

RATS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 27 November 2026

H. Birkholz  
Fraunhofer SIT  
E. Voit  
Cisco  
W. Pan  
Huawei  
26 May 2026

Attestation Event Stream Subscription  
draft-ietf-rats-network-device-subscription-12

## Abstract

This document defines how to subscribe to YANG Event Streams for Remote Attestation Procedures (RATS). Specifically, this document defines a YANG module that augments the YANG module for Trusted Platform Module (TPM)-based Challenge-Response Remote Attestation (CHARRA), enabling subscription to RATS Conceptual Messages of the Evidence type and auxiliary Event Logs as part of that Evidence. The module defined requires at least one Trusted Platform Module (TPM) 1.2 or TPM 2.0 (or equivalent hardware implementation providing the same protected capabilities as a TPM) must be available on the Attester on which the YANG server is running.

## 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 27 November 2026.

## Copyright Notice

Copyright (c) 2026 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 . . . . .  | 3  |
| 2. Terminology . . . . .   | 4  |
| 2.1. Requirements Notation . . . . .                                       | 5  |
| 3. Operational Model . . . . .   | 5  |
| 3.1. Sequence Diagrams . . . . .   | 5  |
| 3.1.1. Terminology Mapping . . . . .                                       | 5  |
| 3.1.2. Time Considerations Mapping . . . . .                               | 7  |
| 3.2. Continuously Verifying Freshness . . . . .                            | 10 |
| 3.2.1. TPM 1.2 Quote . . . . .   | 10 |
| 3.2.2. TPM 2 Quote . . . . .   | 10 |
| 4. Remote Attestation Event Stream . . . . .                               | 11 |
| 4.1. Subscription to the <attestation> Event Stream . . . . .              | 11 |
| 4.2. Replaying a History of Previous TPM Extend Operations . . . . .       | 12 |
| 4.2.1. TPM2 Heartbeat . . . . .  | 13 |
| 4.3. YANG Notifications Placed on the <attestation> Event Stream . . . . . | 13 |
| 4.3.1. pcr-extend . . . . .  | 13 |
| 4.3.2. tpm12-attestation . . . . .   | 15 |
| 4.3.3. tpm20-attestation . . . . .   | 15 |
| 4.4. Filtering Evidence at the Attester . . . . .                          | 16 |
| 4.5. Replaying Previous PCR Extend Events . . . . .                        | 16 |
| 4.6. Configuring the <attestation> Event Stream . . . . .                  | 17 |
| 5. YANG Module . . . . .   | 17 |
| 6. Event Streams for Conceptual Messages . . . . .                         | 24 |
| 7. Privacy Considerations . . . . .  | 25 |
| 8. Security Considerations . . . . .                                       | 25 |
| 8.1. Remote ATttestation procedureS (RATS) . . . . .                       | 25 |
| 8.2. Yet Another Next Generation (YANG) . . . . .                          | 25 |
| 8.3. Other . . . . .   | 25 |
| 9. IANA Considerations . . . . .   | 25 |
| 10. References . . . . .   | 26 |
| 10.1. Normative References . . . . .                                       | 26 |
| 10.2. Informative References . . . . .                                     | 27 |
| Acknowledgements . . . . .   | 28 |
| Authors' Addresses . . . . .   | 28 |

## 1. Introduction

[RFC9683] and [RFC9684] define the operational prerequisites and a YANG Model for acquiring Evidence from a network device containing at least one TPM 1.2 or TPM 2.0 (or equivalent hardware implementations providing the same protected capabilities [TCG-Glossary] as a TPM). However, these documents are based on the challenge-response interaction model (CHARRA in Section 7.1 of [I-D.ietf-rats-reference-interaction-models]), which has limitations. One such limitation is that it is the responsibility of a Verifier to request signed Evidence from a separate Attester containing a TPM. This means that the interval between a security-relevant change event occurring and the event becoming visible to the interested RATS entities, such as a Verifiers or a Relying Parties, can be unacceptably long. It is common to convey Conceptual Messages ad-hoc or periodically via requests. As new technologies emerge, some of these solutions require Conceptual Messages to be conveyed from one RATS entity to another without the need for continuous polling. Subscription to YANG Notifications [RFC8639] provides a set of standardized tools to facilitate these emerging requirements. This memo specifies a YANG augmentation for subscribing to YANG-modelled remote attestation Evidence, as defined in [RFC9684].

Essentially, the limitation of poll-based interactions has two adverse effects:

1. Conceptual Messages are not streamed to interested consumers of information (e.g., Verifiers or Relying Parties) as soon as they are generated.
2. Even if they were streamed, the freshness of Conceptual Messages cannot be appraised in every scenario. This is particularly important for Conceptual Messages, such as Evidence, that depend heavily on freshness.

This specification addresses the first adverse effect by enabling consumers of Conceptual Messages (subscribers) to request a continuous stream of new or updated Conceptual Messages via an [RFC8639] subscription to an <attestation> Event Stream. This new Event Stream is defined in this document and is provided by the producer of Conceptual Messages (the publisher). As covered by this document, via a Verifier's subscription to an Attester's Evidence, the Attester will continuously stream a requested set of freshly generated Evidence to the subscribing Verifier. For example, when a network device's Evidence changes following events such as booting, updating, control unit failover, plugging in or out of forwarding units, an attack, or certificate lifetime change, the network device will generate fresh Evidence available to the subscribing Verifier.

The second adverse effect stems from the use of nonces in the challenge-response interaction model Section 7.1 of [I-D.ietf-rats-reference-interaction-models] realized in [RFC9684]. According to [RFC9684], an Attester must wait for a new nonce from a Verifier before generating a new TPM Quote. To address delays resulting from this wait, this specification allows freshness to be asserted asynchronously via the streaming attestation interaction model [I-D.ietf-rats-reference-interaction-models]. To convey a RATS Conceptual Message, an initial nonce is provided when subscribing to an Event Stream.

There are several options to populate or refresh the nonce value provided by the initial subscription. All of these methods are out-of-band of an established subscription to YANG Notifications. Two alternative methods are taken into account by this document:

1. A central provider supplies new, fresh nonces (e.g., via a Handle Provider that distributes Epoch IDs to all entities in a domain as described in [RFC9334] and as facilitated by the Uni-Directional Remote Attestation described in Section 7.2 of [I-D.ietf-rats-reference-interaction-models]), or
2. A nonce can be updated by -- potentially periodically or ad-hoc -- sending out-of-band TPM Quote requests as facilitated by [RFC9684].

