Stream: Internet Engineering Task Force (IETF)
RFC: 9684
Category: Standards Track
Published: November 2024
ISSN: 2070-1721
Authors: H. Birkholz, *Fraunhofer SIT | ATHENE*; M. Eckel, *Fraunhofer SIT | ATHENE*; S. Bhandari, *ThoughtSpot*; E. Voit, *Cisco*; B. Sulzen, *Cisco*; L. Xia, *Huawei*; T. Laffey, *HPE*; G. C. Fedorkow, *Juniper*

RFC 9684
A YANG Data Model for Challenge-Response-Based Remote Attestation (CHARRA) Procedures Using Trusted Platform Modules (TPMs)

Abstract

This document defines the YANG Remote Procedure Calls (RPCs) and configuration nodes that are required to retrieve attestation evidence about integrity measurements from a device, following the operational context defined in RFC 9683 "TPM-based Network Device Remote Integrity Verification". Complementary measurement logs originating from one or more Roots of Trust for Measurement (RTMs) are also provided by the YANG RPCs. The defined module requires the inclusion of the following in the device components of the composite device on which the YANG server is running: at least one Trusted Platform Module (TPM) of either version 1.2 or 2.0 as well as a corresponding TPM Software Stack (TSS), or an equivalent hardware implementation that includes the protected capabilities as provided by TPMs as well as a corresponding software stack.

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/rfc9684.

Copyright Notice

Table of Contents

1. Introduction

This document is based on the general terminology defined in Remote ATtestation procedureS (RATS) architecture [RFC9334] and uses the operational context defined in [RFC9683] as well as the interaction model and information elements defined in [RATS-Interaction-Models]. The

currently supported hardware security modules (HSMs) are the Trusted Platform Modules (TPMs) [TPM1.2] [TPM2.0] as specified by the Trusted Computing Group (TCG). One TPM, or multiple TPMs in the case of a composite device, is required in order to use the YANG module defined in this document. Each TPM is used as a Root of Trust for Storage (RTS) in order to store system security measurement Evidence. And each TPM is used as a Root of Trust for Reporting (RTR) in order to retrieve attestation Evidence. This is done by using a YANG RPC to request a quote that exposes a rolling hash of the security measurements held internally within the TPM.

Specific terms imported from [RFC9334] and used in this document include Attester, composite device, and Evidence.

Specific terms imported from [TPM2.0-Key] and used in this document include Endorsement Key (EK), Initial Attestation Key (IAK), Attestation Identity Key (AIK), and Local Attestation Key (LAK).

1.1. 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.

2. The YANG Module for Basic Remote Attestation Procedures

One or more TPMs **MUST** be embedded in a composite device that provides attestation Evidence via the YANG module defined in this document. The ietf-tpm-remote-attestation YANG module enables a composite device to take on the role of an Attester, in accordance with the RATS architecture [RFC9334] and the corresponding challenge-response interaction model defined in [RATS-Interaction-Models]. A fresh nonce with an appropriate amount of entropy [NIST-915121] **MUST** be supplied by the YANG client in order to enable a proof-of-freshness with respect to the attestation Evidence provided by the Attester running the YANG datastore. Further, this nonce is used to prevent replay attacks. The method for communicating the relationship of each individual TPM to the specific measured component within the composite device is out of the scope of this document.

2.1. YANG Modules

In this section, the two YANG modules are defined.

2.1.1. ietf-tpm-remote-attestation

This YANG module imports modules from [RFC6991] with prefix 'yang', [RFC8348] with prefix 'hw', [RFC9642] with prefix 'ks', and ietf-tcg-algs.yang Section 2.1.2.3 with prefix 'taa'. Additionally, references are made to [RFC6933], [TPM1.2-Commands], [TPM2.0-Arch], [TPM2.0-Structures], [TPM2.0-Key], [TPM1.2-Structures], [BIOS-Log], and [CEL], as well as Appendix B.

2.1.1.1. Features

This module supports the following features:

'mtpm': Indicates that multiple TPMs on the device can support remote attestation. For example, this feature could be used in cases where multiple line cards are present, each with its own TPM.

'bios': Indicates that the device supports the retrieval of BIOS and Unified Extensible Firmware Interface (UEFI) event logs [BIOS-Log].

'ima': Indicates that the device supports the retrieval of event logs from the Linux Integrity Measurement Architecture (IMA, see Appendix A).

'netequip_boot': Indicates that the device supports the retrieval of netequip boot event logs. See Appendixes A and B.

2.1.1.2. Identities

This module supports the following types of attestation event logs: 'bios', 'ima', and 'netequip_boot'.

2.1.1.3. Remote Procedure Calls (RPCs)

In the following sections, RPCs for attestation procedures for both TPM 1.2 and TPM 2.0 are defined.

2.1.1.3.1. tpm12-challenge-response-attestation

This RPC allows a Verifier to request via the *TPM Quote* operation, signed TPM Platform Configuration Registers (PCRs) from a cryptoprocessor compliant with TPM 1.2. Where the feature 'mtpm' is active, and one or more 'certificate-name' is not provided, all cryptoprocessors compliant with TPM 1.2 will respond. The YANG tree diagram of this RPC is as follows:

```
+---x tpm12-challenge-response-attestation {taa:tpm12}?
   +---w input
   |  +---w tpm12-attestation-challenge
   |     +---w pcr-index*          pcr
   |     +---w nonce-value         binary
   |     +---w certificate-name*   certificate-name-ref
   |             {tpm:mtpm}?
   +--ro output
      +--ro tpm12-attestation-response* []
         +--ro certificate-name    certificate-name-ref
         +--ro up-time?            uint32
         +--ro TPM_QUOTE2?         binary
```

2.1.1.3.2. tpm20-challenge-response-attestation

This RPC allows a Verifier to request signed TPM PCRs (*TPM Quote* operation) from a cryptoprocessor compliant with TPM 2.0. Where the feature 'mtpm' is active, and one or more 'certificate-name' is not provided, all cryptoprocessors compliant with TPM 2.0 will respond. The YANG tree diagram of this RPC is as follows:

```
+---x tpm20-challenge-response-attestation {taa:tpm20}?
   +---w input
   |  +---w tpm20-attestation-challenge
   |     +---w nonce-value            binary
   |     +---w tpm20-pcr-selection* []
   |     |  +---w tpm20-hash-algo?   identityref
   |     |  +---w pcr-index*         pcr
   |     +---w certificate-name*      certificate-name-ref
   |              {tpm:mtpm}?
   +--ro output
      +--ro tpm20-attestation-response* []
         +--ro certificate-name       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
```

An example of an RPC challenge requesting PCRs 0-7 from a SHA-256 bank could look like the following:

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <tpm20-attestation-challenge
    xmlns="urn:ietf:params:xml:ns:yang:ietf-tpm-remote-attestation">
    <certificate-name>
        (identifier of a TPM signature key with which the Attester is
        supposed to sign the attestation data)
    </certificate-name>
    <nonce>
    0xe041307208d9f78f5b1bbecd19e2d152ad49de2fc5a7d8dbf769f6b8ffdeab9
    </nonce>
    <tpm20-pcr-selection>
      <tpm20-hash-algo
          xmlns="urn:ietf:params:xml:ns:yang:ietf-tcg-algs">
        TPM_ALG_SHA256
      </tpm20-hash-algo>
      <pcr-index>0</pcr-index>
      <pcr-index>1</pcr-index>
      <pcr-index>2</pcr-index>
      <pcr-index>3</pcr-index>
      <pcr-index>4</pcr-index>
      <pcr-index>5</pcr-index>
      <pcr-index>6</pcr-index>
      <pcr-index>7</pcr-index>
    </tpm20-pcr-selection>
  </tpm20-attestation-challenge>
</rpc>
```

A successful response could be formatted as follows:

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <tpm20-attestation-response
    xmlns="urn:ietf:params:xml:ns:yang:ietf-tpm-remote-attestation">
    <certificate-name
        xmlns="urn:ietf:params:xml:ns:yang:ietf-keystore">
        (instance of certificate name in the keystore)
    </certificate-name>
    <attestation-data>
       (raw attestation data, i.e., the TPM quote; this includes,
       among other information, a composite digest of requested PCRs,
       the nonce, and TPM 2.0 clock information.)
    </attestation-data>
    <quote-signature>
        (signature over attestation-data using the TPM key
        identified by sig-key-id)
    </quote-signature>
  </tpm20-attestation-response>
</rpc-reply>
```

2.1.1.4. log-retrieval

This RPC allows a Verifier to acquire the Evidence that was extended into specific TPM PCRs. The YANG tree diagram of this RPC is as follows:

```
+---x log-retrieval
   +---w input
   |  +---w log-type        identityref
   |  +---w log-selector* []
   |     +---w name*                      string
   |     +---w (index-type)?
   |     |  +--:(last-entry)
   |     |  |  +---w last-entry-value?    binary
   |     |  +--:(index)
   |     |  |  +---w last-index-number?   uint64
   |     |  +--:(timestamp)
   |     |     +---w timestamp?           yang:date-and-time
   |     +---w log-entry-quantity?        uint16
   +--ro output
      +--ro system-event-logs
         +--ro node-data* []
            +--ro name?         string
            +--ro up-time?      uint32
            +--ro log-result
               +--ro (attested_event_log_type)
                  +--:(bios) {bios}?
                  |  +--ro bios-event-logs
                  |     +--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*     binary
                    +--:(ima) {ima}?
                    |  +--ro ima-event-logs
                    |     +--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
                    +--:(netequip_boot) {netequip_boot}?
                       +--ro boot-event-logs
                          +--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
```

2.1.1.5. Data Nodes

This section provides a high-level description of the data nodes that contain the configuration and operational objects within the YANG data model. For more details, please see the YANG module itself in Figure 1.

