

A YANG Data Model for a Truststore
draft-ietf-netconf-trust-anchors-05

Abstract

This document defines a YANG 1.1 data model for configuring global sets of X.509 certificates and SSH host-keys that can be referenced by other data models for trust. While the SSH host-keys are uniquely for the SSH protocol, the X.509 certificates may have multiple uses, including authenticating protocol peers and verifying signatures.

Editorial Note (To be removed by RFC Editor)

This draft contains many placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. No other RFC Editor instructions are specified elsewhere in this document.

Artwork in this document contains shorthand references to drafts in progress. Please apply the following replacements:

- o "XXXX" --> the assigned RFC value for this draft
- o "YYYY" --> the assigned RFC value for [draft-ietf-netconf-crypto-types](#)

Artwork in this document contains placeholder values for the date of publication of this draft. Please apply the following replacement:

- o "2019-06-07" --> the publication date of this draft

The following Appendix section is to be removed prior to publication:

- o [Appendix A](#). Change Log

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 December 9, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	3
1.2.	Tree Diagram Notation	3
2.	The Trust Anchors Model	3
2.1.	Tree Diagram	3
2.2.	Example Usage	4
2.3.	YANG Module	7
3.	Security Considerations	14
4.	IANA Considerations	14
4.1.	The IETF XML Registry	14
4.2.	The YANG Module Names Registry	15
5.	References	15
5.1.	Normative References	15
5.2.	Informative References	15
Appendix A.	Change Log	17
A.1.	00 to 01	17
A.2.	01 to 02	17
A.3.	02 to 03	17
A.4.	03 to 04	17
A.5.	04 to 05	17
	Acknowledgements	18

Author's Address [18](#)

[1.](#) Introduction

This document defines a YANG 1.1 [[RFC7950](#)] data model for configuring global sets of X.509 certificates and SSH host-keys that can be referenced by other data models for trust. While the SSH host-keys are uniquely for the SSH protocol, the X.509 certificates may be used for multiple uses, including authenticating protocol peers and verifying signatures.

This document is compliant with Network Management Datastore Architecture (NMDA) [[RFC8342](#)]. For instance, to support trust anchors installed during manufacturing, it is expected that such data may appear only in <operational>.

[1.1.](#) Requirements Language

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.

[1.2.](#) Tree Diagram Notation

Tree diagrams used in this document follow the notation defined in [[RFC8340](#)].

[2.](#) The Trust Anchors Model

[2.1.](#) Tree Diagram

The following tree diagram provides an overview of the "ietf-truststore" module.

```

module: ietf-truststore
  +-rw truststore
    +-rw certificates* [name] {x509-certificates}?
      | +-rw name          string
      | +-rw description?  string
      | +-rw certificate* [name]
      |   +-rw name          string
      |   +-rw cert          trust-anchor-cert-cms
      |   +---n certificate-expiration
      |     +- expiration-date  yang:date-and-time
    +-rw host-keys* [name] {ssh-host-keys}?
      +-rw name          string
  
```



```

    +--rw description?   string
    +--rw host-key* [name]
        +--rw name       string
        +--rw host-key   ct:ssh-host-key

grouping local-or-truststore-certs-grouping
  +-- (local-or-truststore)
    +--:(local) {local-definitions-supported}?
      | +-- local-definition
      |   +-- cert*                               trust-anchor-cert-cms
      |   +---n certificate-expiration
      |       +-- expiration-date   yang:date-and-time
    +--:(truststore) {truststore-supported,x509-certificates}?
      +-- truststore-reference?  ts:certificates-ref

grouping local-or-truststore-host-keys-grouping
  +-- (local-or-truststore)
    +--:(local) {local-definitions-supported}?
      | +-- local-definition
      |   +-- host-key*             ct:ssh-host-key
      |   +-- cert*                 trust-anchor-cert-cms
      |   +---n certificate-expiration
      |       +-- expiration-date   yang:date-and-time
    +--:(truststore) {truststore-supported,ssh-host-keys}?
      +-- truststore-reference?  ts:host-keys-ref

grouping truststore-grouping
  +-- certificates* [name] {x509-certificates}?
    | +-- name?           string
    | +-- description?    string
    | +-- certificate* [name]
    |   +-- name?         string
    |   +-- cert          trust-anchor-cert-cms
    |   +---n certificate-expiration
    |       +-- expiration-date   yang:date-and-time
  +-- host-keys* [name] {ssh-host-keys}?
    +-- name?           string
    +-- description?    string
    +-- host-key* [name]
      +-- name?         string
      +-- host-key      ct:ssh-host-key

```

2.2. Example Usage

The following example illustrates trust anchors in <operational> as described by [Section 5.3 in \[RFC8342\]](#). This datastore view illustrates data set by the manufacturing process alongside conventional configuration. This trust anchors instance has six sets of pinned certificates and one set of pinned host keys.

