YANG Module for Basic Challenge-Response-based Remote Attestation Procedures
draft-birkholz-rats-basic-yang-module-00

Abstract

This document defines a YANG RPC and a minimal datastore tree required to retrieve attestation evidence about integrity measurements from a composite device with one or more roots of trust for reporting. Complementary measurement logs are also provided by the YANG RPC originating from one or more roots of trust of measurement. The module defined requires a TPM 2.0 and corresponding Trusted Software Stack included in the device components of the composite device the YANG server is running on.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 13, 2019.
1. Introduction

This document is based on the terminology defined in the [I-D.birkholz-attestation-terminology] and uses the interaction model and information elements defined in the [I-D.birkholz-reference-ra-interaction-model] document. The currently supported hardware security module (HWM) - sometimes also referred to as an embedded secure element (eSE) - is the Trusted Platform Module (TPM) 2.0 specified by the Trusted Computing Group (TCG). One or more TPM 2.0 embedded in the components of a composite device - sometimes also referred to as an aggregate device - are required in order to use the YANG module defined in this document. A TPM 2.0 is used as a root of trust for reporting (RTR) in order to retrieve attestation evidence from a composite device. Additionally, it is used as a root of trust for measurement (RTM) in order to provide event logs - sometimes also referred to as measurement logs.
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 RFC 2119, BCP 14 [RFC2119].

2. The YANG Module for Basic Remote Attestation Procedures

One or more TPM 2.0 MUST be embedded in the composite device that is providing attestation evidence via the YANG module defined in this document. The ietf-basic-remote-attestation YANG module enables a composite device to take on the role of Claimant and Attester in accordance with the Remote Attestation Procedures (RATS) architecture [I-D.birkholz-attestation-terminology] and the corresponding challenge-response interaction model defined in the [I-D.birkholz-reference-ra-interaction-model] document. A fresh nonce with an appropriate amount of entropy 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. The functions of this YANG module are restricted to 0-1 TPM 2.0 per hardware component.

2.1. Tree format

<CODE BEGINS>
module: ietf-basic-remote-attestation
  +--ro rats-support-structures
   |  +--ro supported-algos*         uint16
   |  +--ro tpms* [tpm_name]
   |     +--ro tpm_name              string
   |     |  +--ro tpm-physical-index?    int32 {ietfhw:entity-mib}?
   |  +--ro compute-nodes* [node-name]
   |     +--ro node-name              string
   |     |  +--ro node-physical-index?    int32 {ietfhw:entity-mib}?
   +--ro endorsement-certificates
     +--ro certificate* [tpm_name]
     |  +--ro tpm_name                string
     |  +--ro tpm-physical-index?      int32 {ietfhw:entity-mib}?
     +--ro endorsement-certificate   binary

rpcs:
  +----x tpm2-challenge-response-attestation
   |  +----w input
   |     |  +----w tpm2-attestation-challenge
   |     |     |  +----w pcr-list* []
   |     |     |     |  +----w pcr
   |     |     |     |     |  +----w pcr-indices*        uint8