Container 'rats-support-structures': This houses the set of information relating to remote attestation for a device. This includes specific device TPM(s), the compute nodes (such as line cards) on which the TPM(s) reside, and the algorithms supported across the platform.

Container 'tpms': This provides configuration and operational details for each supported TPM, including the tpm-firmware-version, PCRs that may be quoted, certificates that are associated with that TPM, and the current operational status. Of note are the certificates that are associated with that TPM. As a certificate is associated with a particular TPM Attestation Key, knowledge of the certificate allows a specific TPM to be identified.

```
+--rw tpms
   +--rw tpm* [name]
      +--rw name                string
      +--ro hardware-based      boolean
      +--ro physical-index?     int32 {hw:entity-mib}?
      +--ro path?               string
      +--ro compute-node        compute-node-ref {tpm:mtpm}?
      +--ro manufacturer?       string
      +--rw firmware-version    identityref
      +--rw tpm12-hash-algo?    identityref {taa:tpm12}?
      +--rw tpm12-pcrs*         pcr
      +--rw tpm20-pcr-bank* [tpm20-hash-algo]  {taa:tpm20}?
      |  +--rw tpm20-hash-algo    identityref
      |  +--rw pcr-index*         tpm:pcr
      +--ro status              enumeration
      +--rw certificates
         +--rw certificate* [name]
            +--rw name            string
            +--rw keystore-ref?   leafref {ks:asymmetric-keys}?
            +--rw type?           enumeration
```

Container 'attester-supported-algos': This identifies which TCG hash algorithms are available for use on the Attesting platform. An operator will use this information to limit algorithms available for use by RPCs to just a desired set from the universe of all hash algorithms allowed by the TCG.

```
+--rw attester-supported-algos
   +--rw tpm12-asymmetric-signing*   identityref {taa:tpm12}?
   +--rw tpm12-hash*                 identityref {taa:tpm12}?
   +--rw tpm20-asymmetric-signing*   identityref {taa:tpm20}?
   +--rw tpm20-hash*                 identityref {taa:tpm20}?
```

Container 'compute-nodes': When there is more than one TPM supported, this container maintains the set of information related to the compute node associated with a specific TPM. This allows each specific TPM to identify to which 'compute-node' it belongs.

```
+--rw compute-nodes {tpm:mtpm}?
   +--ro compute-node* [node-id]
      +--ro node-id                string
      +--ro node-physical-index?   int32 {hw:entity-mib}?
      +--ro node-name?             string
      +--ro node-location?         string
```