Both approaches assume that clock drift can occur between the entities involved. Consequently, other conditions arising in different application scenarios ought to be considered in the same way. For example, the time of Claims collection ought to be taken into account as it potentially impacts the freshness of Evidence.

The scope of this document is limited to the removal of the two adverse effects described when using the specified YANG augmentation. In essence, the YANG augmentation enables RATS Verifiers to maintain a continuous appraisal procedure of verifiably fresh Attester Evidence without relying on continuous polling.

## 2. Terminology

The following terms are imported from [RFC9334]: Attester, Conceptual Message, Evidence, Relying Party, and Verifier. Also imported are the time definitions time(VG), time(NS), time(EG), time(RG), and time(RA) from that document's Appendix A. The following terms are imported from [RFC8639]: Event Stream, Subscription, Publisher, Event Stream Filter, Dynamic Subscription.

## 2.1. Requirements Notation

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.

## 3. Operational Model

[RFC9683] describes the conveyance of TPM-based Evidence from a Verifier to an Attester using the CHARRA interaction model Section 7.1 of [I-D.ietf-rats-reference-interaction-models]. The operational model and corresponding sequence diagram described in this section is based on [RFC9684]. The basis for interoperability required for additional types of Event Streams is covered in Section 6. The following sub-section focuses on subscription to YANG Notifications to the <attestation> Event Stream.

### 3.1. Sequence Diagrams

This section illustrates the subscription interaction model by mapping terms from [I-D.ietf-rats-reference-interaction-models] and illustrating timing consideration based on Figure 3 of [RFC9683]. Both sequence diagrams Figure 1 and Figure 2 highlight TPM-specific aspects and the Dynamic Subscription (as specified in [RFC8639]) to an <attestation> Event Stream. The contents of the <attestation> Event Stream are defined below within Section 4.

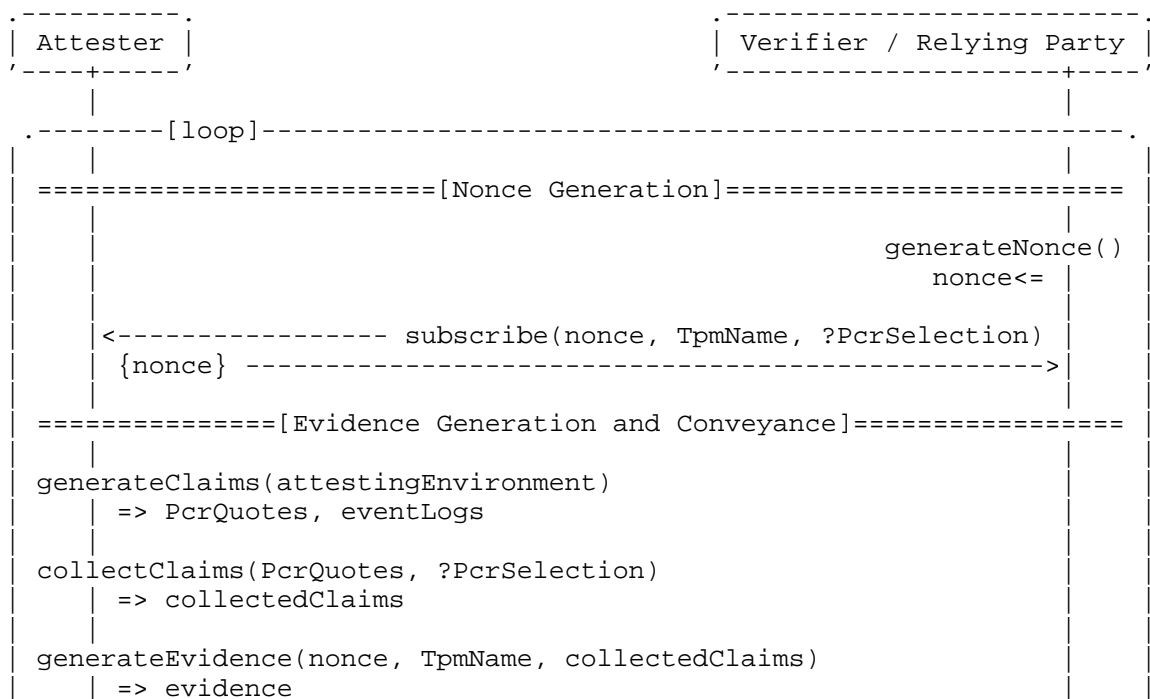
#### 3.1.1. Terminology Mapping

The terms defined in [RFC9684] are mapped to the model described in Section 7.3.1 of [I-D.ietf-rats-reference-interaction-models] (Streaming Remote Attestation without a Broker) to produce the sequence diagram Figure 1.

The terminology mapping is as follows:

- \* handle is substituted with nonce, a nonce generated by the Verifier. This is a nonce-value byte string, obtained from either the tpm12-challenge-response-attestation RPC or the tpm20-challenge-response-attestation RPC, as specified in [RFC9684]. Hence, in this case, a 'nonce' value is populated out of band and can be used more than once within the scope of a subscription, based on local policies that enforce freshness requirements (see definition of handle in Section 6 of [I-D.ietf-rats-reference-interaction-models]).

- \* attEnvIDs is substituted with TpmName, a TPM "name" text string selected from the tpms Container, as specified in [RFC9684]
- \* claimsSelection is substituted with PcrSelection, an optional "pcr-index" from either the tpm12-challenge-response-attestation RPC or the tpm20-challenge-response-attestation RPC as specified in [RFC9684]. If no PCR is selected, all PCR banks are returned.
- \* claims is substituted with PcrQuotes, which is the "output" of either the tpm12-challenge-response-attestation RPC or the tpm20-challenge-response-attestation RPC, as specified in [RFC9684]. Unlike event logs, there is no delta to a previous iteration of PCR Quotes during a subscription; all new (selected) Quotes are conveyed as fresh Evidence.
- \* eventLogs represents "system-event-logs" that are in the "output" of the log-retrieval RPC, as defined in [RFC9684].
- \* eventLogsDelta represents "system-event-logs" as specified in the "output" of the log-retrieval RPC as defined in [RFC9684] where the "output" is limited as if the "input" were parameterized via an index type (last-entry, index, timestamp) set to the last event in the previously conveyed eventLogs.