Watsen

Expires December 9, 2019

[Page 4]

```
<truststore
  xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore"
  xmlns:or="urn:ietf:params:xml:ns:yang:ietf-origin">

  <!-- Manufacturer's trusted root CA certs -->
  <certificates or:origin="or:system">
    <name>manufacturers-root-ca-certs</name>
    <description>
      Certificates built into the device for authenticating
      manufacturer-signed objects, such as TLS server certificates,
      vouchers, etc. Note, though listed here, these are not
      configurable; any attempt to do so will be denied.
    </description>
    <certificate>
      <name>Manufacturer Root CA cert 1</name>
      <cert>base64encodedvalue==</cert>
    </certificate>
    <certificate>
      <name>Manufacturer Root CA cert 2</name>
      <cert>base64encodedvalue==</cert>
    </certificate>
  </certificates>

  <!-- specific end-entity certs for authenticating servers -->
  <certificates or:origin="or:intended">
    <name>explicitly-trusted-server-certs</name>
    <description>
      Specific server authentication certificates for explicitly
      trusted servers. These are needed for server certificates
      that are not signed by a CA.
    </description>
    <certificate>
      <name>Fred Flintstone</name>
      <cert>base64encodedvalue==</cert>
    </certificate>
  </certificates>

  <!-- trusted CA certs for authenticating servers -->
  <certificates or:origin="or:intended">
    <name>explicitly-trusted-server-ca-certs</name>
    <description>
      Trust anchors (i.e. CA certs) that are used to authenticate
      server connections. Servers are authenticated if their
      certificate has a chain of trust to one of these CA
      certificates.
    </description>
    <certificate>
      <name>ca.example.com</name>
```



```
<cert>base64encodedvalue==</cert>
</certificate>
</certificates>

<!-- specific end-entity certs for authenticating clients -->
<certificates or:origin="or:intended">
  <name>explicitly-trusted-client-certs</name>
  <description>
    Specific client authentication certificates for explicitly
    trusted clients. These are needed for client certificates
    that are not signed by a CA.
  </description>
  <certificate>
    <name>George Jetson</name>
    <cert>base64encodedvalue==</cert>
  </certificate>
</certificates>

<!-- trusted CA certs for authenticating clients -->
<certificates or:origin="or:intended">
  <name>explicitly-trusted-client-ca-certs</name>
  <description>
    Trust anchors (i.e. CA certs) that are used to authenticate
    client connections. Clients are authenticated if their
    certificate has a chain of trust to one of these CA
    certificates.
  </description>
  <certificate>
    <name>ca.example.com</name>
    <cert>base64encodedvalue==</cert>
  </certificate>
</certificates>

<!-- trusted CA certs for random HTTPS servers on Internet -->
<certificates or:origin="or:system">
  <name>common-ca-certs</name>
  <description>
    Trusted certificates to authenticate common HTTPS servers.
    These certificates are similar to those that might be
    shipped with a web browser.
  </description>
  <certificate>
    <name>ex-certificate-authority</name>
    <cert>base64encodedvalue==</cert>
  </certificate>
</certificates>

<!-- specific SSH host keys for authenticating clients -->
```

Watsen

Expires December 9, 2019

[Page 6]

```

<host-keys or:origin="or:intended">
  <name>explicitly-trusted-ssh-host-keys</name>
  <description>
    Trusted SSH host keys used to authenticate SSH servers.
    These host keys would be analogous to those stored in
    a known_hosts file in OpenSSH.
  </description>
  <host-key>
    <name>corp-fw1</name>
    <host-key>base64encodedvalue==</host-key>
  </host-key>
</host-keys>

```

```

</truststore>

```

The following example illustrates the "certificate-expiration" notification in use with the NETCONF protocol.

===== NOTE: '\ ' line wrapping per BCP XX (RFC XXXX) =====

```

<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2018-05-25T00:01:00Z</eventTime>
  <truststore xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore">
    <certificates>
      <name>explicitly-trusted-client-certs</name>
      <certificate>
        <name>George Jetson</name>
        <certificate-expiration>
          <expiration-date>2018-08-05T14:18:53-05:00</expiration-dat\
e>
        </certificate-expiration>
      </certificate>
    </certificates>
  </truststore>
</notification>

```

2.3. YANG Module

This YANG module imports modules from [[RFC8341](#)] and [[I-D.ietf-netconf-crypto-types](#)].

```

<CODE BEGINS> file "ietf-truststore@2019-06-07.yang"
module ietf-truststore {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-truststore";
  prefix ts;