2.1.1.6. YANG Module

```
<CODE BEGINS> file "ietf-tpm-remote-attestation@2024-10-22.yang"

module ietf-tpm-remote-attestation {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang"
          + ":ietf-tpm-remote-attestation";
  prefix tpm;

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-hardware {
    prefix hw;
  }
  import ietf-keystore {
    prefix ks;
  }
  import ietf-tcg-algs {
    prefix taa;
  }

  organization
    "IETF RATS (Remote ATtestation procedureS) Working Group";
  contact
    "WG Web  : <https://datatracker.ietf.org/wg/rats/>
     WG List : <mailto:rats@ietf.org>
     Author  : Eric Voit <evoit@cisco.com>
     Author  : Henk Birkholz <henk.birkholz@ietf.contact>
     Author  : Michael Eckel <michael.eckel@sit.fraunhofer.de>
     Author  : Shwetha Bhandari <shwetha.bhandari@thoughtspot.com>
     Author  : Bill Sulzen <bsulzen@cisco.com>
     Author  : Liang Xia (Frank) <frank.xialiang@huawei.com>
     Author  : Tom Laffey <tom.laffey@hpe.com>
     Author  : Guy C. Fedorkow <gfedorkow@juniper.net>";
  description
    "A YANG module to enable remote attestation procedures based
     on TPM 1.2 and TPM 2.0 using a challenge-response
     interaction model and the Quote primitive operations defined
     by TPM 1.2 and TPM 2.0.

     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 (RFC 2119) (RFC 8174) when, and only when,
     they appear in all capitals, as shown here.

     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 Revised 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 9684; see the
```

```
     RFC itself for full legal notices.";

  revision 2024-10-22 {
    description
      "Initial version";
    reference
      "RFC 9684: A YANG Data Model for Challenge-Response-Based
       Remote Attestation (CHARRA) Procedures Using Trusted Platform
       Modules (TPMs)";
  }

  /*****************/
  /*   Features    */
  /*****************/

  feature mtpm {
    description
      "The device supports the remote attestation of multiple
       TPM-based cryptoprocessors.";
  }

  feature bios {
    description
      "The device supports the BIOS logs.";
    reference
      "BIOS-Log:
       TCG PC Client Platform Firmware Profile Specification,
       https://trustedcomputinggroup.org/wp-content/uploads/
       TCG-PC-Client-Platform-Firmware-Profile-Version-1.06-
       Revision-52_pub-2.pdf, Section 10.4.5.2";
  }

  feature ima {
    description
      "The device supports Integrity Measurement Architecture logs.
       Many variants of IMA logs exist in the deployment.  Each
       encodes the log entry contents as the specific measurements
       that get hashed into a PCRs as Evidence.  See the reference
       below for one example of such an encoding.";
    reference
      "CEL:
       Canonical Event Log Format,
       https://www.trustedcomputinggroup.org/wp-content/uploads/
       TCG_IWG_CEL_v1_r0p41_pub.pdf, Section 5.1.6";
  }

  feature netequip_boot {
    description
      "The device supports the netequip_boot logs.";
    reference
      "RFC 9684: A YANG Data Model for Challenge-Response-Based
       Remote Attestation (CHARRA) Procedures Using Trusted Platform
       Modules (TPMs), Appendix B";
  }

  /*****************/
  /*   Typedefs    */
  /*****************/
```

```
typedef pcr {
  type uint8 {
    range "0..31";
  }
  description
    "Valid index number for a PCR.  A PCR index compliant with
     TPM 2.0 extends from 0-31.  At this time, a typical TPM would
     have no more than 32 PCRs.";
}

typedef compute-node-ref {
  type leafref {
    path "/tpm:rats-support-structures/tpm:compute-nodes"
       + "/tpm:compute-node/tpm:node-id";
  }
  description
    "This type is used to reference a hardware node.  Note that an
     implementer might include an alternative leafref pointing to a
     different YANG module node specifying hardware structures.";
}

typedef certificate-name-ref {
  type leafref {
    path "/tpm:rats-support-structures/tpm:tpms/tpm:tpm"
       + "/tpm:certificates/tpm:certificate/tpm:name";
  }
  description
    "A type that allows identification of a TPM-based
     certificate.";
}

/******************/
/*   Identities   */
/******************/

identity attested_event_log_type {
  description
    "Base identity allowing categorization of the reasons why an
     attested measurement has been taken on an Attester.";
}

identity ima {
  base attested_event_log_type;
  description
    "An event type recorded in IMA.";
}

identity bios {
  base attested_event_log_type;
  description
    "An event type associated with BIOS/UEFI.";
}

identity netequip_boot {
  base attested_event_log_type;
  description
    "An event type associated with Network Equipment Boot.";
```

```
  }

  /*****************/
  /*   Groupings   */
  /*****************/

  grouping tpm20-hash-algo {
    description
      "The cryptographic algorithm used to hash the PCRs compliant
       with TPM 2.0.  This must be from the list of platform-
       supported options.";
    leaf tpm20-hash-algo {
      type identityref {
        base taa:hash;
      }
      must '. = /tpm:rats-support-structures'
         + '/tpm:attester-supported-algos/tpm:tpm20-hash' {
        error-message "This platform does not support "
                    + "tpm20-hash-algo";
      }
      description
        "The hash scheme that is used to hash a PCR compliant with
         TPM 2.0.  This must be one of those supported by a platform.
         Where this object does not appear, the default value of
         'taa:TPM_ALG_SHA256' will apply.";
    }
  }

  grouping tpm12-hash-algo {
    description
      "The cryptographic algorithm used to hash the PCRs compliant
       with TPM 1.2.";
    leaf tpm12-hash-algo {
      type identityref {
        base taa:hash;
      }
      must '. = /tpm:rats-support-structures'
         + '/tpm:attester-supported-algos/tpm:tpm12-hash' {
        error-message "This platform does not support "
                    + "tpm12-hash-algo";
      }
      description
        "The hash scheme that is used to hash a PCR compliant with
         TPM 1.2.  This MUST be one of those supported by a platform.
         Where this object does not appear, the default value of
         'taa:TPM_ALG_SHA1' will apply.";
    }
  }

  grouping nonce {
    description
      "A random number intended to guarantee freshness and for use
       as part of a replay-detection mechanism.";
    leaf nonce-value {
      type binary;
      mandatory true;
      description
        "A cryptographically generated random number that should
```

```
          not be predictable prior to its issuance from a random
          number generation function.  The random number MUST be
          derived from an entropy source external to the Attester.

          Note that a nonce sent into a TPM will typically be 160 or
          256 binary digits long.  (This is 20 or 32 bytes.)  So if
          fewer binary digits are sent, this nonce object will be
          padded with leading zeros within Quotes returned from the
          TPM.  Additionally, if more bytes are sent, the nonce will
          be trimmed to the most significant binary digits.";
    }
  }

  grouping tpm12-pcr-selection {
    description
      "A Verifier can request one or more PCR values using its
       individually created Attestation Key Certificate (AC).
       The corresponding selection filter is represented in this
       grouping.";
    leaf-list pcr-index {
      type pcr;
      description
        "The numbers/indexes of the PCRs.  In addition, any selection
         of PCRs MUST verify that the set of PCRs requested are a
         subset of the set of PCRs exposed in the leaf-list
         /tpm:rats-support-structures
         /tpm:tpms/tpm:tpm[name=current()]/tpm:tpm12-pcrs";
    }
  }

  grouping tpm20-pcr-selection {
    description
      "A Verifier can acquire one or more PCR values, which are
       hashed together in a TPM2B_DIGEST coming from the TPM2.
       The selection list of desired PCRs and the hash algorithm
       is represented in this grouping.";
    list tpm20-pcr-selection {
      unique "tpm20-hash-algo";
      description
        "Specifies the list of PCRs and hash algorithms that can be
         returned within a TPM2B_DIGEST.";
      reference
        "TPM2.0-Structures:
         Trusted Platform Module Library Part 2:  Structures,
         Revision 01.83, https://trustedcomputinggroup.org/
         wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf,
         Section 10.9.7";
      uses tpm20-hash-algo;
      leaf-list pcr-index {
        type pcr;
        description
          "The numbers of the PCRs that are being tracked
           with a hash based on the tpm20-hash-algo.  In addition,
           any selection of PCRs MUST verify that the set of PCRs
           requested are a subset of the set of selected PCR indexes
           available for that specific TPM.";
      }
    }
```

```
  }

  grouping certificate-name-ref {
    description
      "Identifies a certificate in a keystore.";
    leaf certificate-name {
      type certificate-name-ref;
      mandatory true;
      description
        "Identifies a certificate in a keystore.";
    }
  }

  grouping tpm-name {
    description
      "A unique TPM on a device.";
    leaf name {
      type string;
      description
        "Unique system-generated name for a TPM on a device.";
    }
  }

  grouping node-uptime {
    description
      "Uptime in seconds of the node.";
    leaf up-time {
      type uint32;
      description
        "Uptime in seconds of this node reporting its data.";
    }
  }

  grouping tpm12-attestation {
    description
      "Contains an instance of cryptoprocessor measurements signed
       according to TPM 1.2.  It is supplemented by unsigned
       Attester information.";
    uses node-uptime;
    leaf pcr-data {
      type binary;
      description
        "The value created and signed for the quote
         (type TPM_PCR_INFO_SHORT), i.e., the 'pcrData' part of
          a TPM1.2 Quote2 operation result.";
      reference
        "TPM1.2-Commands:
         TPM Main Part 3 Commands, Rev116,
         https://trustedcomputinggroup.org/wp-content/uploads
         /TPM-Main-Part-3-Commands_v1.2_rev116_01032011.pdf,
         Section 16.5";
    }
    leaf version-info {
      type binary;
      description
        "The version info (type TPM_CAP_VERSION_INFO),
         i.e., the 'versionInfo' part of a TPM1.2 Quote2
         operation result.";
```

```
      reference
        "TPM1.2-Commands:
         TPM Main Part 3 Commands, Rev116,
         https://trustedcomputinggroup.org/wp-content/uploads
         /TPM-Main-Part-3-Commands_v1.2_rev116_01032011.pdf,
         Section 16.5";
    }
    leaf sig {
      type binary;
      description
        "The signature generated across the signed data,
         i.e., the 'sig' part of a TPM1.2 Quote2 operation
         result.";
      reference
        "TPM1.2-Commands:
         TPM Main Part 3 Commands, Rev116,
         https://trustedcomputinggroup.org/wp-content/uploads
         /TPM-Main-Part-3-Commands_v1.2_rev116_01032011.pdf,
         Section 16.5";
    }
  }

  grouping tpm20-attestation {
    description
      "Contains an instance of cryptoprocessor measurements signed
       according to TPM 2.0.  It is supplemented by unsigned
       Attester information.";
    leaf quote-data {
      type binary;
      mandatory true;
      description
        "A hash of the latest PCR values (and the hash algorithm
         used) that have been returned from an Attester for the
         selected PCRs and hash algorithms.";
      reference
        "TPM2.0-Structures:
         Trusted Platform Module Library Part 2:  Structures,
         Revision 01.83, https://trustedcomputinggroup.org/
         wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf,
         Section 10.12.1";
    }
    leaf quote-signature {
      type binary;
      description
        "Quote signature returned by TPM Quote.  The signature was
         generated using the key associated with the
         certificate 'name'.";
      reference
        "TPM2.0-Structures:
         Trusted Platform Module Library Part 2:  Structures,
         Revision 01.83, https://trustedcomputinggroup.org/
         wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf,
         Section 11.2.1";
    }
    uses node-uptime;
    list unsigned-pcr-values {
      description
        "PCR values in each PCR bank.  This might appear redundant
```

```
          with the TPM2B_DIGEST, but that digest is calculated across
          multiple PCRs.  Having to verify across multiple PCRs does
          not necessarily make it easy for a Verifier to appraise just
          the minimum set of PCR information that has changed since
          the last received TPM2B_DIGEST.  Put another way, why should
          a Verifier reconstruct the proper value of all PCR Quotes
          when only a single PCR has changed?
          To help this happen, if the Attester does know specific PCR
          values, the Attester can provide these individual values via
          'unsigned-pcr-values'.  By comparing this information to
          what has previously been validated, it is possible for a
          Verifier to confirm the Attester's signature while
          eliminating significant processing.  Note that there should
          never be a result where an unsigned PCR value differs from
          what may be reconstructed from within the PCR quote and
          the event logs.
          If there is a difference, a signed result that has been
          verified from retrieved logs is considered definitive.";
      uses tpm20-hash-algo;
      list pcr-values {
        key "pcr-index";
        description
          "List of one PCR bank.";
        leaf pcr-index {
          type pcr;
          description
            "PCR index number.";
        }
        leaf pcr-value {
          type binary;
          description
            "PCR value.";
          reference
            "TPM2.0-Structures:
             Trusted Platform Module Library Part 2:  Structures,
             Revision 01.83, https://trustedcomputinggroup.org/
             wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf,
             Section 10.9.7";
        }
      }
    }
  }

  grouping log-identifier {
    description
      "Identifier for type of log to be retrieved.";
    leaf log-type {
      type identityref {
        base attested_event_log_type;
      }
      mandatory true;
      description
        "The corresponding identity of the measurement log type.";
    }
  }

  grouping boot-event-log {
    description
```

```
        "Defines a specific instance of an event log entry
         and corresponding to the information used to
         extend the PCR.";
      leaf event-number {
        type uint32;
        description
          "Unique event number of this event, which monotonically
           increases within a given event log.  The maximum event
           number should not be reached, nor is wrapping back to
           an earlier number supported.";
      }
      leaf event-type {
        type uint32;
        description
          "BIOS log event type.";
        reference
          "BIOS-Log:
           TCG PC Client Platform Firmware Profile Specification,
           https://trustedcomputinggroup.org/wp-content/uploads/
           TCG-PC-Client-Platform-Firmware-Profile-Version-1.06-
           Revision-52_pub-2.pdf, Section 10.4.1";
      }
      leaf pcr-index {
        type pcr;
        description
          "Defines the PCR index that this event extended.";
      }
      list digest-list {
        description
          "Hash of event data.";
        leaf hash-algo {
          type identityref {
            base taa:hash;
          }
          description
            "The hash scheme that is used to compress the event data in
             each of the leaf-list digest items.";
        }
        leaf-list digest {
          type binary;
          description
            "The hash of the event data using the algorithm of the
             'hash-algo' against 'event data'.";
        }
      }
      leaf event-size {
        type uint32;
        description
          "Size of the event data.";
      }
      leaf-list event-data {
        type binary;
        description
          "The event data.  This is a binary structure
           of size 'event-size'. For more on what
           might be recorded within this object
           see BIOS-Log, Section 10, which details
           viable events that might be recorded.";
```

```
      reference
        "BIOS-Log:
         TCG PC Client Platform Firmware Profile Specification,
         https://trustedcomputinggroup.org/wp-content/uploads/
         TCG-PC-Client-Platform-Firmware-Profile-Version-1.06-
         Revision-52_pub-2.pdf, Section 10";
    }
  }

  grouping bios-event-log {
    description
      "Measurement log created by the BIOS/UEFI.";
    list bios-event-entry {
      key "event-number";
      description
        "Ordered list of the TCG-described event log
         that extended the PCRs in the order they
         were logged.";
      uses boot-event-log;
    }
  }

  grouping ima-event {
    description
      "Defines a hash log extend event for IMA measurements.";
    reference
      "CEL:
       Canonical Event Log Format,
       https://www.trustedcomputinggroup.org/wp-content/uploads/
       TCG_IWG_CEL_v1_r0p41_pub.pdf, Section 4.3";
    leaf event-number {
      type uint64;
      description
        "Unique event number of this event, which monotonically
         increases.  The maximum event number should not be
         reached, nor is wrapping back to an earlier number
         supported.";
    }
    leaf ima-template {
      type string;
      description
        "Name of the template used for event logs,
         e.g., ima, ima-ng, ima-sig.";
    }
    leaf filename-hint {
      type string;
      description
        "File name (including the path) that was measured.";
    }
    leaf filedata-hash {
      type binary;
      description
        "Hash of filedata as updated based upon the
         filedata-hash-algorithm.";
    }
    leaf filedata-hash-algorithm {
      type string;
      description
```

```
        "Algorithm used for filedata-hash.";
    }
    leaf template-hash-algorithm {
      type string;
      description
        "Algorithm used for template-hash.";
    }
    leaf template-hash {
      type binary;
      description
        "hash(filedata-hash, filename-hint)";
    }
    leaf pcr-index {
      type pcr;
      description
        "Defines the PCR index that this event extended.";
    }
    leaf signature {
      type binary;
      description
        "Digital file signature that provides a
         fingerprint for the file being measured.";
    }
  }

  grouping ima-event-log {
    description
      "Measurement log created by IMA.";
    list ima-event-entry {
      key "event-number";
      description
        "Ordered list of IMA event logs by event-number.";
      uses ima-event;
    }
  }

  grouping network-equipment-boot-event-log {
    description
      "Measurement log created by Network Equipment Boot.  The
       Network Equipment Boot format is identical to the IMA