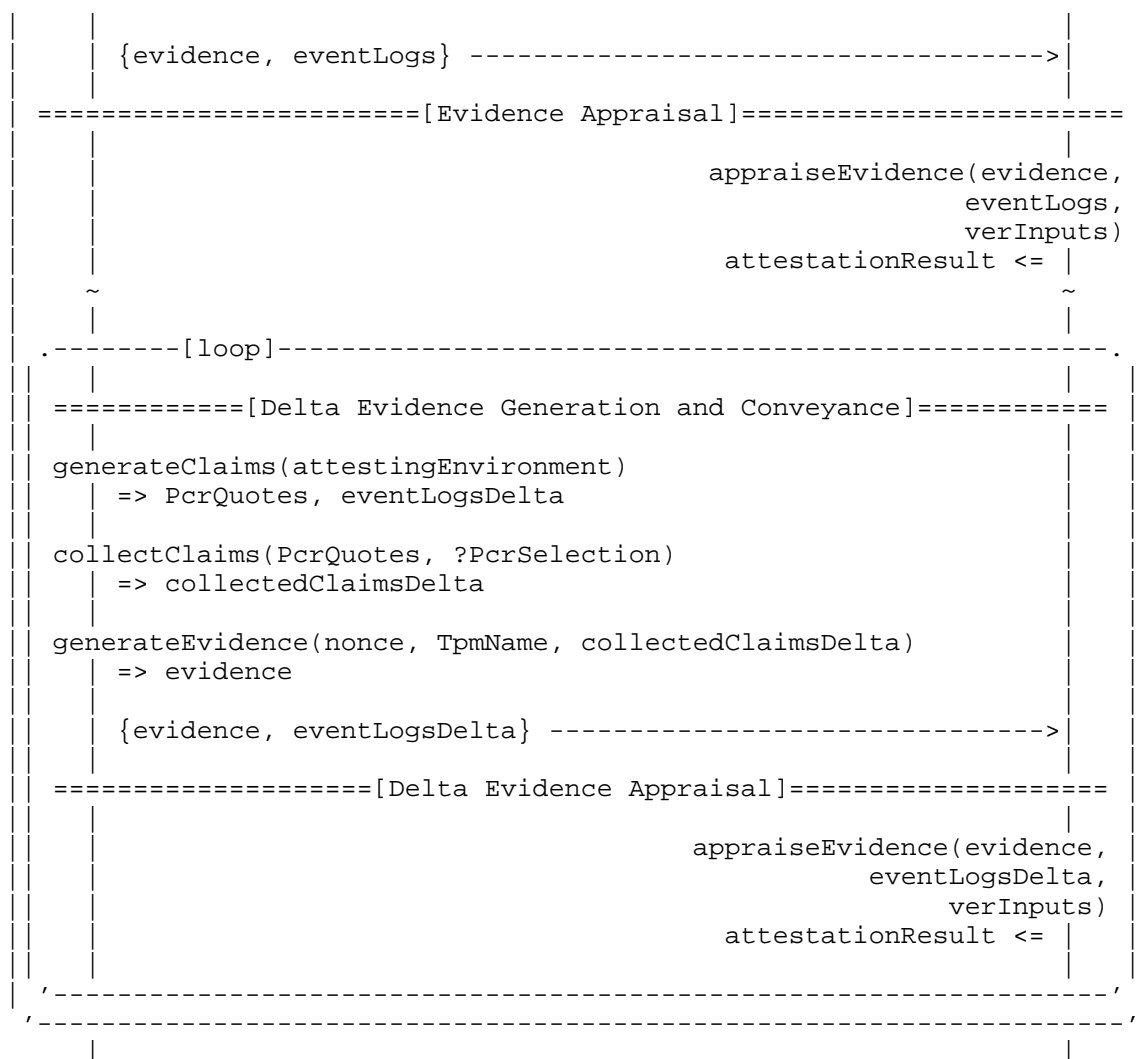


Figure 1: YANG Subscription Model for Remote Attestation

### 3.1.2. Time Considerations Mapping

[RFC9334] defines "Relevant Events over Time" in RATS which also provides the input for Figure 3 of [RFC9683]. The following sequence diagram focusses on matching the defined events with the interactions between the Attester and the Verifying Relying Party. The action of conveying "collectClaims", which is defined in Section 6 of [I-D.ietf-rats-reference-interaction-models], is not defined by [RFC9334]. As a result, that action cannot be matched to a specified

event time.

