SIDROPS Working Group
Internet Draft
Intended status: Standards Track
Expires: July 20, 2026

Y. Liu
China Mobile
C. Lin
New H3C Technologies
H. Wang
Huawei
J. Roy
J. Haas
Juniper Networks, Inc.
H. Liu
ZTE
D. Ma
ZDNS

January 20, 2026

YANG Data Model for RPKI to Router Protocol
draft-ietf-sidrops-rtr-yang-01

Abstract

This document defines YANG data models for configuring and managing Resource Public Key Infrastructure (RPKI) to Router Protocol (RFC6810 and RFC8210).

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 July 20, 2026.

Table of Contents

1. Introduction

[RFC6810] and [RFC8210] describes a protocol to deliver Resource Public Key Infrastructure (RPKI) prefix origin data and router keys from a trusted cache server to a router, referred to as RPKI-Router protocol.

This document defines YANG [RFC7950] data models for configuring and managing RPKI-Router Protocol ([RFC6810], [RFC8210], and [I-D.ietf-sidrops-8210bis]).

1.1. Terminology

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. Model Overview

Two YANG data models are defined in this document.

The ietf-rpki-rtr.yang data model provides the methods for configuring and managing RPKI-Router Protocol. It includes:

- Connectivity parameters, such as RPKI cache server IP address and destination port.
- Session parameters, such as purge time, refresh time, response time.
- Session status and statistics, such as session ID, serial number, number of received and transmitted messages.

The ietf-rpki-table.yang data model provides the methods for managing records of RPKI-Router Protocol. It includes:

- ROA records.
- Router-key records.
- ASPA records.