       format.  In contrast to the IMA log, the Network Equipment
       Boot log includes every measurable event from an Attester,
       including the boot stages of BIOS, Bootloader, etc. In
       essence, the scope of events represented in this format
       combines the scope of BIOS events and IMA events.";
    list boot-event-entry {
      key "event-number";
      description
        "Ordered list of Network Equipment Boot event logs
         by event-number, using the IMA event format.";
      uses ima-event;
    }
  }

  grouping event-logs {
    description
      "A selector for the log and its type.";
    choice attested_event_log_type {
```

```
      mandatory true;
      description
        "Event log type determines the event log's content.";
      case bios {
        if-feature "bios";
        description
          "BIOS/UEFI event logs.";
        container bios-event-logs {
          description
            "BIOS/UEFI event logs.";
          uses bios-event-log;
        }
      }
      case ima {
        if-feature "ima";
        description
          "IMA event logs.";
        container ima-event-logs {
          description
            "IMA event logs.";
          uses ima-event-log;
        }
      }
      case netequip_boot {
        if-feature "netequip_boot";
        description
          "Network Equipment Boot event logs.";
        container boot-event-logs {
          description
            "Network Equipment Boot event logs.";
          uses network-equipment-boot-event-log;
        }
      }
    }
  }

  /**********************/
  /*   RPC operations   */
  /**********************/

  rpc tpm12-challenge-response-attestation {
    if-feature "taa:tpm12";
    description
      "This RPC accepts the input for TSS TPM 1.2 commands made to
       the attesting device.";
    input {
      container tpm12-attestation-challenge {
        description
          "This container includes every information element defined
           in the reference challenge-response interaction model for
           remote attestation.  Corresponding values are based on
           TPM 1.2 structure definitions";
        uses tpm12-pcr-selection;
        uses nonce;
        leaf-list certificate-name {
          if-feature "tpm:mtpm";
          type certificate-name-ref;
          must "/tpm:rats-support-structures/tpm:tpms"
```

```
            + "/tpm:tpm[tpm:firmware-version='taa:tpm12']"
            + "/tpm:certificates/"
            + "/tpm:certificate[name=current()]" {
          error-message "Not an available TPM1.2 AIK certificate.";
        }
        description
          "When populated, the RPC will only get a Quote for the
           TPMs associated with these certificate(s).";
      }
    }
  }
  output {
    list tpm12-attestation-response {
      unique "certificate-name";
      description
        "The binary output of TPM 1.2 TPM_Quote/TPM_Quote2,
         including the PCR selection and other associated
         attestation evidence metadata.";
      uses certificate-name-ref {
        description
          "Certificate associated with this tpm12-attestation.";
      }
      uses tpm12-attestation;
    }
  }
}

rpc tpm20-challenge-response-attestation {
  if-feature "taa:tpm20";
  description
    "This RPC accepts the input for TSS TPM 2.0 commands of the
     managed device.  Composite devices may contain several TPMs;
     /hardware/component/physical-index from the hardware
     management YANG module is used to refer to dedicated TPMs in
     composite devices; however, devices without TPMs are not
     covered.";
  input {
    container tpm20-attestation-challenge {
      description
        "This container includes every information element defined
         in the reference challenge-response interaction model for
         remote attestation.  Corresponding values are based on
         TPM 2.0 structure definitions.";
      uses nonce;
      uses tpm20-pcr-selection;
      leaf-list certificate-name {
        if-feature "tpm:mtpm";
        type certificate-name-ref;
        must "/tpm:rats-support-structures/tpm:tpms"
           + "/tpm:tpm[tpm:firmware-version='taa:tpm20']"
           + "/tpm:certificates/"
           + "/tpm:certificate[name=current()]" {
          error-message "Not an available TPM2.0 AIK certificate.";
        }
        description
          "When populated, the RPC will only get a Quote for the
           TPMs associated with the certificates.";
      }
```

```
      }
    }
    output {
      list tpm20-attestation-response {
        unique "certificate-name";
        description
          "The binary output of TPM2_Quote from one TPM of the
           node which is identified by node-id:  an attestation
           structure (TPMS_ATTEST), including a length, and a
           signature (TPMT_SIGNATURE) over that structure.";
        reference
          "TPM2.0-Structures:
           Trusted Platform Module Library Part 2:  Structures,
           Revision 01.83, https://trustedcomputinggroup.org/
           wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf,
           Section 10.12.12";
        uses certificate-name-ref {
          description
            "Certificate associated with this tpm20-attestation.";
        }
        uses tpm20-attestation;
      }
    }
  }

  rpc log-retrieval {
    description
      "Log entries are identified either via indices or by providing
       the last line received.  The number of lines returned can be
       limited.  The type of log is a choice that can be augmented.";
    input {
      uses log-identifier;
      list log-selector {
        description
          "Only log entries that meet all of the provided selection
           criteria are to be returned by the RPC output.";
        leaf-list name {
          type string;
          description
            "Name of one or more unique TPMs on a device.  If this
             object exists, a selection should pull only the objects
             related to these TPM(s).  If it does not exist, all
             qualifying TPMs that are 'hardware-based' equals true
             on the device are selected.  When this selection
             criteria is provided, it will be considered as a logical
             AND with any other selection criteria provided.";
        }
        choice index-type {
          description
            "Last log entry received, log index number, or
             timestamp.";
          case last-entry {
            description
              "The last entry of the log already retrieved.";
            leaf last-entry-value {
              type binary;
              description
                "Content of a log event that matches 1:1 with a
```

```
                unique event record contained within the log.  Log
                entries after this will be passed to the
                requester.  Note: if log entry values are not
                unique, this MUST return an error.";
            }
          }
          case index {
            description
              "Numeric index of the last log entry retrieved, or
               zero.";
            leaf last-index-number {
              type uint64;
              description
                "The last numeric index number of a log entry.
                 Zero means to start at the beginning of the log.
                 Entries after this will be passed to the
                 requester.";
            }
          }
          case timestamp {
            leaf timestamp {
              type yang:date-and-time;
              description
                "Timestamp from which to start the extraction.  The
                 next log entry after this timestamp is to
                 be sent.";
            }
            description
              "Timestamp from which to start the extraction.";
          }
        }
        leaf log-entry-quantity {
          type uint16;
          description
            "The number of log entries to be returned. If omitted, it
             means all of them.";
        }
      }
    }
    output {
      container system-event-logs {
        description
          "The requested data of the measurement event logs.";
        list node-data {
          unique "name";
          description
            "Event logs of a node in a distributed system
             identified by the node name.";
          uses tpm-name;
          uses node-uptime;
          container log-result {
            description
              "The requested entries of the corresponding log.";
            uses event-logs;
          }
        }
      }
    }
```

```
  }

  /***************************************/
  /*   Config and Oper accessible nodes   */
  /***************************************/

  container rats-support-structures {
    description
      "The datastore definition enabling Verifiers or Relying
       Parties to discover the information necessary to use the
       remote attestation RPCs appropriately.";
    container compute-nodes {
      if-feature "tpm:mtpm";
      description
        "Holds the set of device subsystems/components in this
         composite device that support TPM operations.";
      list compute-node {
        key "node-id";
        unique "node-name";
        config false;
        min-elements 2;
        description
          "A component within this composite device that
           supports TPM operations.";
        leaf node-id {
          type string;
          description
            "ID of the compute node, such as Board Serial Number.";
        }
        leaf node-physical-index {
          if-feature "hw:entity-mib";
          type int32 {
            range "1..2147483647";
          }
          config false;
          description
            "The entPhysicalIndex for the compute node.";
          reference
            "RFC 6933: Entity MIB (Version 4) - entPhysicalIndex";
        }
        leaf node-name {
          type string;
          description
            "Name of the compute node.";
        }
        leaf node-location {
          type string;
          description
            "Location of the compute node, such as slot number.";
        }
      }
    }
    container tpms {
      description
        "Holds the set of TPMs within an Attester.";
      list tpm {
        key "name";
        unique "path";
```

```
        description
          "A list of TPMs in this composite device that RATS
           can be conducted with.";
        uses tpm-name;
        leaf hardware-based {
          type boolean;
          config false;
          mandatory true;
          description
            "System-generated indication of whether this is a
             hardware-based TPM.";
        }
        leaf physical-index {
          if-feature "hw:entity-mib";
          type int32 {
            range "1..2147483647";
          }
          config false;
          description
            "The entPhysicalIndex for the TPM.";
          reference
            "RFC 6933: Entity MIB (Version 4) - entPhysicalIndex";
        }
        leaf path {
          type string;
          config false;
          description
            "Device path to a unique TPM on a device.  This can
             change across reboots.";
        }
        leaf compute-node {
          if-feature "tpm:mtpm";
          type compute-node-ref;
          config false;
          mandatory true;
          description
            "Indicates the compute node measured by this TPM.";
        }
        leaf manufacturer {
          type string;
          config false;
          description
            "TPM manufacturer name.";
        }
        leaf firmware-version {
          type identityref {
            base taa:cryptoprocessor;
          }
          mandatory true;
          description
            "Identifies the cryptoprocessor API set supported.  This
             is automatically configured by the device and should not
             be changed.";
        }
        uses tpm12-hash-algo {
          when "derived-from-or-self(firmware-version, 'taa:tpm12')";
          if-feature "taa:tpm12";
          refine "tpm12-hash-algo" {
```

```
            description
              "The hash algorithm overwrites the default used for
               PCRs on this TPM1.2-compliant cryptoprocessor.";
          }
        }
        leaf-list tpm12-pcrs {
          when "derived-from-or-self(../firmware-version, "
             + "'taa:tpm12')";
          if-feature "taa:tpm12";
          type pcr;
          description
            "The PCRs that may be extracted from this TPM1.2-
             compliant cryptoprocessor.";
        }
        list tpm20-pcr-bank {
          when "derived-from-or-self(../firmware-version, "
             + "'taa:tpm20')";
          if-feature "taa:tpm20";
          key "tpm20-hash-algo";
          description
            "Specifies the list of PCRs that may be extracted for
             a specific hash algorithm on this TPM2-compliant
             cryptoprocessor.  A bank is a set of PCRs that are
             extended using a particular hash algorithm.";
          reference
            "TPM2.0-Structures:
             Trusted Platform Module Library Part 2:  Structures,
             Revision 01.83, https://trustedcomputinggroup.org/
             wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf,
             Section 10.9.7";
          leaf tpm20-hash-algo {
            type identityref {
              base taa:hash;
            }
            must '/tpm:rats-support-structures'
               + '/tpm:attester-supported-algos'
               + '/tpm:tpm20-hash' {
              error-message "This platform does not support "
                          + "tpm20-hash-algo";
            }
            description
              "The hash scheme actively being used to hash
               one or more TPM2.0 PCRs.";
          }
          leaf-list pcr-index {
            type tpm:pcr;
            description
              "Defines which TPM2.0 PCRs are available to be
               extracted.";
          }
        }
        leaf status {
          type enumeration {
            enum operational {
              value 0;
              description
                "The TPM currently is running normally and
                 is ready to accept and process TPM quotes.";
```

```
            reference
              "TPM2.0-Arch: Trusted Platform Module Library
               Part 1:  Architecture,
               https://trustedcomputinggroup.org/wp-content/
               uploads/TPM-2.0-1.83-Part-1-Architecture.pdf,
               Section 12";
          }
          enum non-operational {
            value 1;
            description
              "TPM is in a state such as startup or shutdown, which
               precludes the processing of TPM quotes.";
          }
        }
        config false;
        mandatory true;
        description
          "TPM chip self-test status.";
      }
      container certificates {
        description
          "The TPM's certificates, including EK Certificates
           and Attestation Key Certificates.";
        list certificate {
          key "name";
          description
            "Three types of certificates can be accessed via
             this statement, including Initial Attestation
             Key Certificate, Local Attestation Key Certificate, or
             Endorsement Key Certificate.";
          leaf name {
            type string;
            description
              "An arbitrary name uniquely identifying a certificate
               associated with a key within a TPM.";
          }
          leaf keystore-ref {
            if-feature "ks:central-keystore-supported";
            if-feature "ks:asymmetric-keys";
            type leafref {
              path "/ks:keystore/ks:asymmetric-keys"
                 + "/ks:asymmetric-key/ks:name";
            }
            description
              "A reference to a specific certificate of an
               asymmetric key in the keystore.";
          }
          leaf type {
            type enumeration {
              enum endorsement-certificate {
                value 0;
                description
                  "Endorsement Key (EK) Certificate type.";
                reference
                  "TPM2.0-Key:
                   TPM 2.0 Keys for Device Identity and Attestation
                   https://trustedcomputinggroup.org/wp-content/
                   uploads/TPM-2p0-Keys-for-Device-Identity-
```

```
                    and-Attestation_v1_r12_pub10082021.pdf,
                    Section 3.11";
                }
                enum initial-attestation-certificate {
                  value 1;
                  description
                    "Initial Attestation Key (IAK) Certificate
                     type.";
                  reference
                    "TPM2.0-Key:
                     TPM 2.0 Keys for Device Identity and Attestation
                     https://trustedcomputinggroup.org/wp-content/
                     uploads/TPM-2p0-Keys-for-Device-Identity-
                     and-Attestation_v1_r12_pub10082021.pdf,
                     Section 3.2";
                }
                enum local-attestation-certificate {
                  value 2;
                  description
                    "Local Attestation Key (LAK) Certificate type.";
                  reference
                    "TPM2.0-Key:
                     TPM 2.0 Keys for Device Identity and Attestation
                     https://trustedcomputinggroup.org/wp-content/
                     uploads/TPM-2p0-Keys-for-Device-Identity-
                     and-Attestation_v1_r12_pub10082021.pdf,
                     Section 3.2";
                }
              }
              description
                "Function supported by this certificate from within
                 the TPM.";
            }
          }
        }
      }
    }
    container attester-supported-algos {
      description
        "Identifies which TPM algorithms are available for use on an
         attesting platform.";
      leaf-list tpm12-asymmetric-signing {
        when "../../tpm:tpms"
           + "/tpm:tpm[tpm:firmware-version='taa:tpm12']";
        if-feature "taa:tpm12";
        type identityref {
          base taa:asymmetric;
        }
        description
          "Platform-supported TPM1.2 asymmetric algorithms.";
      }
      leaf-list tpm12-hash {
        when "../../tpm:tpms"
           + "/tpm:tpm[tpm:firmware-version='taa:tpm12']";
        if-feature "taa:tpm12";
        type identityref {
          base taa:hash;
        }
```

```
          description
            "Platform-supported TPM1.2 hash algorithms.";
        }
        leaf-list tpm20-asymmetric-signing {
          when "../../tpm:tpms"
             + "/tpm:tpm[tpm:firmware-version='taa:tpm20']";
          if-feature "taa:tpm20";
          type identityref {
            base taa:asymmetric;
          }
          description
            "Platform-supported TPM2.0 asymmetric algorithms.";
        }
        leaf-list tpm20-hash {
          when "../../tpm:tpms"
             + "/tpm:tpm[tpm:firmware-version='taa:tpm20']";
          if-feature "taa:tpm20";
          type identityref {
            base taa:hash;
          }
          description
            "Platform-supported TPM2.0 hash algorithms.";
        }
      }
    }
  }

<CODE ENDS>
```