```



```
import ietf-netconf-acm {
  prefix nacm;
  reference
    "RFC 8341: Network Configuration Access Control Model";
}

import ietf-crypto-types {
  prefix ct;
  reference
    "RFC YYYY: Common YANG Data Types for Cryptography";
}

organization
  "IETF NETCONF (Network Configuration) Working Group";

contact
  "WG Web:   <http://datatracker.ietf.org/wg/netconf/>
  WG List:  <mailto:netconf@ietf.org>
  Author:   Kent Watsen <mailto:kent+ietf@watsen.net>";

description
  "This module defines a truststore to centralize management
  of trust anchors including both X.509 certificates and
  SSH host keys.

  Copyright (c) 2019 IETF Trust and the persons identified
  as authors of the code. All rights reserved.

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

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

  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.";

revision 2019-06-07 {
  description
```



```
    "Initial version";
  reference
    "RFC XXXX: A YANG Data Model for a Truststore";
}

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

feature truststore-supported {
  description
    "The 'truststore-supported' feature indicates that the
    server supports the truststore.";
}

feature local-definitions-supported {
  description
    "The 'local-definitions-supported' feature indicates that
    the server supports locally-defined trust anchors.";
}

feature x509-certificates {
  description
    "The 'x509-certificates' feature indicates that the server
    implements the /truststore/certificates subtree.";
}

feature ssh-host-keys {
  description
    "The 'ssh-host-keys' feature indicates that the server
    implements the /truststore/host-keys subtree.";
}

/*****/
/*   Typedefs   */
/*****/

typedef certificates-ref {
  type leafref {
    path "/ts:truststore/ts:certificates/ts:name";
  }
  description
    "This typedef enables modules to easily define a reference
    to a set of certificates defined in the truststore.";
}

typedef host-keys-ref {
  type leafref {
```



```
    path "/ts:truststore/ts:host-keys/ts:name";
  }
  description
    "This typedef enables modules to easily define a reference
    to a set of host keys defined in the truststore.";
}

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

grouping local-or-truststore-certs-grouping {
  description
    "A grouping that expands to allow trust anchors to be
    either stored locally, within the using data model, or be
    a reference to trust anchors stored in the truststore.";
  choice local-or-truststore {
    mandatory true;
    case local {
      if-feature "local-definitions-supported";
      container local-definition {
        description
          "Container to hold the local trust anchor definitions.
          A list is defined so as to be symmetric with the
          truststore definition.";
        uses ct:trust-anchor-certs-grouping;
      }
    }
    case truststore {
      if-feature "truststore-supported";
      if-feature "x509-certificates";
      leaf truststore-reference {
        type ts:certificates-ref;
        description
          "A reference to a set of trust anchors that exists
          in the truststore.";
      }
    }
  }
  description
    "A choice between an inlined definition and a definition
    that exists in the truststore.";
}

grouping local-or-truststore-host-keys-grouping {
  description
    "A grouping that expands to allow trust anchors to be
    either stored locally, within the using data model, or be
```



```
    a reference to trust anchors stored in the truststore.";
choice local-or-truststore {
  mandatory true;
  case local {
    if-feature "local-definitions-supported";
    container local-definition {
      description
        "Container to hold the local trust anchor definitions.
        A list is defined so as to be symmetric with the
        truststore definition.";
      leaf-list host-key {
        nacm:default-deny-write;
        type ct:ssh-host-key;
        description
          "The binary data for this host key.";
        reference
          "RFC YYYY: Common YANG Data Types for Cryptography";
      }
      uses ct:trust-anchor-certs-grouping;
    }
  }
  case truststore {
    if-feature "truststore-supported";
    if-feature "ssh-host-keys";
    leaf truststore-reference {
      type ts:host-keys-ref;
      description
        "A reference to a set of trust anchors that exists
        in the truststore.";
    }
  }
  description
    "A choice between an inlined definition and a definition
    that exists in the truststore.";
}

grouping truststore-grouping {
  description
    "Grouping definition enables use in other contexts.  If ever
    done, implementations SHOULD augment new 'case' statements
    into local-or-keystore 'choice' statements to supply leafrefs
    to the new location.";
  list certificates {
    if-feature "x509-certificates";
    key "name";
    description
```



```
    "A list of certificates. These certificates can be
    used by a server to authenticate clients, or by a client
    to authenticate servers. Each list of certificates
    SHOULD be specific to a purpose, as the list as a whole
    may be referenced by other modules. For instance, a
    RESTCONF server's configuration might use a specific list
    of certificates for when authenticating RESTCONF
    client connections.";
  leaf name {
    type string;
    description
      "An arbitrary name for this list of certificates.";
  }
  leaf description {
    type string;
    description
      "An arbitrary description for this list of
      certificates.";
  }
  list certificate {
    key "name";
    description
      "A certificate.";
    leaf name {
      type string;
      description
        "An arbitrary name for this certificate. The
        name must be unique across all lists of
        certificates (not just this list) so that leafrefs
        from another module can resolve to unique values.";
    }
    uses ct:trust-anchor-cert-grouping {
      refine "cert" {
        mandatory true;
      }
    }
  }
}

list host-keys {
  if-feature "ssh-host-keys";
  key "name";
  description
    "A list of host keys. These host-keys can
    be used by clients to authenticate SSH servers. Each
    list of host keys SHOULD be specific to a purpose,
    so the list as a whole may be referenced by other modules.
    For instance, a NETCONF client's configuration might
    point to a specific list of host keys for when
```



```
        authenticating specific SSH servers.";
    leaf name {
        type string;
        description
            "An arbitrary name for this list of SSH
             host keys.";
    }
    leaf description {
        type string;
        description
            "An arbitrary description for this list of SSH
             host keys.";
    }
    list host-key {
        key "name";
        description
            "A host key.";
        leaf name {
            type string;
            description
                "An arbitrary name for this host-key. Must be
                 unique across all lists of host-keys (not just
                 this list) so that a leafref to it from another module
                 can resolve to unique values.";
        }
        leaf host-key {
            type ct:ssh-host-key;
            mandatory true;
            description
                "The binary public key data for this host key.";
            reference
                "RFC YYYY: Common YANG Data Types for Cryptography";
        }
    }
}

/*****
/*  Protocol accessible nodes  */
*****/

container truststore {
    nacm:default-deny-write;
    description
        "The truststore contains sets of X.509 certificates and
         SSH host keys.";
    uses truststore-grouping;
}
```



```
}  
<CODE ENDS>
```