3. RPKI to Router YANG Module

3.1. Tree View

The complete tree of the ietf-rpki-rtr.yang data model is represented as following. See [RFC8340] for an explanation of the symbols used.

```
   module: ietf-rpki-rtr

     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol:
       +--rw rpki-rtr
          +--rw sessions
             +--rw session* [server-address]
                +--rw server-address           inet:ip-address
                +--rw server-port?             inet:port-number
                +--rw local-address?           union
                +--rw local-port?              inet:port-number
                +--rw enabled?                 boolean
                +--rw preference?              uint32
                +--rw description?             string
                +--ro session-state?           enumeration
                +--rw enable-authentication?   boolean
                +--rw authentication
                |  +--rw (option)?
                |     +--:(md5)
                |     |  +--rw md5-password?   ianach:crypt-hash
                |     +--:(ssh)
                |     |  +--rw client-identity
                |     |  |  +--rw username?  string
                |     |  |  +--rw public-key!
                |     |  |                   {userauth-publickey}?
                |     |  |  +--rw password!
                |     |  |                   {userauth-password}?
                |     |  |  +--rw hostbased!
                |     |  |                   {userauth-hostbased}?
                |     |  |  +--rw none?      empty {userauth-none}?
                |     |  |  +--rw certificate!
                |     |  |                   {sshcmn:ssh-x509-certs}?
                |     |  +--rw server-authentication
                |     |  |  +--rw ssh-host-keys!
                |     |  |  +--rw ca-certs!  {sshcmn:ssh-x509-certs}?
                |     |  |  +--rw ee-certs!  {sshcmn:ssh-x509-certs}?
                |     |  +--rw transport-params
                |     |  |       {ssh-client-transport-params-config}?
                |     |  +--rw keepalives!   {ssh-client-keepalives}?
                |     |     +--rw max-wait?       uint16
                |     |     +--rw max-attempts?   uint8
                |     +--:(tcp-ao-keychain)
                |        +--rw keychain-name?  key-chain:key-chain-ref
                +--rw roa-limit
                |  +--rw max-number?              uint64
                |  +--rw threshold-percentage?    uint8
                |  +--rw over-threshold-action?   enumeration
                |  +--rw reconnect-interval?      uint32
```

```
               +--rw aspa-limit
               |  +--rw max-number?              uint64
               |  +--rw threshold-percentage?    uint8
               |  +--rw over-threshold-action?   enumeration
               |  +--rw reconnect-interval?      uint32
               +--ro statistics
               |  +--ro total-roa-records? yang:zero-based-counter64
               |  +--ro ipv4-roa-records?  yang:zero-based-counter64
               |  +--ro ipv6-roa-records?  yang:zero-based-counter64
               |  +--ro router-key-records?
               |  |                        yang:zero-based-counter64
               |  +--ro aspa-records? yang:zero-based-counter64
               +--ro connection-data
               |  +--ro flaps?                          uint32
               |  +--ro last-session-up-down?           yang:timestamp
               |  +--ro last-update-sync-timestamp?     yang:timestamp
               |  +--ro last-full-sync-timestamp?       yang:timestamp
               |  +--ro last-serial-query-timestamp?    yang:timestamp
               |  +--ro last-reset-query-timestamp?     yang:timestamp
               |  +--ro last-eod-received?              yang:timestamp
               |  +--ro last-config-change-timestamp? yang:timestamp
               |  +--ro last-error-timestamp?           yang:timestamp
               |  +--ro last-connection-error-timestamp?
               |  |                                     yang:timestamp
               |  +--ro last-connection-timestamp?      yang:timestamp
               |  +--ro error-reason?                   string
               +--ro protocol-data
               |  +--ro protocol-version?   uint32
               |  +--ro refresh-time?       yang:timestamp
               |  +--ro response-time?      yang:timestamp
               |  +--ro purge-time?         yang:timestamp
               |  +--ro hold-time?          yang:timestamp
               |  +--ro record-lifetime?    yang:timestamp
               |  +--ro retry-interval?     uint32
               |  +--ro expire-interval?    uint32
               |  +--ro session-id?         uint16
               |  +--ro serial-full?        uint32
               |  +--ro serial-incremental? uint32
               |  +--ro in-total-messages?  yang:zero-based-counter64
               |  +--ro out-total-messages? yang:zero-based-counter64
               +--ro pdu-counters
               |  +--ro serial-notify?    yang:zero-based-counter64
               |  +--ro cache-response?   yang:zero-based-counter64
               |  +--ro ipv4-prefix?      yang:zero-based-counter64
               |  +--ro ipv6-prefix?      yang:zero-based-counter64
               |  +--ro end-of-data?      yang:zero-based-counter64
               |  +--ro cache-reset?      yang:zero-based-counter64
               |  +--ro reset-query?      yang:zero-based-counter64
```

```
               |  +--ro serial-query?     yang:zero-based-counter64
               +--ro error-pdu-counters
                  +--ro corrupt-data?      yang:zero-based-counter64
                  +--ro internal-error?    yang:zero-based-counter64
                  +--ro unsupported-protocol-version?
                  |                        yang:zero-based-counter64
                  +--ro unsupported-pdu-type?
                  |                        yang:zero-based-counter64
                  +--ro unexpected-protocol-version?
                  |                        yang:zero-based-counter64
                  +--ro no-data-available? yang:zero-based-counter64
                  +--ro invalid-request?   yang:zero-based-counter64
                  +--ro withdrawal-unknown-record?
                  |                        yang:zero-based-counter64
                  +--ro duplicate-announcement-received?
                                           yang:zero-based-counter64
```