Figure 1

2.1.2. ietf-tcg-algs

This document has encoded the TCG Algorithm definitions of Table 3 of [TCG-Algos], revision 1.32. By including this full table as a separate YANG file within this document, it is possible for other YANG modules to leverage the contents of this module. Specific references to [TPM1.2-Structures], [TPM2.0], [RFC2104], [RFC8017], [RFC8032], [ISO-IEC-9797-1], [ISO-IEC-9797-2], [ISO-IEC-10116], [ISO-IEC-10118-3], [ISO-IEC-14888-3], [ISO-IEC-15946-1], [ISO-IEC-18033-3], [IEEE-Std-1363-2000], [IEEE-Std-1363a-2004], [NIST-FIPS-202], [NIST-SP800-38C], [NIST-SP800-38D], [NIST-SP800-38F], [NIST-SP800-56A], and [NIST-SP800-108] exist within the YANG module.

2.1.2.1. Features

There are two types of features supported: 'tpm12' and 'tpm20'. Support for either of these features indicates that a cryptoprocessor supporting the corresponding type of TCG TPM API is present on an Attester. Most commonly, only one type of cryptoprocessor will be available on an Attester.

2.1.2.2. Identities

There are three types of identities in this model:

1. Cryptographic functions supported by a TPM algorithm; these include 'asymmetric', 'symmetric', 'hash', 'signing', 'anonymous_signing', 'encryption_mode', 'method', and 'object_type'. The definitions of each of these are in Table 2 of [TCG-Algos].
2. API specifications for TPM types: 'tpm12' and 'tpm20'
3. Specific algorithm types: Each algorithm type defines which cryptographic functions may be supported, and on which type of API specification. It is not required that an implementation of a specific TPM will support all algorithm types. The contents of each specific algorithm mirrors the contents of Table 3 of [TCG-Algos].