| | |  | +---w (algo-registry-type)  
| | | | +---:(tcg)  
| | | | | +---w tcg-hash-algo-id? uint16  
| | | | | +---:(ietf)  
| | | | | | +---w ietf-ni-hash-algo-id? uint8  
| | | | +---w nonce-value binary  
| | | +---w (signature-identifier-type)  
| | | | | +---:(TPM_ALG_ID)  
| | | | | | +---w TPM_ALG_ID-value? uint16  
| | | | | | +---:(COSE_Algorithm)  
| | | | | | | +---w COSE_Algorithm-value? int32  
| | | | | | +---w (key-identifier)?  
| | | | | | | +---:(public-key)  
| | | | | | | | +---w pub-key-id? binary  
| | | | | | | | +---((uuid)  
| | | | | | | | | +---w uuid-value? binary  
| | | | | | +---w tpm_name? string  
| | | | +---w tpm-physical-index? int32 {ietfhw:entity-mib}?  
| | +---ro output  
| | | +---ro tpm2-attestation-response* [tpm_name]  
| | | | +---ro tpm_name string  
| | | | +---ro tpm-physical-index? int32 {ietfhw:entity-mib}?  
| | | | +---ro up-time? uint32  
| | | | +---ro node-name? string  
| | | +---ro node-physical-index? int32 {ietfhw:entity-mib}?  
| | +---ro tpms-attest  
| | | | +---ro pcrdigest? binary  
| | | | +---ro tpms-attest-result? binary  
| | | | +---ro tpms-attest-result-length? uint32  
| | | +---ro tpmt-signature? binary  
| +---x basic-trust-establishment  
| +---w input  
| | | +---w nonce-value binary  
| | +---w (signature-identifier-type)  
| | | | | +---:(TPM_ALG_ID)  
| | | | | | +---w TPM_ALG_ID-value? uint16  
| | | | | | +---:(COSE_Algorithm)  
| | | | | | | +---w COSE_Algorithm-value? int32  
| | | | | | +---w tpm_name? string  
| | | | +---w tpm-physical-index? int32 {ietfhw:entity-mib}?  
| | +---ro output  
| | | +---ro attestation-certificates* [tpm_name]  
| | | | +---ro tpm_name string  
| | | | +---ro tpm-physical-index? int32 {ietfhw:entity-mib}?  
| | | | +---ro up-time? uint32  
| | | | +---ro node-name? string  
| | | +---ro node-physical-index? int32 {ietfhw:entity-mib}?  

Internet-Draft          BRAT                        March 2019

+--ro certificate-name?                      string
+--ro attestation-certificate?              ietfct:end-entity-cert-cms
+--ro (key-identifier)?
   +--:(public-key)
   |   +--ro pub-key-id?          binary
   +--:(uuid)
   +--ro uuid-value?          binary
+---x log-retrieval
+---w input
   +---w log-selector* [node-name]
   |   +---w node-name                      string
   |   +---w node-physical-index?           int32 {ietfhw:entity-mib}?
   |   +--:(last-entry)
   |   |   +--w last-entry-value?          binary
   |   +--:(index)
   |   |   +--w index-number?               uint64
   |   +--:(timestamp)
   |   |   +--w timestamp?                   yang:date-and-time
   +---w log-type                         identityref
   +---w pcr-list* []
   |   +---w pcr
   |   |   +---w pcr-indices*               uint8
   |   |   +--:(algo-registry-type)
   |   |     +--:(tcg)
   |   |     |   +--w tcg-hash-algo-id?       uint16
   |   |     +--:(ietf)
   |   |     |   +--w ietf-ni-hash-algo-id?    uint8
   |   +--w log-entry-quantity?            uint16
+---w output
+--ro system-event-logs
+--ro node-data* [node-name]
   +--ro node-name                      string
   +--ro node-physical-index?           int32 {ietfhw:entity-mib}?
   +--ro up-time?                       uint32
   +--ro tpm-updated* [tpm_name]
   |   +--ro tpm_name                    string
   |   +--ro tpm-physical-index?          int32 {ietfhw:entity-mib}?
   +--ro log-result
   +--ro (log-type)
     +--:(bios)
     |   +--ro bios-event-logs
     |   |   +--ro bios-event-entry* [event-number]
     |   |   |   +--ro event-number               uint32
     |   |   |   +--ro event-type?                uint32
     |   |   |   +--ro pcr-index?                 uint16
     |   |   +--ro digest-list* []
     |   |     |   +--ro (algo-registry-type)
<CODE BEGINS>
module ietf-basic-remote-attestation {
  namespace "urn:ietf:params:xml:ns:yang:ietf-basic-remote-attestation";
  prefix "yang-brat";

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-hardware {
    prefix ietfhw;
  }
  import ietf-crypto-types {
    prefix ietfct;
  }

  organization "Fraunhofer SIT";
  contact "Henk Birkholz
    Fraunhofer Institute for Secure Information Technology
    Email: henk.birkholz@sit.fraunhofer.de";
  description "A YANG module to enable a TPM 2.0 based remote attestation procedure.
    Copyright (C) Fraunhofer SIT (2018).";
  revision "2018-06-15" {