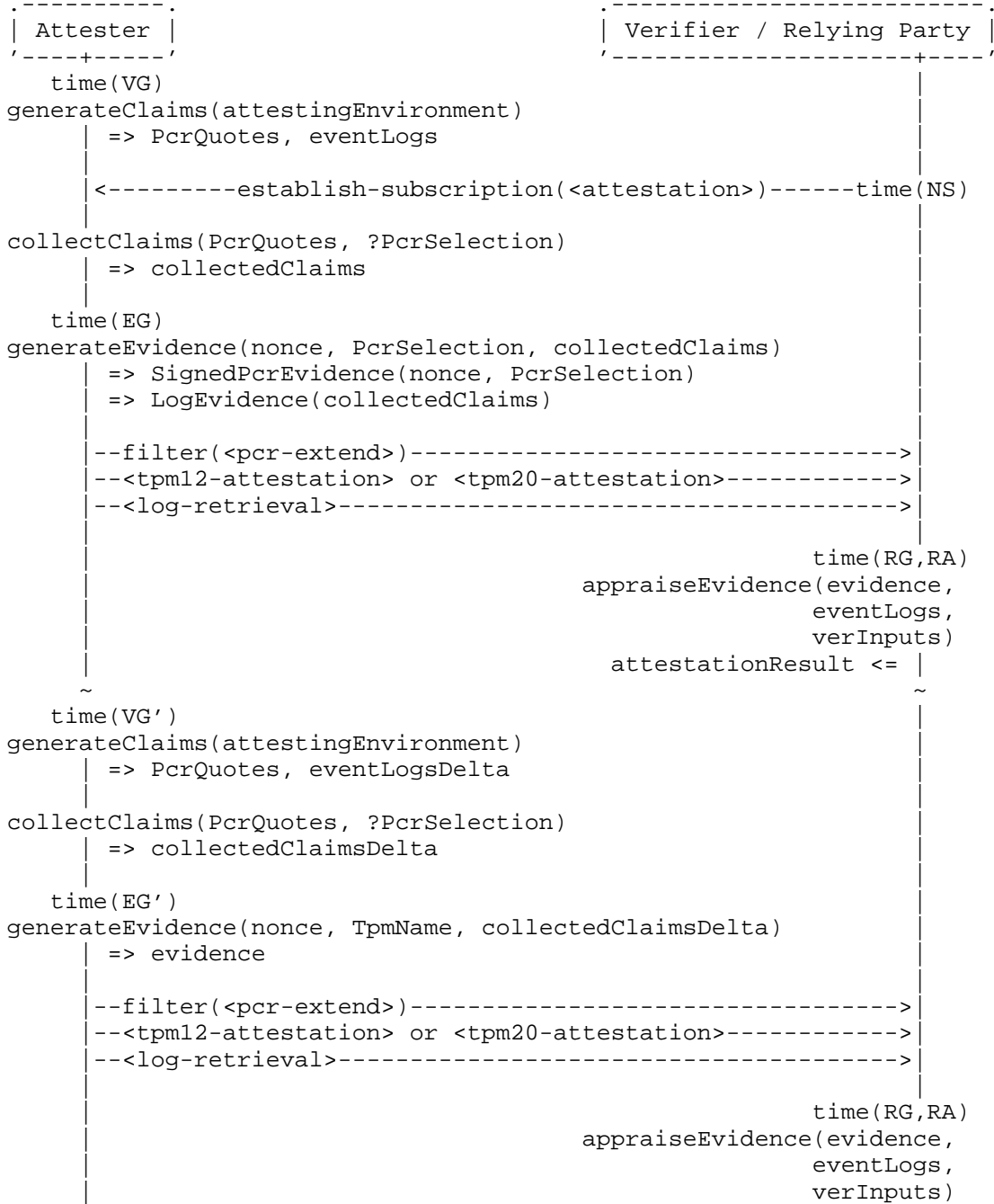




Figure 2: YANG Subscription Model for Remote Attestation

- \* time(VG, RG, RA) are identical to the corresponding time definitions from [RFC9683].
- \* time(VG', RG', RA') are subsequent instances of the corresponding times from Figure 3 in [RFC9683].
- \* time(NS) the subscriber generates a nonce and makes an [RFC8639] <establish-subscription> request based on that nonce value. This request also includes the augmentations defined in this document's YANG model. Key subscription RPC parameters include:
  - the nonce,
  - a set of PCRs of interest which the Verifier wants to be appraised, and
  - an optional filter that can reduce the logged events on the <attestation> stream pushed to the Verifier.
- \* time(EG) an initial response of Evidence is returned to the Verifier. This includes:
  - a replay of filtered log entries, which have extended into a PCR of interest since boot, are sent in the <pcr-extend> notification, and
  - a signed TPM quote that contains at least the PCRs from the <establish-subscription> RPC are included in a <tpm12-attestation> or <tpm20-attestation>). This quote must have been generated based on the nonce value provided at time(NS).
- \* time(VG', EG') this occurs when a PCR is extended subsequent to time(EG). Immediately after the extension, the following information needs to be pushed to the Verifier:
  - any values extended into a PCR of interest,
  - a signed TPM Quote showing the result the PCR extension, and
  - a nonce value (see 'handle' above or Section 6 of [I-D.ietf-rats-reference-interaction-models]), which is either the initially received nonce or a more recently received nonce

value, for example, a nonce value extracted or derived from an Epoch ID (see Section 10.3 of [RFC9334]) that contains a new nonce value or equivalent qualified data used as a nonce value.

One way to acquire a new time synchronisation that allows for the reuse of the initially received nonce as a fresh handle is elaborated on in Section 3.2 below.

### 3.2. Continuously Verifying Freshness

As there is no new Verifier nonce provided at time(EG'), it is important to validate the freshness of TPM Quotes which are delivered at that time. Methods of doing this verification vary based on the capabilities of the TPM cryptoprocessor used.

#### 3.2.1. TPM 1.2 Quote

The [RFC8639] notification format includes the <eventTime> object. This can be used to determine the amount of time subsequent to the initial subscription each notification was sent. However, this time is not part of the signed results which are returned from the Quote and therefore is not trustworthy as objects returned as part of the Quote. Therefore, a Verifier MUST periodically issue a new nonce and receive this nonce within a TPM quote response in order to ensure the freshness of the results. This can be done using the <tpm12-challenge-response-attestation> RPC from [RFC9684].

#### 3.2.2. TPM 2 Quote

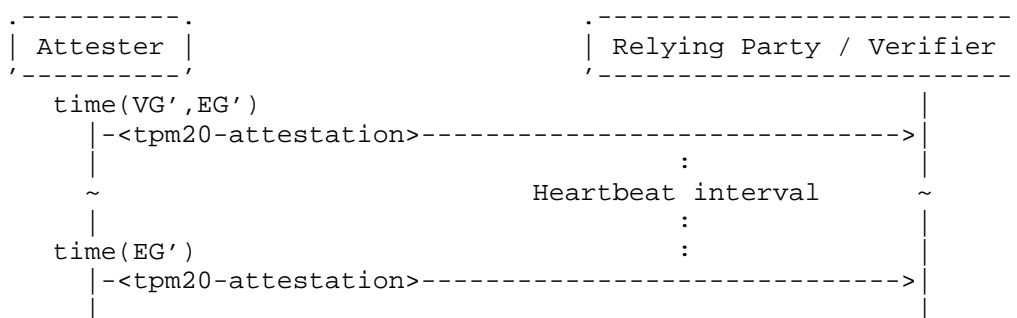
When the Attester includes a TPM2-compliant cryptoprocessor, internal time-related counters are included within the signed TPM Quote. By including an initial nonce in the [RFC8639] subscription request, fresh values for these counters are pushed to the Verifier as part of the first TPM Quote. As shown by [I-D.birkholz-rats-tuda], subsequent TPM Quotes delivered to the Verifier out-of-band can be appraised for freshness based on the predictable incrementing of these time-related counters.

The relevant internal time-related counters defined within [TPM2.0] can be seen within <tpms-clock-info>. These counters include the <clock>, <reset-counter>, and <restart-counter> objects. The rules for appraising these objects are as follows:

- \* If the <clock> has incremented for no more than the same duration as both the <eventTime> and the Verifier's internal time since the initial time(EG) and any previous time(EG'), then the TPM Quote may be considered fresh. Note that [TPM2.0] allows for +/- 15% clock drift. However, many hardware implementations significantly

improve on this maximum drift. If available, chip specific maximum drifts SHOULD be considered during the appraisal procedure of the Verifier.

- \* If the <reset-counter>, <restart-counter> has incremented. The existing subscription MUST be terminated, and a new <establish-subscription> SHOULD be generated.
- \* If a TPM Quote on any subscribed PCR has not been pushed to the Verifier for a duration of an Attester defined heartbeat interval, then a new TPM Quote notification SHOULD be sent to the Verifier. This may often be the case, as certain PCRs might be infrequently updated.