2.1.2.3. YANG Module

```
<CODE BEGINS> file "ietf-tcg-algs@2024-10-22.yang"

module ietf-tcg-algs {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tcg-algs";
  prefix taa;

  organization
    "IETF RATS (Remote ATtestation procedureS) Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/rats/>
     WG List:  <mailto:rats@ietf.org>
     Author:   Eric Voit <mailto:evoit@cisco.com>";
  description
    "This module defines identities for asymmetric algorithms.

     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 (RFC 2119) (RFC 8174) when, and only when,
     they appear in all capitals, as shown here.

     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 Revised 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 9684; see the
     RFC itself for full legal notices.";

  revision 2024-10-22 {
    description
      "Initial version";
    reference
```

```
      "RFC 9684: A YANG Data Model for Challenge-Response-Based
       Remote Attestation (CHARRA) Procedures Using Trusted Platform
       Modules (TPMs)";
  }

  /*****************/
  /*   Features    */
  /*****************/

  feature tpm12 {
    description
      "This feature indicates algorithm support for the TPM 1.2 API
       per Section 4.8 of TPM1.2-Structures.";
    reference
      "TPM1.2-Structures: TPM Main Part 2 TPM Structures,
       https://trustedcomputinggroup.org/wp-content/uploads/
       TPM-Main-Part-2-TPM-Structures_v1.2_rev116_01032011.pdf
       TPM_ALGORITHM_ID values, Section 4.8";
  }

  feature tpm20 {
    description
      "This feature indicates algorithm support for the TPM 2.0 API
       per Section 11.4 of Trusted Platform Module Library Part 1:
       Architecture.";
    reference
      "TPM2.0-Arch: Trusted Platform Module Library Part 1:
       Architecture, https://trustedcomputinggroup.org/wp-content/
       uploads/TPM-2.0-1.83-Part-1-Architecture.pdf, Section 11.4";
  }

  /*****************/
  /*  Identities   */
  /*****************/

  identity asymmetric {
    description
      "A TCG-recognized asymmetric algorithm with a public and
       private key.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2,
       https://trustedcomputinggroup.org/resource/
       tcg-algorithm-registry/TCG-_Algorithm_Registry_r1p32_pub";
  }

  identity symmetric {
    description
      "A TCG-recognized symmetric algorithm with only a private
       key.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2";
  }

  identity hash {
    description
      "A TCG-recognized hash algorithm that compresses input data to
       a digest value or indicates a method that uses a hash.";
    reference
```

```
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2";
  }

  identity signing {
    description
      "A TCG-recognized signing algorithm";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2";
  }

  identity anonymous_signing {
    description
      "A TCG-recognized anonymous signing algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2";
  }

  identity encryption_mode {
    description
      "A TCG-recognized encryption mode.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2";
  }

  identity method {
    description
      "A TCG-recognized method such as a mask generation function.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2";
  }

  identity object_type {
    description
      "A TCG-recognized object type.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 2";
  }

  identity cryptoprocessor {
    description
      "Base identity identifying a crytoprocessor.";
  }

  identity tpm12 {
    if-feature "tpm12";
    base cryptoprocessor;
    description
      "Supportable by a TPM 1.2.";
    reference
      "TPM1.2-Structures:
       TPM Main Part 2 TPM Structures,
       https://trustedcomputinggroup.org/wp-content/uploads/
       TPM-Main-Part-2-TPM-Structures_v1.2_rev116_01032011.pdf
       TPM_ALGORITHM_ID values, Section 4.8";
  }

  identity tpm20 {
    if-feature "tpm20";
```

```
    base cryptoprocessor;
    description
      "Supportable by a TPM 2.0";
    reference
      "TPM2.0-Structures:
       Trusted Platform Module Library Part 2:  Structures,
       Revision 01.83, https://trustedcomputinggroup.org/
       wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf";
  }

  identity TPM_ALG_RSA {
    if-feature "tpm12 or tpm20";
    base tpm12;
    base tpm20;
    base asymmetric;
    base object_type;
    description
      "RSA algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       RFC 8017. ALG_ID: 0x0001";
  }

  identity TPM_ALG_TDES {
    if-feature "tpm12";
    base tpm12;
    base symmetric;
    description
      "Block cipher with various key sizes (Triple Data Encryption
       Algorithm, commonly called Triple Data Encryption Standard)
       Note: Was banned in TPM 1.2, v94";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 18033-3. ALG_ID: 0x0003";
  }

  identity TPM_ALG_SHA1 {
    if-feature "tpm12 or tpm20";
    base hash;
    base tpm12;
    base tpm20;
    description
      "SHA1 algorithm - Deprecated due to insufficient cryptographic
       protection.  However, it is still useful for hash algorithms
       where protection is not required.";
    reference
      "TCG-Algos: TCG Algorithm Registry Rev1.34, Table 3, and
       ISO/IEC 10118-3. ALG_ID: 0x0004";
  }

  identity TPM_ALG_HMAC {
    if-feature "tpm12 or tpm20";
    base tpm12;
    base tpm20;
    base hash;
    base signing;
    description
      "Hash Message Authentication Code (HMAC) algorithm.";
```

```
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 9797-2, and
       RFC 2104. ALG_ID: 0x0005";
  }

  identity TPM_ALG_AES {
    if-feature "tpm12";
    base tpm12;
    base symmetric;
    description
      "The AES algorithm with various key sizes.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 18033-3. ALG_ID: 0x0006";
  }

  identity TPM_ALG_MGF1 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    base method;
    description
      "Hash-based mask-generation function.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       IEEE Std 1363-2000, and
       IEEE Std 1363a-2004.
       ALG_ID: 0x0007";
  }

  identity TPM_ALG_KEYEDHASH {
    if-feature "tpm20";
    base tpm20;
    base hash;
    base object_type;
    description
      "An encryption or signing algorithm using a keyed hash.  These
       may use XOR for encryption or an HMAC for signing and may
       also refer to a data object that is neither signing nor
       encrypting.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3.
       ALG_ID: 0x0008";
  }

  identity TPM_ALG_XOR {
    if-feature "tpm12 or tpm20";
    base tpm12;
    base tpm20;
    base hash;
    base symmetric;
    description
      "The XOR encryption algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3.
       ALG_ID: 0x000A";
  }
```

```
  identity TPM_ALG_SHA256 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    description
      "The SHA-256 algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10118-3. ALG_ID: 0x000B";
  }

  identity TPM_ALG_SHA384 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    description
      "The SHA-384 algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10118-3. ALG_ID: 0x000C";
  }

  identity TPM_ALG_SHA512 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    description
      "The SHA-512 algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10118-3. ALG_ID: 0x000D";
  }

  identity TPM_ALG_NULL {
    if-feature "tpm20";
    base tpm20;
    description
      "Null algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3.
       ALG_ID: 0x0010";
  }

  identity TPM_ALG_SM3_256 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    description
      "The ShangMi 3 (SM3) hash algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10118-3:2018. ALG_ID: 0x0012";
  }

  identity TPM_ALG_SM4 {
    if-feature "tpm20";
    base tpm20;
```

```
    base symmetric;
    description
      "ShangMi 4 (SM4) symmetric block cipher.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3.
       ALG_ID: 0x0013";
  }

  identity TPM_ALG_RSASSA {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base signing;
    description
      "Signature algorithm defined in Section 8.2
       (RSASSA-PKCS1-v1_5) of RFC 8017.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       RFC 8017.  ALG_ID: 0x0014";
  }

  identity TPM_ALG_RSAES {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base encryption_mode;
    description
      "Signature algorithm defined in Section 7.2
       (RSAES-PKCS1-v1_5) of RFC 8017.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       RFC 8017. ALG_ID: 0x0015";
  }

  identity TPM_ALG_RSAPSS {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base signing;
    description
      "Padding algorithm defined in Section 8.1 (RSASSA-PSS)
       of RFC 8017.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       RFC 8017. ALG_ID: 0x0016";
  }

  identity TPM_ALG_OAEP {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base encryption_mode;
    description
      "Padding algorithm defined in Section 7.1 (RSAES-OAEP)
       of RFC 8017.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       RFC 8017. ALG_ID: 0x0017";
```

```
  }

  identity TPM_ALG_ECDSA {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base signing;
    description
      "Signature algorithm using elliptic curve cryptography (ECC).";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 14888-3. ALG_ID: 0x0018";
  }

  identity TPM_ALG_ECDH {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base method;
    description
      "Secret sharing using ECC.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-56A. ALG_ID: 0x0019";
  }

  identity TPM_ALG_ECDAA {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base signing;
    base anonymous_signing;
    description
      "Elliptic-curve-based, anonymous signing scheme.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       TCG TPM 2.0 Library. ALG_ID: 0x001A";
  }

  identity TPM_ALG_SM2 {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base signing;
    base encryption_mode;
    base method;
    description
      "SM2 - depending on context, either an elliptic-curve based,
       signature algorithm, an encryption scheme, or a key exchange
       protocol.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3.
       ALG_ID: 0x001B";
  }

  identity TPM_ALG_ECSCHNORR {
    if-feature "tpm20";
    base tpm20;
```

```
    base asymmetric;
    base signing;
    description
      "Elliptic-curve-based Schnorr signature.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3.
       ALG_ID: 0x001C";
  }

  identity TPM_ALG_ECMQV {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base method;
    description
      "Two-phase elliptic-curve key.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-56A. ALG_ID: 0x001D";
  }

  identity TPM_ALG_KDF1_SP800_56A {
    if-feature "tpm20";
    base tpm20;
    base hash;
    base method;
    description
      "Concatenation key derivation function.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-56A  (approved alternative1) Section 5.8.1.
       ALG_ID: 0x0020";
  }

  identity TPM_ALG_KDF2 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    base method;
    description
      "Key derivation function.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       IEEE 1363a-2004, KDF2, Section 13.2. ALG_ID: 0x0021";
  }

  identity TPM_ALG_KDF1_SP800_108 {
    base TPM_ALG_KDF2;
    description
      "A key derivation method.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3 and
       NIST SP800-108, Section 4.1, KDF. ALG_ID: 0x0022";
  }

  identity TPM_ALG_ECC {
    if-feature "tpm20";
    base tpm20;
```

```
    base asymmetric;
    base object_type;
    description
      "Prime field ECC.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 15946-1. ALG_ID: 0x0023";
  }

  identity TPM_ALG_SYMCIPHER {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base object_type;
    description
      "Object type for a symmetric block cipher.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       TCG TPM 2.0 Library. ALG_ID: 0x0025";
  }

  identity TPM_ALG_CAMELLIA {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    description
      "The Camellia algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 18033-3. ALG_ID: 0x0026";
  }

  identity TPM_ALG_SHA3_256 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    description
      "ISO/IEC 10118-3 - the SHA-256 algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST FIPS 202. ALG_ID: 0x0027";
  }

  identity TPM_ALG_SHA3_384 {
    if-feature "tpm20";
    base tpm20;
    base hash;
    description
      "The SHA-384 algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST FIPS 202. ALG_ID: 0x0028";
  }

  identity TPM_ALG_SHA3_512 {
    if-feature "tpm20";
    base tpm20;
    base hash;
```

```
    description
      "The SHA-512 algorithm.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST FIPS 202. ALG_ID: 0x0029";
  }

  identity TPM_ALG_CMAC {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base signing;
    description
      "Block Cipher-based Message Authentication Code (CMAC).";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 9797-1:2011, Algorithm 5. ALG_ID: 0x003F";
  }

  identity TPM_ALG_CTR {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base encryption_mode;
    description
      "Counter mode.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10116. ALG_ID: 0x0040";
  }

  identity TPM_ALG_OFB {
    base tpm20;
    base symmetric;
    base encryption_mode;
    description
      "Output Feedback mode.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10116. ALG_ID: 0x0041";
  }

  identity TPM_ALG_CBC {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base encryption_mode;
    description
      "Cipher Block Chaining mode.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10116. ALG_ID: 0x0042";
  }

  identity TPM_ALG_CFB {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
```

```
    base encryption_mode;
    description
      "Cipher Feedback mode.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10116. ALG_ID: 0x0043";
  }

  identity TPM_ALG_ECB {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base encryption_mode;
    description
      "Electronic Codebook mode.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       ISO/IEC 10116. ALG_ID: 0x0044";
  }

  identity TPM_ALG_CCM {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base signing;
    base encryption_mode;
    description
      "Counter with Cipher Block Chaining--Message Authentication
       Code (CCM).";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-38C. ALG_ID: 0x0050";
  }

  identity TPM_ALG_GCM {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base signing;
    base encryption_mode;
    description
      "Galois/Counter Mode (GCM).";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-38D. ALG_ID: 0x0051";
  }

  identity TPM_ALG_KW {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base signing;
    base encryption_mode;
    description
      "AES Key Wrap (KW).";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-38F. ALG_ID: 0x0052";
```

```
  }

  identity TPM_ALG_KWP {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base signing;
    base encryption_mode;
    description
      "AES Key Wrap with Padding (KWP).";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-38F. ALG_ID: 0x0053";
  }

  identity TPM_ALG_EAX {
    if-feature "tpm20";
    base tpm20;
    base symmetric;
    base signing;
    base encryption_mode;
    description
      "Authenticated-Encryption Mode.";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       NIST SP800-38F. ALG_ID: 0x0054";
  }

  identity TPM_ALG_EDDSA {
    if-feature "tpm20";
    base tpm20;
    base asymmetric;
    base signing;
    description
      "Edwards-curve Digital Signature Algorithm (PureEdDSA).";
    reference
      "TCG-Algos: TCG Algorithm Registry, Rev1.34, Table 3, and
       RFC 8032. ALG_ID: 0x0060";
  }
}

<CODE ENDS>
```