<CODE ENDS>
grouping hash-algo {
    description "A selector for the hashing algorithm";
    choice algo-registry-type {
        mandatory true;
        description "Unfortunately, both IETF and TCG have registries here. Choose your weapon wisely.";
        case tcg {
            description "you chose the east door, the tcg space opens up to you.";
            leaf tcg-hash-algo-id {
                type uint16;
                description "This is an index referencing the TCG Algorithm Registry based on TPM_ALG_ID.";
            }
        }
        case ietf {
            description "you chose the west door, the ietf space opens up to you.";
            leaf ietf-ni-hash-algo-id {
                type uint8;
                description "This is an index referencing the Named Information Hash Algorithm Registry.";
            }
        }
    }
}

grouping hash {
    description "The hash value including hash-algo identifier";
    list hash-digests {
        description "The list of hashes.";
        container hash-digest {
            description "A hash value based on a hash algorithm registered by an
uses hash-algo;
leaf hash-value {
  type binary;
  description
  "The binary representaion of the hash value.";
}
}
}
}

grouping nonce {
  description
  "A nonce to show freshness and counter replays.";
  leaf nonce-value {
    type binary;
    mandatory true;
    description
    "This nonce SHOULD be generated via a registered
    cryptographic-strength algorithm. In consequence, the length
    of the nonce depends on the hash algorithm used. The algorithm
    used in this case is independent from the hash algorithm used to
    create the hash-value in the response of the attestor.";
  }
}

grouping pcr-selection {
  description
  "A Verifier can request one or more PCR values uses its
  individually created AC. The corresponding selection filter is
  represented in this grouping. Requesting a PCR value that is not in
  scope of the AC used, detailed exposure via error msg should be avoided.";
  list pcr-list {
    description
    "For each PCR in this list an individual list of banks (hash-algo)
    can be requested. It depends on the datastore, if every bank in
    this grouping is included per PCR (crude), or if each requested
    bank set is returned for each PCR individually (elegant).";
    container pcr {
      description
      "The composite of a PCR number and corresponding bank numbers.";
      leaf-list pcr-indices {
        type uint8;
        description
        "The number of the PCR. At the moment this is limited
        32";
      }
    }
  }
}
uses hash-algo;
}
}
}

grouping pcr-selector {
    description
    "A Verifier can request the generation of an attestation certificate (a signed public attestation key (non-migratable, tpm-resident) wrt one or more PCR values. The corresponding creation input is represented in this grouping. Requesting a PCR value that is not supported results in an error, detailed exposure via error msg should be avoided."
    list pcr-list {
        description
        "For each PCR in this list an individual hash-algo can be requested."
        container pcr {
            description
            "The composite of a PCR number and corresponding bank numbers."
            leaf-list pcr-index {
                type uint8;
                description
                "The numbers of the PCRs that are associated with the created key. At the moment the highest number is 32"
            }
            uses hash-algo;
        }
    }
}

grouping signature-scheme {
    description
    "The signature scheme used to sign the evidence."
    choice signature-identifier-type {
        mandatory true;
        description
        "There are multiple ways to reference a signature type. This used to select the signature algo to sign the quote information response."
        case TPM_ALG_ID {
            description
            "This references the indices of table 9 in the TPM 2.0 structure specification."
            leaf TPM_ALG_ID-value {
                type uint16;
                description
                "The TPM Algo ID."
            }
        }
    }
}
case COSE_Algorithm {
    description
    "This references the IANA COSE Algorithms Registry indices. Every index of this
    registry to be used must be mapable to a TPM_ALG_ID value.";
    leaf COSE_Algorithm-value {
        type int32;
        description
        "The TPM Algo ID.";
    }
}

grouping attestation-key-identifier {
    description
    "A selector for a suitable key identifier.";
    choice key-identifier {
        description
        "Identifier for the attestation key to use for signing
        attestation evidence.";
        case public-key {
            leaf pub-key-id {
                type binary;
                description
                "The value of the identifier for the public key.";
            }
        }
        case uuid {
            description
            "Use a YANG agent generated (and maintained) attestation
            key UUID.";
            leaf uuid-value {
                type binary;
                description
                "The UUID identifying the corresponding public key.";
            }
        }
    }
}

grouping tpm-name {
    description
    "In a system with multiple-TPMs get the data from a specific TPM
    identified by the name and physical-index.";
    leaf tpm_name {
        type string;
        description
        "This references the IANA COSE Algorithms Registry indices. Every index of this
        registry to be used must be mapable to a TPM_ALG_ID value.";
    }
}