3.2. Yang Module

<CODE BEGINS> file "ietf-rpki-rtr@2022-10-18.yang"

```
module ietf-rpki-rtr {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-rpki-rtr";
  prefix "rpki-rtr";
  import ietf-yang-types {
    prefix "yang";
    reference
      "RFC 6991: Common YANG Data Types.";
  }
  import ietf-inet-types {
    prefix "inet";
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349: A YANG Data Model for Routing Management
       (NMDA Version).";
  }
  import iana-crypt-hash {
    prefix "ianach";
    reference
      "RFC 7317: A YANG Data Model for System Management";
  }
  import ietf-ssh-client {
    prefix "ssh";
    reference
      "RFC XXXX: YANG Groupings for SSH Clients and SSH Servers";
  }
  import ietf-interfaces {
    prefix "if";
       reference
         "RFC 8343, A YANG Data Model for Interface Management.";
  }
  import ietf-key-chain {
    prefix key-chain;
    reference
      "RFC 8177: YANG Key Chain.";
  }
  organization
    "IETF SIDROPS Working Group";
  contact
    "TBD";
```

```
    description
      "This module describes a YANG model for the Resource Public
       Key Infrastructure (RPKI) to Router Protocol configuration.
       This YANG model conforms to the Network Management
       Datastore Architecture (NMDA) as described in RFC 8342.
       Copyright (c) 2022 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 XXXX;
       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.";
    reference "RFC 8210";
    revision 2022-10-18 {
      description
        "Initial Version";
      reference
        "RFC 8210, YANG Data Model for RPKI to Router Protocol";
    }
    typedef ipv4-pfx-len {
      type uint8 {
        range "0 .. 32";
      }
      description
        "IPv4 Prefix Length.";
    }
    typedef ipv6-pfx-len {
      type uint8 {
        range "0 .. 128";
      }
      description
        "IPv6 Prefix Length.";
    }
    typedef subject-key-id {
      type binary {
        length 20;
      }
      description
        "Subject Key Identifier.";
    }
```

```
     identity rpki-rtr {
       base rt:routing-protocol;
       description
         "RPKI to Router protocol.";
     }
     grouping records-limit {
       description
         "Limit of records that can be received from the RPKI
          cache server.";
       leaf max-number {
         type uint64;
         description
           "Configures the maximum number of ROAs that can be
            received from the RPKI cache server.";
       }
       leaf threshold-percentage {
         type uint8 {
           range "0..100";
         }
         units "percent";
         description
           "Configures the threshold percentage for ROA maximum
            number.";
       }
       leaf over-threshold-action {
         type enumeration {
           enum alert-only {
             description
               "Generates alert messages.";
           }
           enum discard {
             description
               "Discards excess ROAs.";
           }
           enum reconnect {
             description
               "Diconncets with the RPKI cache server,
                and tries to reconnect after reconnection
                timer expires.";
           }
           enum idle-forever {
             description
               "Diconncets with the RPKI cache server
                forever.";
           }
         }
         description
           "The action to taken when ROA number exceeds
```

```
            threshold.";
        }
        leaf reconnect-interval {
          type uint32 {
            range "1..30000";
          }
          units "minutes";
          description
            "Time interval for the reconnection timer.";
        }
      }
      grouping aspa-overall-records {
        description
          "ASPAs received from all RPKI cache servers.";
        list aspas {
          key "customer-asn";
          description
            "An entry of ASPA.";
          leaf customer-asn {
            type inet:as-number;
            description
              "The AS number of a customer.";
          }
          leaf server-address {
            type inet:ip-address;
            description
              "IP address of the RPKI cache server.";
          }
          list provider-asns {
            key "provider-asn";
            description
              "Providers of the customer.";
            leaf provider-asn {
              type inet:as-number;
              description
                "The AS number of a provider.";
            }
          }
        }
      }
      grouping aspa-server-records {
        description
          "ASPAs received from a RPKI cache server.";
        list aspas {
          key "customer-asn";
          description
            "An entry of ASPA.";
          leaf customer-asn {
```

```
          type inet:as-number;
          description
            "The AS number of a customer.";
        }
        list provider-asns {
          key "provider-asn";
          description
            "Providers of the customer.";
          leaf provider-asn {
            type inet:as-number;
            description
              "The AS number of a provider.";
          }
        }
      }
    }
    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol" {
      when "derived-from-or-self(rt:type, 'rpki-rtr')" {
        description
          "This augmentation is valid for a routing protocol
           instance of RPKI to Router.";
      }
      description
        "RPKI to Router protocol augmentation of ietf-routing module
         control-plane-protocol.";
      container rpki-rtr {
        description
          "Configuration parameters for the RPKI to Router Protocol.";
        container sessions {
          description
            "Parameters of RPKI sessions to cache servers.";
          list session {
            key "server-address";
            description
              "Each entry contains parameters for a RPKI session
               identified by the 'server-address' key.";
            leaf server-address {
              type inet:ip-address;
              mandatory true;
              description