Note that not all cryptographic functions are required for use by ietf-tpm-remote-attestation.yang. However, the full definition of Table 3 of [TCG-Algos] will allow use by additional YANG specifications.

3. IANA Considerations

This document registers the following namespace URIs in the [XML-Registry] per [RFC3688]:

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

Registrant Contact: The IESG.

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

URI: urn:ietf:params:xml:ns:yang:ietf-tcg-algs

Registrant Contact: The IESG.

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

This document registers the following YANG modules in the registry [YANG-Parameters] per Section 14 of [RFC6020]:

Name: ietf-tpm-remote-attestation

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

Prefix: tpm

Reference: draft-ietf-rats-yang-tpm-charra (RFC form)

Name: ietf-tcg-algs

Namespace: urn:ietf:params:xml:ns:yang:ietf-tcg-algs

Prefix: taa

Reference: draft-ietf-rats-yang-tpm-charra (RFC form)

4. Security Considerations

The YANG module ietf-tpm-remote-attestation.yang specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

Of special consideration are the following nodes:

- In the 'tpms' container, the 'certificates' will expose certificates used for attestation, potentially allowing selection of a certificate that might be compromised. The 'type' could also be misconfigured to represent a different type of key, which might alter how a Verifier might evaluate the results.
- Within the 'attester-supported-algos' container, each leaf-list will expose and potentially allow changing of the encryption algorithms supported by a device.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., *config true*, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., *edit-config*) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes as well as their sensitivity/vulnerability:

Container '/rats-support-structures/attester-supported-algos': 'tpm12-asymmetric-signing', 'tpm12-hash', 'tpm20-asymmetric-signing', and 'tpm20-hash'. All could be populated with algorithms that are not supported by the underlying physical TPM installed by the equipment vendor. A vendor should restrict the ability to configure unsupported algorithms.

Container: '/rats-support-structures/tpms': 'name': Although shown as 'rw', it is system generated. Therefore, it should not be possible for an operator to add or remove a TPM from the configuration.

'tpm20-pcr-bank': It is possible to configure PCRs that are not being extended by system software for extraction. This could unnecessarily use TPM resources.

'certificates': It is possible to provision a certificate that does not correspond to an AIK within the TPM 1.2, or to an Attestation Key (AK) within the TPM 2.0, respectively. In such a case, calls to an RPC requesting this specific certificate could result in either no response or a response from an unexpected TPM.

RPC 'tpm12-challenge-response-attestation': The receiver of the RPC response must verify that the certificate is for an active AIK, i.e., the certificate has been confirmed by a third party as being able to support Attestation on the targeted TPM 1.2.

RPC 'tpm20-challenge-response-attestation': The receiver of the RPC response must verify that the certificate is for an active AK, i.e., the private key confirmation of the quote signature within the RPC response has been confirmed by a third party to belong to an entity legitimately able to perform Attestation on the targeted TPM 2.0.

RPC 'log-retrieval': Requesting a large volume of logs from the Attester could require significant system resources and create a denial of service.

Information collected through the RPCs above could reveal specific versions of software and configurations of endpoints that could identify vulnerabilities on those systems. Therefore, RPCs should be protected by NACM [RFC8341] with a default setting of deny-all to limit the extraction of attestation data by only authorized Verifiers.

For the YANG module ietf-tcg-algs.yang, please use care when selecting specific algorithms. The introductory section of [TCG-Algos] highlights that some algorithms should be considered legacy, and recommends implementers and adopters diligently evaluate available information such as governmental, industrial, and academic research before selecting an algorithm for use.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

Event logs (bios-log, ima-log, netequip-boot-log) typically contain hash values (digests) of running boot and OS software. Passive attackers can use these hash values to identify software versions and thus launch targeted attacks on known vulnerabilities. Hence, bios-log, ima-log, and netequip-boot-log are considerd sensitive.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

The 'log-retrieval' RPC operation is considered sensitive since it enables retrieval of logs (bios-log, ima-log, netequip-boot-log) that typically contain hash values (digests) of running boot and OS software. This allows specifics of loaded software including BIOS and operating system software to be understood externally.

The other two RPC operations, 'tpm20-challenge-response-attestation' and 'tpm12-challenge-response-attestation', will expose values indicating the internal operational state of the device. These values could also be correlated to specifics of running software as well.

5. References

5.1. Normative References

[BIOS-Log] Trusted Computing Group, "TCG PC Client Platform Firmware Profile Specification", Family "2.0" Level 00 Revision 1.03 Version 51, 1 May 2017, <https://trustedcomputinggroup.org/wp-content/uploads/PC-ClientSpecific_Platform_Profile_for_TPM_2p0_Systems_v51.pdf>.

[CEL] Trusted Computing Group, "Canonical Event Log Format", Version 1.0 Revision 0.41, 25 February 2022, <https://trustedcomputinggroup.org/wp-content/uploads/TCG_IWG_CEL_v1_r0p41_pub.pdf>.

[IEEE-Std-1363-2000] IEEE, "IEEE Standard Specifications for Public-Key Cryptography", IEEE Std 1363-2000, DOI 10.1109/IEEESTD.2000.92292, August 2000, <https://ieeexplore.ieee.org/document/891000>.

[IEEE-Std-1363a-2004] IEEE, "IEEE Standard Specifications for Public-Key Cryptography - Amendment 1: Additional Techniques", IEEE Std 1363a-2004, DOI 10.1109/IEEESTD.2004.94612, September 2004, <https://ieeexplore.ieee.org/document/1335427>.

[ISO-IEC-10116] ISO/IEC, "Information technology - Security techniques - Modes of operation for an n-bit block cipher", Edition 4, ISO/IEC 10116:2017, July 2017, <https://www.iso.org/standard/64575.html>.

[ISO-IEC-10118-3] ISO/IEC, "IT Security techniques - Hash-functions - Part 3: Dedicated hash-functions", Edition 4, ISO/IEC 10118-3:2018, October 2018, <https://www.iso.org/standard/67116.html>.

[ISO-IEC-14888-3] ISO/IEC, "Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Edition 4, ISO/IEC 14888-3:2018, November 2018, <https://www.iso.org/standard/76382.html>.

[ISO-IEC-15946-1] ISO/IEC, "Information technology - Security techniques - Cryptographic techniques based on elliptic curves - Part 1: General", Edition 3, ISO/IEC 15946-1:2016, July 2016, <https://www.iso.org/standard/65480.html>.

[ISO-IEC-18033-3] ISO/IEC, "Information technology - Security techniques - Encryption algorithms - Part 3: Block ciphers", Edition 2, ISO/IEC 18033-3:2010, December 2010, <https://www.iso.org/standard/54531.html>.

[ISO-IEC-9797-1] ISO/IEC, "Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher", Edition 2, ISO/IEC 9797-1:2011, November 2011, <https://www.iso.org/standard/50375.html>.

[ISO-IEC-9797-2] ISO/IEC, "Information security - Message authentication codes (MACs) - Part 2: Mechanisms using a dedicated hash-function", Edition 3, ISO/IEC 9797-2:2021, June 2021, <https://www.iso.org/standard/75296.html>.

[NIST-FIPS-202] NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", NIST FIPS 202, DOI 10.6028/NIST.FIPS.202, August 2015, <https://csrc.nist.gov/publications/detail/fips/202/final>.

[NIST-SP800-108] Chen, L., "Recommendation for Key Derivation Using Pseudorandom Functions", DOI 10.6028/NIST.SP.800-108r1-upd1, NIST SP 800-108r1-upd1, February 2024, <https://csrc.nist.gov/pubs/sp/800/108/r1/upd1/final>.

[NIST-SP800-38C] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality", NIST SP 800-38C, DOI 10.6028/NIST.SP.800-38C, July 2007, <https://csrc.nist.gov/publications/detail/sp/800-38c/final>.

[NIST-SP800-38D] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC", NIST SP 800-38D, DOI 10.6028/NIST.SP.800-38D, November 2007, <https://csrc.nist.gov/publications/detail/sp/800-38d/final>.