"Name of the TPM or All";

leaf tpm-physical-index {
    if-feature ietfhw:entity-mib;
    type int32 {
        range "1..2147483647";
    }
    config false;
    description
        "The entPhysicalIndex for the TPM."
    reference
        "RFC 6933: Entity MIB (Version 4) - entPhysicalIndex";
}

grouping compute-node {
    description
        "In a distributed system with multiple compute nodes
        this is the node identified by name and physical-index.";
    leaf node-name {
        type string;
        description
            "Name of the compute node or All";
    }
    leaf node-physical-index {
        if-feature ietfhw:entity-mib;
        type int32 {
            range "1..2147483647";
        }
        config false;
        description
            "The entPhysicalIndex for the compute node."
        reference
            "RFC 6933: Entity MIB (Version 4) - entPhysicalIndex";
    }
}

identity log-type {
    description
        "The type of logs available.";
}

identity bios {
  base log-type;
  description
    "Measurement log created by the BIOS/UEFI."
}

identity ima {
  base log-type;
  description
    "Measurement log created by IMA."
}

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

grouping boot-event-log {
  description
    "Defines an event log corresponding to the event that extended the PCR";
  leaf event-number {
    type uint32;
    description
      "Unique event number of this event"
  }
  leaf event-type {
    type uint32;
    description
      "log event type"
  }
  leaf pcr-index {
    type uint16;
    description
      "Defines the PCR index that this event extended"
  }
  list digest-list {
    description
      "Hash of event data"
    uses hash-algo;
  }
  leaf-list digest {
  }
}
type binary;
  description
  "The hash of the event data";
}
}
leaf event-size {
  type uint32;
  description
  "Size of the event data";
}
leaf-list event-data {
  type uint8;
  description
  "the event data size determined by event-size";
}
}
}
}
}

grouping ima-event {
  description
  "Defines an hash log extend event for IMA measurements";
  leaf event-number {
    type uint64;
    description
    "Unique number for this event for sequencing";
  }
  leaf ima-template {
    type string;
    description
    "Name of the template used for event logs
    for e.g. ima, ima-ng";
  }
  leaf filename-hint {
    type string;
    description
    "File that was measured";
  }
  leaf filedata-hash {
    type binary;
    description
    "Hash of filedata";
  }
  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 uint16;
    description
    "Defines the PCR index that this event extended";
}

leaf signature {
    type binary;
    description
    "The file signature";
}

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

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

grouping event-logs {
    description
    "A selector for the log and its type.";
    choice log-type {
        mandatory true;
        description
        "Event log type determines the event logs content.";
    }

    case bios {

rpc tpm2-challenge-response-attestation {
    description "This RPC accepts the input for TSS commands of the managed device. ComponentIndex from the hardware manager YANG module to refer to dedicated TPM in composite devices, e.g. smart NICs, is still a TODO.";
    input {
        container tpm2-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 pcr-selection;
            uses nonce;
            uses signature-scheme;
            uses attestation-key-identifier;
        }
        uses tpm-name;
    }
    output {
        list tpm2-attestation-response {
            key tpm_name;
            description "The binary output of TPM2b_Quote. An TPMS_ATTEST structure";
        }
    }
}
Internet-Draft                    BRAT                        March 2019

including a length, encapsulated in a signature*

leaf pcrdigest |
  type binary;
  description
  "split out value of TPMS_QUOTE_INFO for convenience"
}
leaf tpm-attest-result |
  type binary;
  description
  "The complete TPM generate structure including signature*
leaf tpm-attest-result-length |
  type uint32;
  description
  "Length of attest result provided by the TPM structure*

leaf tpm-signature |
  type binary;
  description
  "Split out value of the signature for convenience. TODO: check for length values that complement binary value data node leafs.