                "The IP address of the RPKI cache server resembling
                 a session";
            }
            leaf server-port {
              type inet:port-number;
              description
                "The remote port for the connection
```

```
                to the RPKI cache server";
            }
            leaf local-address {
              type union {
                type inet:ip-address;
                type if:interface-ref;
              }
              description
                "The local IP (either IPv4 or IPv6) address to use for
                 the connection to the RPKI cache server. This may be
                 expressed as either an IP address or reference to the
                 name of an interface.";
            }
            leaf local-port {
              type inet:port-number;
              description
                "The local port for the connection
                 to the RPKI cache server";
            }
            leaf enabled {
              type boolean;
              default "true";
              description
                "Whether the RPKI cache server is enabled.";
            }
            leaf preference {
              type uint32;
              description
                "The router's preference to connect to that cache.
                 The lower the value, the more preferred.";
            }
            leaf description {
              type string;
              description
                "Textual description of the RPKI cache server";
            }
            leaf session-state {
              type enumeration {
                enum idle {
                  description
                    "The session is down.";
                }
                enum connect {
                  description
                    "The session is waiting for the underlying
                     transport session to be established.";
                }
                enum establish {
```

```
                description
                  "The session is up.";
              }
              enum ex-incr {
                description
                  "Incremental update of ROAs in progress.";
              }
              enum ex-full {
                description
                  "Full update of ROA records in progress.";
              }
            }
            config false;
            description
              "The session state.";
          }
          leaf enable-authentication {
            type boolean;
            default "false";
            description
              "Whether the session is secured.";
          }
          container authentication {
            when "../enable-authentication = 'true'";
            description
              "Container for describing how a particular session
               is to be secured.";
            choice option {
              description
                "Choice for session secruring methods.";
              case md5 {
                leaf md5-password {
                  type ianach:crypt-hash;
                  description
                    "The password for md5 authentication.";
                }
                description
                  "Uses TCP-MD5 to secure the session.";
              }
              case ssh {
                 uses ssh:ssh-client-grouping {
                  reference
                    "RFC XXXX: YANG Groupings for SSH Clients and
                     SSH Servers";
                }
                description
                  "Uses SSH to secure the session.";
              }
```

```
                case tcp-ao-keychain {
                  leaf keychain-name {
                    type key-chain:key-chain-ref;
                    description
                      "Name of key chain.";
                    reference
                      "RFC 8177: YANG Key Chain.";
                  }
                  description
                    "Uses key-chain to secure the session.";
                }
              }
            }
            container roa-limit {
              description
                "Limit of ROA records that can be received from the
                 RPKI cache server.";
              uses records-limit;
            }
            container aspa-limit {
              description
                "Limit of ASPA records that can be received from the
                 RPKI cache server.";
              uses records-limit;
            }
            container statistics {
              config false;
              description
                "Statistics of the RPKI cache server.";
              leaf total-roa-records {
                type yang:zero-based-counter64;
                description
                  "The total number of ROAs received
                   from the RPKI cache server.";
              }
              leaf ipv4-roa-records {
                type yang:zero-based-counter64;
                description
                  "The number of ROAs for IPv4 prefixes received
                   from the RPKI cache server.";
              }
              leaf ipv6-roa-records {
                type yang:zero-based-counter64;
                description
                  "The number of ROAs for IPv6 prefixes received
                   from the RPKI cache server.";
              }
              leaf router-key-records {
```

```
            type yang:zero-based-counter64;
            description
              "The number of router keys received from the RPKI
               cache server.";
          }
          leaf aspa-records {
            type yang:zero-based-counter64;
            description
              "The number of ASPAs received from the RPKI
               cache server.";
          }
        }
        container connection-data {
          config false;
          description
            "State information relating to the connection
             with the RPKI cache server.";
          leaf flaps {
            type uint32;
            description
              "Count for number of flaps observed on the
               session.";
          }
          leaf last-session-up-down {
            type yang:timestamp;
            description