#### 4. Remote Attestation Event Stream

The <attestation> Event Stream is an [RFC8639] compliant Event Stream which is defined within this section and within the YANG Module of [RFC9684]. This Event Stream contains YANG notifications which carry Evidence to assist a Verifier in appraising the Trustworthiness Level of an Attester. Data Nodes within Section 4.6 allow the configuration of this Event Stream's contents on an Attester.

This <attestation> Event Stream may only be exposed on Attesters supporting [RFC9683]. As with [RFC9683], it is up to the Verifier to understand which types of cryptoprocessors and keys are acceptable.

##### 4.1. Subscription to the <attestation> Event Stream

To establish a subscription to an Attester in a way which provides provably fresh Evidence, initial randomness must be provided to the Attester. This is done via the augmentation of a <nonce-value> into [RFC8639] the <establish-subscription> RPC. Additionally, a Verifier must ask for PCRs of interest from a platform.

```
augment /sn:establish-subscription/sn:input:
  +---w nonce-value      binary
  +---w pcr-index*       tpm:pcr
```

The result of the subscription will be that passing of the following information:

1. <tpm12-attestation> and <tpm20-attestation> notifications which include the provided <nonce-value>. These attestation notifications MUST at least include all the <pcr-indicies> requested in the RPC.
2. a series of <pcr-extend> notifications which reference the requested PCRs on all TPM based cryptoprocessors on the Attester.
3. <tpm12-attestation> and <tpm20-attestation> notifications generated within a few seconds of the <pcr-extend> notifications. These attestation notifications MUST at least include any PCRs extended.

If the Verifier does not want to see the logged extend operations for all PCRs available from an Attester, an Event Stream Filter should be applied. This filter will remove Evidence from any PCRs which are not interesting to the Verifier.

#### 4.2. Replaying a History of Previous TPM Extend Operations

Unless it is relying on Reference Values for TPM Quotes only, a Verifier will need to acquire a history of PCR extensions since the Attester has been booted. This history may be requested from the Attester as part of the <establish-subscription> RPC. This request is accomplished by placing a very old <replay-start-time> within the original RPC request. As the very old <replay-start-time> will pre-date the time of Attester boot, a <replay-start-time-revision> will be returned in the <establish-subscription> RPC response, indicating when the Attester booted. Immediately following the response (and before the notifications above) one or more <pcr-extend> notifications which document all extend operations which have occurred for the requested PCRs since boot will be sent. Multiple extend operations to a single PCR index on a single TPM can be included within a single notification.

Note that if a Verifier has a partial history of extensions, the <replay-start-time> can be adjusted so that already known extensions are not forwarded.

The end of this history replay will be indicated with the [RFC8639] <replay-completed> notification. For more on this sequence, see Section 2.4.2.1 of [RFC8639].

After the <replay-complete> notification is provided, a TPM Quote will be requested and the result passed to the Verifier via a <tpm12-attestation> and <tpm20-attestation> notification. If there have been any additional extend operations which have changed a subscribed PCR value in this quote, these MUST be pushed to the Verifier before the <tpm12-attestation> and <tpm20-attestation> notification.

At this point, the Verifier has sufficient Evidence to appraise the reported extend operations for each PCR, as well as to compare a Reference Value derived from the replay of the Event Log history of extensions of the PCR value against those extensions signed by the TPM in its most recent Quote.

#### 4.2.1. TPM2 Heartbeat

For TPM 2.0, every requested PCR MUST be sent within an <tpm20-attestation> and no less frequent than once per heartbeat interval. This MAY be done with a single <tpm20-attestation> notification that includes all requested PCRs inside every heartbeat interval. This MAY be done with several <tpm20-attestation> notifications at different times during a heartbeat interval.

### 4.3. YANG Notifications Placed on the <attestation> Event Stream

#### 4.3.1. pcr-extend

This notification type documents when a subscribed PCR is extended within a single TPM cryptoprocessor. It SHOULD be emitted no less than the <marshalling-period> after the PCR is first extended. (The reason for the marshalling is that it is quite possible that multiple extensions to the same PCR have been made in quick succession, and these should be reflected in the same notification.) This notification MUST be emitted prior to a <tpm12-attestation> or <tpm20-attestation> notification which has included and signed the results of any specific PCR extension. If PCR extending events occur during the generation of the <tpm12-attestation> or <tpm20-attestation> notification, the marshalling period MUST be extended so that a new <pcr-extend> is not sent until the corresponding notifications have been sent.

```

+---n pcr-extend
+--ro certificate-name      certificate-name-ref
+--ro pcr-index-changed*   tpm:pcr
+--ro attested-event* []
+--ro attested-event
+--ro extended-with        binary
+--ro (event-details)?
+--:(bios-event-log) {tpm:bios}?
|   +--ro bios-event-entry* [event-number]
|   |   +--ro event-number      uint32
|   |   +--ro event-type?       uint32
|   |   +--ro pcr-index?        pcr
|   |   +--ro digest-list* []
|   |   |   +--ro hash-algo?    identityref
|   |   |   +--ro digest*       binary
|   |   +--ro event-size?       uint32
|   |   +--ro event-data*       uint8
+--:(ima-event-log) {tpm:ima}?
|   +--ro ima-event-entry* [event-number]
|   |   +--ro event-number      uint64
|   |   +--ro ima-template?     string
|   |   +--ro filename-hint?    string
|   |   +--ro filedata-hash?    binary
|   |   +--ro filedata-hash-algorithm? string
|   |   +--ro template-hash-algorithm? string
|   |   +--ro template-hash?    binary
|   |   +--ro pcr-index?        pcr
|   |   +--ro signature?        binary
+--:(netequi-boot-event-log) {tpm:netequi_boot}?
|   +--ro boot-event-entry* [event-number]
|   |   +--ro event-number      uint64
|   |   +--ro ima-template?     string
|   |   +--ro filename-hint?    string
|   |   +--ro filedata-hash?    binary
|   |   +--ro filedata-hash-algorithm? string
|   |   +--ro template-hash-algorithm? string
|   |   +--ro template-hash?    binary
|   |   +--ro pcr-index?        pcr
|   |   +--ro signature?        binary

```

Each <pcr-extend> MUST include one or more values being extended into the PCR. These are passed within the <extended-with> object. For each extension, details of the event SHOULD be provided within the <event-details> object. The format of any included <event-details> is identified by the <event-type>. This document includes two YANG structures which may be inserted into the <event-details>. These two structures are: <ima-event-log> and <bios-event-log>. Implementations wanting to provide additional documentation of a type of PCR extension may choose to define additional YANG structures which can be placed into <event-details>.

