right way forward?
2. Should environment be simplified to focus on some identity characteristics shared by CoMID/CoSWID?
3. Do constraints mechanisms adequately cover the TA landscape implied by RATS architecture?
4. Other questions...
Entity Attestation Token (EAT) Collection Type

draft-frost-rats-eat-collection-01
IETF 114, July 2022, RATS WG

Simon Frost
Arm
simon.frost@arm.com
EAT Collections Draft Introduction

• Purpose: an extension for EAT top level object for use cases where there may be no top level signer

• Current EAT needs top level object to be CWT / JWT / DEB
  • exception is UCCS extension, but that is for tightly defined use case

• The Collection extension is used where the overall token is made up of multiple independent tokens with an internally defined integrity relationship

• In particular where the number of tokens present may change
  • or the definition of the leaf signer may vary by deployment

• e.g. migrating info currently defined as x.509 certificate chain to EAT

• e.g. platform / workload token parts from Arm CCA
### CCA ‘Direct Sign’ model

<table>
<thead>
<tr>
<th>Platform (EAT)</th>
<th>Workload (EAT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...claims...</td>
<td>...claims...</td>
</tr>
<tr>
<td>Challenge: H(workload claims)</td>
<td>Challenge: RP nonce</td>
</tr>
</tbody>
</table>

Platform Key

### CCA ‘Delegated Sign’ model

<table>
<thead>
<tr>
<th>Platform (EAT)</th>
<th>Workload (EAT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...claims...</td>
<td>...claims...</td>
</tr>
<tr>
<td>Challenge: H(RAK_{pub})</td>
<td>Challenge: RP nonce</td>
</tr>
</tbody>
</table>

Platform Key

RAK

Future potential to add additional token(s)

---

Could format into single EAT DEB...

Could format into top level workload with Platform as submod...

TBD...

Significant format & (hence code) change for subtle differences by deployment...

...instead, treat as Collection entries...
EAT Collections Format

• Tagged Map containing:
  • Collection Identifier (optional) -> EAT profile claim
  • One or more entries consisting of CWT / JWT / DEB
    • Tags on map entries may be meaningful to the profile
  • Each entry must have its own integrity and an integrity relationship to other entries
    • Custom (profile) defined
    • 1:1 or 1:n
EAT Collections Draft Status

• draft-00 released to WG
• draft-01 released addressing review comments from -00
  • CDDL embracing multiple formats for member tokens
  • CDDL allowing 1+ members rather than 2+
  • Greater emphasis on security considerations
Thank You!