[NIST-SP800-38F] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping", NIST SP 800-38F, DOI 10.6028/NIST.SP.800-38F, December 2012, <https://csrc.nist.gov/publications/detail/sp/800-38f/final>.

[NIST-SP800-56A] Barker, E., Chen, L., Roginsky, A., Vassilev, A., and R. Davis, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography", NIST SP 800-56A Rev. 3, DOI 10.6028/NIST.SP.800-56Ar3, April 2018, <https://csrc.nist.gov/publications/detail/sp/800-56a/rev-3/final>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[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/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/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/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6933] Bierman, A., Romascanu, D., Quittek, J., and M. Chandramouli, "Entity MIB (Version 4)", RFC 6933, DOI 10.17487/RFC6933, May 2013, <https://www.rfc-editor.org/info/rfc6933>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[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/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8348] Bierman, A., Bjorklund, M., Dong, J., and D. Romascanu, "A YANG Data Model for Hardware Management", RFC 8348, DOI 10.17487/RFC8348, March 2018, <https://www.rfc-editor.org/info/rfc8348>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[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/info/rfc9334>.

[RFC9642] Watsen, K., "A YANG Data Model for a Keystore", RFC 9642, DOI 10.17487/RFC9642, October 2024, <https://www.rfc-editor.org/info/rfc9642>.

[RFC9683] Fedorkow, G. C., Voit, E., and J. Fitzgerald-McKay, "Remote Integrity Verification of Network Devices Containing Trusted Platform Modules", RFC 9683, DOI 10.17487/RFC9683, November 2024, <https://www.rfc-editor.org/info/rfc9683>.

[TCG-Algos] Trusted Computing Group, "TCG Algorithm Registry", Family "2.0" Level 00 Revision 01.34, 24 August 2023, <https://trustedcomputinggroup.org/wp-content/uploads/TCG-Algorithm-Registry-Revision-1.34_pub-1.pdf>.

[TPM1.2] Trusted Computing Group, "TPM 1.2 Main Specification", TPM Main Specification Level 2 Version 1.2, Revision 116, 1 March 2011, <https://trustedcomputinggroup.org/resource/tpm-main-specification/>.

[TPM1.2-Commands] Trusted Computing Group, "TPM Main Part 3 Commands", TPM Main Specification Level 2 Version 1.2, Revision 116, 1 March 2011, <https://trustedcomputinggroup.org/wp-content/uploads/TPM-Main-Part-3-Commands_v1.2_rev116_01032011.pdf>.

[TPM1.2-Structures] Trusted Computing Group, "TPM Main Part 2 TPM Structures", TPM Main Specification Level 2 Version 1.2, Revision 116, 1 March 2011, <https://trustedcomputinggroup.org/wp-content/uploads/TPM-Main-Part-2-TPM-Structures_v1.2_rev116_01032011.pdf>.

[TPM2.0] Trusted Computing Group, "TPM 2.0 Library", Trusted Platform Module Library Specification, Family "2.0", Level 00, Revision 01.83, March 2024, <https://trustedcomputinggroup.org/resource/tpm-library-specification/>.

[TPM2.0-Arch] Trusted Computing Group, "Trusted Platform Module Library Part 1: Architecture", Family "2.0", Level 00, Revision 01.83, 25 January 2024, <https://trustedcomputinggroup.org/wp-content/uploads/TPM-2.0-1.83-Part-1-Architecture.pdf>.

[TPM2.0-Key] Trusted Computing Group, "TPM 2.0 Keys for Device Identity and Attestation", Version 1.00, Revision 12, 8 October 2021, <https://trustedcomputinggroup.org/wp-content/uploads/TPM-2p0-Keys-for-Device-Identity-and-Attestation_v1_r12_pub10082021.pdf>.

[TPM2.0-Structures] Trusted Computing Group, "Trusted Platform Module Library Part 2: Structures", Family "2.0", Level 00, Revision 01.83, 25 January 2024, <https://trustedcomputinggroup.org/wp-content/uploads/TPM-2.0-1.83-Part-2-Structures.pdf>.

[UEFI-Secure-Boot] Unified Extensible Firmware Interface (UEFI) Forum, Inc., "Unified Extensible Firmware Interface (UEFI) Specification", Section 32.1: Secure Boot, Version 2.10, 29 August 2022, <https://uefi.org/sites/default/files/resources/UEFI_Spec_2_10_Aug29.pdf>.

5.2. Informative References

[IMA-Template-Management] The kernel development community, "IMA Template Management Mechanism", Linux Kernel 6.11, 15 September 2024, <https://www.kernel.org/doc/html/v6.11/security/IMA-templates.html>.

[NIST-915121] NIST, "True Randomness Can't be Left to Chance: Why entropy is important for information security", <https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=915121>.

[RATS-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-11, 22 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-reference-interaction-models-11>.

[XML-Registry] IANA, "IETF XML Registry", <https://www.iana.org/assignments/xml-registry/>.

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

Appendix A. Integrity Measurement Architecture (IMA)

IMA extends the principles of Measured Boot [TPM2.0-Arch] and Secure Boot [UEFI-Secure-Boot] to the Linux operating system, applying it to operating system applications and files. IMA has been part of the Linux integrity subsystem of the Linux kernel since 2009 (kernel version 2.6.30). The IMA mechanism represented by the YANG module in this specification is rooted in the kernel version 5.16 [IMA-Template-Management]. IMA enables the protection of system integrity by collecting (commonly referred to as measuring) and storing measurements (called Claims in the context of IETF RATS) of files before execution so that these measurements can be used later, at system runtime, in remote attestation procedures. IMA acts in support of the Appraisal of Evidence (which includes measurement Claims) by leveraging Reference Values stored in extended file attributes.

In support of the Appraisal of Evidence, IMA maintains an ordered list (with no duplicates) of measurements in kernel space, the Stored Measurement Log (SML), for all files that have been measured before execution since the operating system was started. Although IMA can be used without a TPM, it is typically used in conjunction with a TPM to anchor the integrity of the SML

in a hardware-protected secure storage location, i.e., PCRs provided by TPMs. IMA provides the SML in both binary and ASCII representations in the Linux security file system *securityfs* (`/sys/kernel/security/ima/`).

IMA templates define the format of the SML, i.e., which fields are included in a log record. Examples are file path, file hash, user ID, group ID, file signature, and extended file attributes. IMA comes with a set of predefined template formats and also allows a custom format, i.e., a format consisting of template fields supported by IMA. Template usage is typically determined by boot arguments passed to the kernel. Alternatively, the format can also be hard-coded into custom kernels. IMA templates and fields are extensible in the kernel source code. As a result, more template fields can be added in the future.

IMA policies define which files are measured using the IMA policy language. Built-in policies can be passed as boot arguments to the kernel. Custom IMA policies can be defined once during runtime or be hard-coded into a custom kernel. If no policy is defined, no measurements are taken and IMA is effectively disabled.

A comprehensive description of the content fields of the Linux IMA TLV format can be found in Table 16 of the Canonical Event Log (CEL) specification [CEL]. The CEL specification also illustrates the use of templates to enable extended or customized IMA TLV formats in Section 5.1.6.

Appendix B. IMA for Network Equipment Boot Logs

Network equipment can generally implement similar IMA-protected functions to generate measurements (Claims) about the boot process of a device and enable corresponding remote attestation. Network Equipment Boot Logs combine the measurement and logging of boot components and operating system components (executables and files) into a single log file in a format identical to the IMA format. Note that the format used for logging measurement of boot components in this scheme differs from the boot logging strategy described elsewhere in this document.

During the boot process of the network device, i.e., from BIOS to the end of the operating system and user-space, all files executed can be measured and logged in the order of their execution. When the Verifier initiates a remote attestation process (e.g., challenge-response remote attestation as defined in this document), the network equipment takes on the role of an Attester and can convey to the Verifier Claims that comprise the measurement log as well as the corresponding PCR values (Evidence) of a TPM.

The Verifier can appraise the integrity (compliance with the Reference Values) of each executed file by comparing its measured value with the Reference Value. Based on the execution order, the Verifier can compute a PCR Reference Value (by replaying the log) and compare it to the measurement log Claims obtained in conjunction with the PCR Evidence to assess their trustworthiness with respect to an intended operational state.

Network equipment usually executes multiple components in parallel. This holds not only during the operating system loading phase, but also even during the BIOS boot phase. With this measurement log mechanism, network equipment can assume the role of an Attester, proving to the Verifier the trustworthiness of its boot process. Using the measurement log, Verifiers can precisely identify mismatching log entries to infer potentially tampered components.

This mechanism also supports scenarios that modify files on the Attester that are subsequently executed during the boot phase (e.g., updating/patching) by simply updating the appropriate Reference Values in Reference Integrity Manifests that inform Verifiers about how an Attester is composed.

Authors' Addresses

Henk Birkholz
Fraunhofer SIT | ATHENE Center
Rheinstrasse 75
64295 Darmstadt
Germany
Email: henk.birkholz@ietf.contact

Michael Eckel
Fraunhofer SIT | ATHENE Center
Rheinstrasse 75
64295 Darmstadt
Germany
Email: michael.eckel@sit.fraunhofer.de

Shwetha Bhandari
ThoughtSpot
Email: shwetha.bhandari@thoughtspot.com

Eric Voit
Cisco Systems
Email: evoit@cisco.com

Bill Sulzen
Cisco Systems
Email: bsulzen@cisco.com

Liang Xia (Frank)
Huawei Technologies
Yuhuatai District
101 Software Avenue
Nanjing
Jiangsu, 210012
China
Email: Frank.Xialiang@huawei.com

Tom Laffey
Hewlett Packard Enterprise
Email: tom.laffey@hpe.com

Guy C. Fedorkow
Juniper Networks
10 Technology Park Drive
Westford, Massachusetts 01886
United States of America
Email: gfedorkow@juniper.net