#### 4.3.2. tpm12-attestation

This notification contains an instance of a TPM1.2 style signed cryptoprocessor measurement. It is supplemented by Attester information which is not signed. This notification is generated and emitted from an Attester when at least one PCR identified within the subscribed <pcr-indices> has changed from the previous <tpm12-attestation> notification. This notification MUST NOT include the results of any PCR extensions not previously reported by a <pcr-extend>. This notification SHOULD be emitted as soon as a TPM Quote can extract the latest PCR hashed values. This notification MUST be emitted prior to a subsequent <pcr-extend>.

```
+---n tpm12-attestation {taa:TPM12}?
  +--ro certificate-name      tpm:certificate-name-ref
  +--ro up-time?              uint32
  +--ro TPM_QUOTE2?           binary
  +--ro TPM12-hash-algo?      identityref
  +--ro unsigned-pcr-values* []
    +--ro pcr-index*          tpm:pcr
    +--ro pcr-value*          binary
```

All YANG objects above are defined within [RFC9684]. The <tpm12-attestation> is not replayable.

#### 4.3.3. tpm20-attestation

This notification contains an instance of TPM2 style signed cryptoprocessor measurements. It is supplemented by Attester information which is not signed. This notification is generated at two points in time:

- \* every time at least one PCR has changed from a previous <tpm20-attestation>. In this case, the notification SHOULD be emitted within 10 seconds of the corresponding <pcr-extend> being sent:

- \* after a locally configurable minimum heartbeat period since a previous <tpm20-attestation> was sent.

```
+---n tpm20-attestation {taa:TPM20}?  
  +--ro certificate-name      tpm:certificate-name-ref  
  +--ro TPMS_QUOTE_INFO      binary  
  +--ro quote-signature?     binary  
  +--ro up-time?             uint32  
  +--ro unsigned-pcr-values* []  
    +--ro TPM20-hash-algo?   identityref  
    +--ro pcr-values* [pcr-index]  
      +--ro pcr-index        pcr  
      +--ro pcr-value?       binary
```

All YANG objects above are defined within [RFC9684]. The <tpm20-attestation> is not replayable.

#### 4.4. Filtering Evidence at the Attester

It can be useful not to receive all Evidence related to a PCR. An example of this is when a Verifier maintains Reference Values (known good values) of a PCR. In this case, it is not necessary to send a log of each consecutive extend operation.

To accomplish this reduction, when an RFC8639 <establish-subscription> RPC is sent, a <stream-filter> as per RFC8639, Section 2.2 can be set to discard a <pcr-extend> notification when the <pcr-index-changed> is uninteresting to the verifier.

#### 4.5. Replaying Previous PCR Extend Events

To verify the value of a PCR, a Verifier must either know that the value is a "known good" value (see Section 2.3.3 of [KGV] about Reference Values) or be able to reconstruct the hash value by viewing all the PCR-Extends since the Attester rebooted. Wherever a hash reconstruction might be needed, the <attestation> Event Stream MUST support the RFC8639 <replay> feature. Through the <replay> feature, it is possible for a Verifier to retrieve and sequentially hash all of the PCR extending events since an Attester booted. Thereby, the Verifier has access to all the Evidence needed to verify a PCR's current value.

#### 4.6. Configuring the <attestation> Event Stream

Figure 3 is tree diagram which exposes the operator configurable elements of the <attestation> Event Stream. This allows an Attester to select what information should be available on the stream. A fetch operation also allows an external device such as a Verifier to understand the current configuration of the stream.

Almost all YANG objects below are defined via reference from [RFC9684]. However, there is one object which is new in this model. <tpm2-heartbeat> defines the maximum amount of time which should pass before a subscriber to the Event Stream should get a <tpm20-attestation> notification from devices which contain a TPM2.

```

augment /tpm:rats-support-structures:
  +--rw tras:marshalling-period?                uint8
  +--rw tras:tpm12-subscribed-signature-scheme?
  |   -> ../tpm:attester-supported-algos/tpm12-asymmetric-signing
  |   {taa:TPM12}?
  +--rw tras:tpm20-subscribed-signature-scheme?
  |   -> ../tpm:attester-supported-algos/tpm20-asymmetric-signing
  |   {taa:TPM20}?
  +--rw tras:tpm20-subscription-heartbeat?      uint16
  |   {taa:TPM20}?

augment /tpm:rats-support-structures/tpm:tpms:
  +--rw tras:subscription-aik?                  tpm:certificate-name-ref
  +--rw (tras:subscribable)?
  |   +--:(tras:tpm12-stream) {taa:tpm12}?
  |   |   +--rw tras:tpm12-hash-algo?          identityref
  |   |   +--rw tras:tpm12-pcr-index*          tpm:pcr
  |   +--:(tras:tpm20-stream) {taa:tpm20}?
  |   |   +--rw tras:tpm20-hash-algo?          identityref
  |   |   +--rw tras:tpm20-pcr-index*          tpm:pcr

```

Figure 3: Configuring the <attestation> Event Stream

#### 5. YANG Module

This YANG module imports modules from [RFC9684] and [RFC8639].

```

<CODE BEGINS> ietf-tpm-remote-attestation-stream@2026-04-10.yang
module ietf-tpm-remote-attestation-stream {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-tpm-remote-attestation-stream";
  prefix tras;