rpc basic-trust-establishment |
  description
  "This RPC creates a tpm-resident, non-migratable key to be used in TPM_Quote commands, an attestation certificate."
input |
  uses nonce;
  uses signature-scheme;
  uses tpm-name;
  leaf certificate-name |
  type string;
  description
  "An arbitrary name for the identity certificate chain requested."

output |

list attestation-certificates {  
  key tpm_name;  
  description "Attestation Certificate data from a TPM identified by the TPM name";  
  uses tpm-name;  
  uses node-uptime;  
  uses compute-node;  
  leaf certificate-name {  
    type string;  
    description "An arbitrary name for this identity certificate or certificate chain.";  
  }  
  leaf attestation-certificate {  
    description "The binary signed certificate chain data for this identity certificate.";  
    type ietfct:end-entity-cert-cms;  
  }  
  uses attestation-key-identifier;  
} }  
  
rpc log-retrieval {  
  description "Logs Entries are either identified via indices or via 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 {  
    list log-selector {  
      key node-name;  
      description "Selection of log entries to be reported.";  
      uses compute-node;  
      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 {  
            description "Content of an log event which matches 1:1 with a unique event record contained within the log. Log entries subsequent to this will be passed to the requestor. Note: if log entry values are not unique, this MUST return an error.";  
        }  
      }  
    }  
  }  
}
type binary;
}
}
case index {
    description
    "Numeric index of the last log entry retrieved, or zero.";
    leaf index-number {
        description
        "The numeric index number of a log entry. Zero means to start at the beginning of the log. Entries subsequent to this will be passed to the requestor.";
        type uint64;
    }
}
case timestamp {
    leaf timestamp {
        type yang:date-and-time;
        description
        "Timestamp from which to start the extraction. The next log entry subsequent to this timestamp is to be sent.";
    }
    description
    "Timestamp from which to start the extraction.";
}
}
uses log-identifier;
uses pcr-selection;
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 {
            key node-name;
            description
            "Event logs of a node in a distributed system identified by the node name";
            uses compute-node;
            uses node-upptime;
            list tpm-updated {

key tpm_name;
  description
  "TPM these events may have recorded data in";
  uses tpm-name;
}
}
}
}
}

container log-result {
  description
  "The requested entries of the corresponding log.";
  uses event-logs;
}
}
}

container rats-support-structures {
  leaf-list supported-algos {
    type uint16;
    description
      "Supported TPM_ALG_ID values for the TPM in question.
      Will include ComponentIndex soon."
  }
  list tpms {
    key tpm_name;
    uses tpm-name;
    description
      "A list of TPMs in this composite device that rats can be conducted with."
  }
  list compute-nodes {
    key node-name;
    uses compute-node;
    description
      "A list names of hardware components in this composite device that rats can be conducted with."
  }
  container endorsement-certificates {
    list certificate {
      key tpm_name;
      uses tpm-name;
      description
        "The TPM’s endorsement-certificate."
    }
    leaf endorsement-certificate {
      type binary;
      mandatory true;
      description
        "The signed pulic endorsement key (EK) and corresponding claims (EK Certificate). In a TPM 2.0 the EK Certificate resides in a well-defined NVRAM location by the TPM vednor.";
    }
  }
}
3. IANA considerations

This document will include requests to IANA:

To be defined yet.

4. Security Considerations

There are always some.

5. Acknowledgements

Not yet.

6. Change Log

Changes from version 00 to version 01:

- Addressed author’s comments
- Extended complementary details about attestation-certificates
- Relabeled chunk-size to log-entry-quantity
- Relabeled location with compute-node or tpm-name where appropriate
- Added a valid entity-mib physical-index to compute-node and tpm-name to map it back to hardware inventory
- Relabeled name to tpm_name
- Removed event-string in last-entry

7. References
7.1. Normative References

[I-D.birkholz-reference-ra-interaction-model]

[I-D.ietf-netconf-crypto-types]


7.2. Informative References

[I-D.birkholz-attestation-terminology]

Authors’ Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Michael Eckel
Huawei Technologies
Feldbergstrasse 78
Darmstadt 64293
Germany

Email: michael.eckel@huawei.com