3. Security Considerations

The YANG module defined in this document is designed to be accessed via YANG based management protocols, such as NETCONF [[RFC6241](#)] and RESTCONF [[RFC8040](#)]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) [[RFC8341](#)] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

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 and their sensitivity/vulnerability:

/: The entire data tree defined by this module is sensitive to write operations. For instance, the addition or removal of any trust anchor may dramatically alter the implemented security policy. For this reason, the NACM extension "default-deny-write" has been set for the entire data tree.

None of the readable data nodes in this YANG module are considered sensitive or vulnerable in network environments.

This module does not define any RPCs, actions, or notifications, and thus the security consideration for such is not provided here.

4. IANA Considerations

4.1. The IETF XML Registry

This document registers one URI in the "ns" subregistry of the IETF XML Registry [[RFC3688](#)]. Following the format in [[RFC3688](#)], the following registration is requested:

URI: urn:ietf:params:xml:ns:yang:ietf-truststore
Registrant Contact: The NETCONF WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

4.2. The YANG Module Names Registry

This document registers one YANG module in the YANG Module Names registry [RFC6020]. Following the format in [RFC6020], the the following registration is requested:

```
name:          ietf-truststore
namespace:     urn:ietf:params:xml:ns:yang:ietf-truststore
prefix:        ta
reference:     RFC XXXX
```

5. References

5.1. Normative References

- [I-D.ietf-netconf-crypto-types]
Watsen, K. and H. Wang, "Common YANG Data Types for Cryptography", [draft-ietf-netconf-crypto-types-06](#) (work in progress), April 2019.
- [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>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [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>>.

5.2. Informative References

- [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>>.
- [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>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", [BCP 215](#), [RFC 8340](#), DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

[Appendix A.](#) Change Log

[A.1.](#) 00 to 01

- o Added features "x509-certificates" and "ssh-host-keys".
- o Added nacm:default-deny-write to "trust-anchors" container.

[A.2.](#) 01 to 02

- o Switched "list pinned-certificate" to use the "trust-anchor-cert-grouping" from crypto-types. Effectively the same definition as before.

[A.3.](#) 02 to 03

- o Updated copyright date, boilerplate template, affiliation, folding algorithm, and reformatted the YANG module.

[A.4.](#) 03 to 04

- o Added groupings 'local-or-truststore-certs-grouping' and 'local-or-truststore-host-keys-grouping', matching similar definitions in the keystore draft. Note new (and incomplete) "truststore" usage!
- o Related to above, also added features 'truststore-supported' and 'local-trust-anchors-supported'.

[A.5.](#) 04 to 05

- o Renamed "trust-anchors" to "truststore"
- o Removed "pinned." prefix everywhere, to match truststore rename
- o Moved everything under a top-level 'grouping' to enable use in other contexts.
- o Renamed feature from 'local-trust-anchors-supported' to 'local-definitions-supported' (same name used in keystore)
- o Removed the "require-instance false" statement from the "*-ref" typedefs.
- o Added missing "ssh-host-keys" and "x509-certificates" if-feature statements

Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by last name): Martin Bjorklund, Nick Hancock, Balazs Kovacs, Eric Voit, and Liang Xia.

Author's Address

Kent Watsen
Watsen Networks

EMail: kent+ietf@watsen.net