```

```
import ietf-subscribed-notifications {
  prefix sn;
  reference
    "RFC 8639: Subscription to YANG Notifications";
}
import ietf-tpm-remote-attestation {
  prefix tpm;
  reference
    "draft-ietf-rats-yang-tpm-charra";
}
import ietf-tcg-algs {
  prefix taa;
}

organization "IETF";
contact
  "WG Web:    <http://tools.ietf.org/wg/rats/>
  WG List:    <mailto:rats@ietf.org>

  Editor:     Eric Voit
              <mailto:evoit@cisco.com>";

description
  "This module contains YANG specification for subscribing
  to attestation streams which contain events that have
  been generated by TPM chips or equivalent hardware
  implementations that include the protected capabilities
  as provided by TPMs.

  Copyright (c) 2024 IETF Trust and the persons identified
  as authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with
  or without modification, is permitted pursuant to, and
  subject to the license terms contained in, the Simplified
  BSD License set forth in Section 4.c of the IETF Trust's
  Legal Provisions Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX
  (https://www.rfc-editor.org/info/rfcXXXX); see the RFC
  itself for full legal notices.";

revision 2024-07-06 {
  description
    "Initial version.";
  reference
    "draft-ietf-rats-network-device-subscription";
```

```
}

/*
 * IDENTITIES
 */

identity pcr-unsubscribable {
  base sn:establish-subscription-error;
  description
    "Requested PCR is unsubscribable by the Attester.";
}

/*
 * Groupings
 */

grouping heartbeat {
  description
    "Allows an Attester to push verifiable, current TPM PCR values
    even when there have been no recent changes to PCRs.";
  leaf tpm20-subscription-heartbeat {
    type uint16;
    units "seconds";
    description
      "Number of seconds before the Attestation stream should send
      a new notification with a fresh quote. This allows
      confirmation that the PCR values haven't changed since the
      last tpm20-attestation.";
  }
}

/*
 * RPCs
 */

augment "/sn:establish-subscription/sn:input" {
  when 'derived-from-or-self(sn:stream, "attestation)';
  description
    "This augmentation adds a nonce to the subscription parameters
    that apply specifically to datastore updates to RPC input.";
  uses tpm:nonce;
  leaf-list pcr-index {
    type tpm:pcr;
    min-elements 1;
    description
      "The numbers/indexes of the PCRs. This will act as a filter
```

```
        for the subscription so that 'tpm-extend' notifications
        related to non-requested PCRs will not be sent to a
        subscriber.";
    }
}

/*
 * NOTIFICATIONS
 */

notification pcr-extend {
  description
    "This notification indicates that one or more PCRs have been
    extended within a TPM based cryptoprocessor. In less than the
    'marshalling-period', it MUST be followed with either a
    corresponding tpm12-attestation or tpm20-attestation
    notification which exposes the result of the PCRs updated.";
  uses tpm:certificate-name-ref;
  leaf-list pcr-index-changed {
    type tpm:pcr;
    min-elements 1;
    description
      "The number of each PCR extended. This list MUST contain the
      set of PCRs descibed within the event log details. This leaf
      can be derived from the list of attested events, but exposing
      it here allows for easy filtering of the notifications of
      interest to a verifier.";
  }
  list attested-event {
    description
      "A set of events which extended an Attester PCR. The
      sequence of elements represented in list must match the
      sequence of events placed into the TPM's PCR.";
    container attested-event {
      description
        "An instance of an event which extended an Attester PCR";
      leaf extended-with {
        type binary;
        mandatory true;
        description
          "Information extending the PCR.";
      }
      choice event-details {
        description
          "Contains the event happened the Attester thought
          was worthy of recording in a PCR.

          choices are of types defined by the identityref
```

```

        base tpm:attested_event_log_type";
    case bios-event-log {
        if-feature "tpm:bios";
        description
            "BIOS/UEFI event log format";
        uses tpm:bios-event-log;
    }
    case ima-event-log {
        if-feature "tpm:ima";
        description
            "IMA event log format";
        uses tpm:ima-event-log;
    }
    case netequip-boot-event-log {
        if-feature "tpm:netequip_boot";
        description
            "IMA event log format";
        uses tpm:network-equipment-boot-event-log;
    }
}
}
}
}

notification tpm12-attestation {
    if-feature "taa:tpm12";
    description
        "Contains an instance of TPM1.2 style signed cryptoprocessor
        measurements. It is supplemented by unsigned Attester
        information.";
    leaf certificate-name {
        type tpm:certificate-name-ref;
        mandatory true;
        description
            "Allows a TPM quote to be associated with a certificate.";
    }
    uses tpm:tpm12-attestation;
    uses tpm:tpm12-hash-algo;
    list unsigned-pcr-values {
        description
            "Allows notifications to be filtered by PCR number or
            PCR value based on via YANG related mechanisms such as PATH.
            This is done without requiring the filtering structure to be
            applied against TCG structured data.";
        leaf-list pcr-index {
            type tpm:pcr;
            min-elements 1;
            description

```

```
        "PCR index number.";
    }
    leaf-list pcr-value {
        type binary;
        description
            "PCR value in a sequence which matches to the
             'pcr-index'.";
    }
}

notification tpm20-attestation {
    if-feature "taa:tpm20";
    description
        "Contains an instance of TPM2 style signed cryptoprocessor
         measurements. It is supplemented by unsigned Attester
         information.";
    leaf certificate-name {
        type tpm:certificate-name-ref;
        mandatory true;
        description
            "Allows a TPM quote to be associated with a certificate.";
    }
    uses tpm:tpm20-attestation {
        description
            "Provides the attestation info. Also ensures PCRs can be
             XPATH filtered by refining the unsigned data so that it
             appears.";
        refine unsigned-pcr-values {
            min-elements 1;
        }
        refine unsigned-pcr-values/pcr-values {
            min-elements 1;
        }
    }
}

/*
 * DATA NODES
 */

augment "/tpm:rats-support-structures" {
    description
        "Defines platform wide 'attestation' stream subscription
         parameters.";
    leaf marshalling-period {
        type uint8;
    }
}
```

```
    default 5;
    description
        "The maximum number of seconds between the time an event
        extends a PCR, and the 'tpm-extend' notification which
        reports it to a subscribed Verifier. This period allows
        multiple extend operations bundled together and handled as a
        group.";
}
leaf tpm12-subscribed-signature-scheme {
    if-feature "taa:tpm12";
    type leafref {
        path "../tpm:attester-supported-algos" +
            "/tpm:tpm12-asymmetric-signing";
    }
    description
        "A single signature-scheme which will be used to sign the
        evidence from a TPM 1.2. which is then placed onto the
        'attestation' event stream.";
}
leaf tpm20-subscribed-signature-scheme {
    if-feature "taa:tpm20";
    type leafref {
        path "../tpm:attester-supported-algos" +
            "/tpm:tpm20-asymmetric-signing";
    }
    description
        "A single signature-scheme which will be used to sign the
        evidence from a TPM 2.0. which is then placed onto the
        'attestation' event stream.";
}
uses heartbeat{
    if-feature "taa:tpm20";
}
}

augment "/tpm:rats-support-structures/tpm:tpms" {
    description
        "Allows the configuration 'attestation' stream parameters for a
        TPM.";
    leaf subscription-aik {
        type tpm:certificate-name-ref;
        description
            "Identifies the certificate-name associated with the
            notifications in the 'attestation' stream.";
    }
    choice subscribable {
        config true;
        description
```

```
"Indicates that the set of notifications which comprise the
'attestation' event stream can be modified or tuned by a
network administrator.";
case tpm12-stream {
  if-feature "taa:tpm12";
  description
    "Configuration elements for a TPM1.2 event stream.";
  uses tpm:tpm12-hash-algo;
  leaf-list tpm12-pcr-index {
    type tpm:pcr;
    description
      "The numbers/indexes of the PCRs which can be
        subscribed.";
  }
}
case tpm20-stream {
  if-feature "taa:tpm20";
  description
    "Configuration elements for a TPM2.0 event stream.";
  uses tpm:tpm20-hash-algo;
  leaf-list tpm20-pcr-index {
    type tpm:pcr;
    description
      "The numbers/indexes of the PCRs which can be
        subscribed.";
  }
}
}
}
}
<CODE ENDS>
```

## 6. Event Streams for Conceptual Messages

Analogous to the [RFC8639] compliant <attestation> Event Stream for the conveyance of remote attestation Evidence as defined in Section Section 4, additional Event Streams can be defined for this YANG augment. Additional Event Streams require separate YANG augment specifications that provide the Event Stream definition and optionally a content format definition either via subscriptions to YANG datastores or dedicated YANG Notifications. It is possible to use either YANG subscription methods to other YANG modules for RATS Conceptual Messages or to define Event Streams for other none-YANG-modeled data. In the context of RATS Conceptual Messages, both options MUST be a specified via YANG augments to this specification.

## 7. Privacy Considerations

The privacy considerations of [RFC9683] (Remote Integrity Verification of Network Devices Containing Trusted Platform Modules) apply. Additionally, the security considerations from [RFC8641] (Subscription to YANG Notifications for Datastore Updates) how information about the system's internal structures or capabilities can be leaked, which could impact personally identifiable information (PII), apply.

There are no additional privacy considerations introduced by this document.

## 8. Security Considerations

### 8.1. Remote ATttestation procedureS (RATS)

The security considerations of [RFC9684] and [RFC9683] apply.

### 8.2. Yet Another Next Generation (YANG)

The security requirements (Section 4.2.5 of [RFC7923]) and the security considerations (Section 5 of [RFC7923]) from RFC7923 (Requirements for Subscription to YANG Datastores) apply. Subscription to YANG Notifications for Datastore Updates ([RFC8641]) illustrates specific security considerations concerning YANG Notifications for Datastore Updates. For example, it provides guidance on identifying sensitive writable subtrees and sensitive readable nodes.

### 8.3. Other

There are no additional security considerations introduced by this document.

## 9. IANA Considerations

This document registers the following namespace URIs in the [xml-registry] as per [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-tpm-remote-attestation-stream

Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

This document registers the following YANG module in the registry [yang-parameters] as per Section 14 of [RFC6020]:

Name: ietf-tpm-remote-attestation-stream

Namespace: urn:ietf:params:xml:ns:yang:ietf-tpm-remote-attestation-stream

Prefix: tras

Reference: draft-ietf-rats-network-device-subscription (RFC form)

## 10. References

### 10.1. Normative References

#### [I-D.ietf-rats-reference-interaction-models]

Birkholz, H., Eckel, M., Pan, W., and E. Voit, "Reference Interaction Models for Remote Attestation Procedures", Work in Progress, Internet-Draft, draft-ietf-rats-reference-interaction-models-17, 2 April 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-reference-interaction-models-17>>.

[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>>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/rfc/rfc3688>>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/rfc/rfc6020>>.

[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>>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/rfc/rfc8639>>.

[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>>.

- [RFC9683] Fedorkow, G. C., Ed., Voit, E., and J. Fitzgerald-McKay, "Remote Integrity Verification of Network Devices Containing Trusted Platform Modules", RFC 9683, DOI 10.17487/RFC9683, December 2024, <<https://www.rfc-editor.org/rfc/rfc9683>>.
- [RFC9684] Birkholz, H., Eckel, M., Bhandari, S., Voit, E., Sulzen, B., Xia, L., Laffey, T., and G. C. Fedorkow, "A YANG Data Model for Challenge-Response-Based Remote Attestation (CHARRA) Procedures Using Trusted Platform Modules (TPMs)", RFC 9684, DOI 10.17487/RFC9684, December 2024, <<https://www.rfc-editor.org/rfc/rfc9684>>.
- [TPM2.0] TCG, "TPM 2.0 Library Specification", n.d., <<https://trustedcomputinggroup.org/resource/tpm-library-specification/>>.

## 10.2. Informative References

- [I-D.birkholz-rats-tuda]  
Fuchs, A., Birkholz, H., McDonald, I., and C. Bormann, "Time-Based Uni-Directional Attestation", Work in Progress, Internet-Draft, draft-birkholz-rats-tuda-07, 10 July 2022, <<https://datatracker.ietf.org/doc/html/draft-birkholz-rats-tuda-07>>.
- [KGV] TCG, "KGV", October 2003, <[https://trustedcomputinggroup.org/wp-content/uploads/TCG-NetEq-Attestation-Workflow-Outline\\_v1r9b\\_pubrev.pdf](https://trustedcomputinggroup.org/wp-content/uploads/TCG-NetEq-Attestation-Workflow-Outline_v1r9b_pubrev.pdf)>.
- [RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <<https://www.rfc-editor.org/rfc/rfc7923>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/rfc/rfc8641>>.
- [TCG-Glossary]  
TCG, "TCG Glossary Version 1.1", May 2017, <<https://trustedcomputinggroup.org/wp-content/uploads/TCG-Glossary-V1.1-Rev-1.0.pdf>>.
- [xml-registry]  
"IETF XML Registry", n.d., <<https://www.iana.org/assignments/xml-registry/xml-registry.xhtml>>.

[yang-parameters]  
"YANG Parameters", n.d.,  
<<https://www.iana.org/assignments/yang-parameters/yang-parameters.xhtml>>.

#### Acknowledgements

Shout-out to Thomas Fossati, Zhuoyao Lin, Yogesh Deshpande, Jun Zhang, Thanassis Giannetsos, Michael Richardson, Ned Smith, and Chunchi (Peter) Liu for their extensive review feedback that was vital to produce this document.

#### Authors' Addresses

Henk Birkholz  
Fraunhofer SIT  
Rheinstrasse 75  
64295 Darmstadt  
Germany  
Email: [henk.birkholz@ietf.contact](mailto:henk.birkholz@ietf.contact)

Eric Voit  
Cisco Systems, Inc.  
Email: [evoit@cisco.com](mailto:evoit@cisco.com)

Wei Pan  
Huawei Technologies  
101 Software Avenue, Yuhuatai District  
Nanjing, Jiangsu  
210012  
China  
Email: [william.panwei@huawei.com](mailto:william.panwei@huawei.com)