              "This timestamp indicates the time that the
               RPKI-RTR session last transitioned in or out
               of the UP state. The value is the timestamp in
               microseconds relative to the Unix Epoch (Jan 1,
               1970 00:00:00 UTC). The RPKI-RTR session uptime
               can be computed by clients as the difference
               between this value and the current time
               in UTC (assuming the session is in the UP
               state, per the session-state leaf).";
            reference
              "RFC 6810: The Resource Public Key Infrastructure.";
          }
          leaf last-update-sync-timestamp {
            type yang:timestamp;
            description
              "Time of last serial sync with cache server.";
          }
          leaf last-full-sync-timestamp {
            type yang:timestamp;
            description
              "Time of last reset sync with cache server.";
          }
```

```
            leaf last-serial-query-timestamp {
              type yang:timestamp;
              description
                "Time of last serial query sent to cache server.";
            }
            leaf last-reset-query-timestamp {
              type yang:timestamp;
              description
                "Time of last reset query sent to cache server.";
            }
            leaf last-eod-received {
              type yang:timestamp;
              description
                "Time in microseconds at which last EOD was
                 received.";
            }
            leaf last-config-change-timestamp {
              type yang:timestamp;
              description
                "Time of last host, port, VRF or local interface
                 change.";
            }
            leaf last-error-timestamp {
              type yang:timestamp;
              description
                "Time of sending/receiving protocol error to/from
                 cache server.";
            }
            leaf last-connection-error-timestamp {
              type yang:timestamp;
              description
                "Time of last connection error to cache server.";
            }
            leaf last-connection-timestamp {
              type yang:timestamp;
              description
                "Time of last connection to cache server.";
            }
            leaf error-reason {
              type string;
              description
                "Reason for error in connection.";
            }
          }
          container protocol-data {
            config false;
            description
              "State parameters related to the RPKI to router
```

```
   protocol";
leaf protocol-version {
  type uint32;
  description
    "The version number of the RPKI to Router
    Protocol.";
}
leaf refresh-time {
  type yang:timestamp;
  description
    "Configures the time a router waits in between
     sending periodic serial queries to the RPKI
     cache server.";
}
leaf response-time {
  type yang:timestamp;
  description
    "Configures the time a router waits for a response
     after sending a serial or reset query to the RPKI
     cache server.";
}
leaf purge-time {
  type yang:timestamp;
  description
    "Configures the time a router waits to keep data
     from the RPKI cache server after the session
     drops.";
}
leaf hold-time {
  type yang:timestamp;
  description
    "Hold-time for this session.";
}
leaf record-lifetime {
  type yang:timestamp;
  description
    "Record-lifetime this session.";
}
leaf retry-interval {
  type uint32;
  description
    "Number of seconds between poll error and cache
     server poll";
}
leaf expire-interval {
  type uint32;
  description
    "Number of seconds to retain data synced from
```

```
            cache server";
        }
        leaf session-id {
          type uint16;
          config false;
          description
            "When a cache server is started, it generates a
             Session ID to identify the instance of the cache
             and to bind it to the sequence of Serial Numbers
             that cache instance will generate.";
          reference
            "RFC 6810, The Resource Public Key Infrastructure
             (RPKI) to Router Protocol
             RFC 8210, The Resource Public Key Infrastructure
             (RPKI) to Router Protocol, Version 1";
        }
        leaf serial-full {
          type uint32;
          config false;
          description
            "A 32-bit strictly increasing unsigned integer which
             wraps from 2^32-1 to 0. It denotes the logical
             version of a cache. It resembles the latest full
             query.";
          reference
            "RFC 6810, The Resource Public Key Infrastructure
             (RPKI) to Router Protocol
             RFC 8210, The Resource Public Key Infrastructure
             (RPKI) to Router Protocol, Version 1";
        }
        leaf serial-incremental {
          type uint32;
          config false;
          description
            "A 32-bit strictly increasing unsigned integer which
             wraps from 2^32-1 to 0. It denotes the logical
             version of a cache. It resembles the latest
             incremental query.";
          reference
            "RFC 6810, The Resource Public Key Infrastructure
             (RPKI) to Router Protocol
             RFC 8210, The Resource Public Key Infrastructure
             (RPKI) to Router Protocol, Version 1";
        }
        leaf in-total-messages {
          type yang:zero-based-counter64;
          description
            "The total number of messages received from the
```

```
                RPKI cache server.";
            }
            leaf out-total-messages {
              type yang:zero-based-counter64;
              description
                "The total number of messages transmitted to the
                 RPKI cache server.";
            }
          }
          container pdu-counters {
            config false;
            description
              "Counters of PDUs that are received from cache";
            leaf serial-notify {
              type yang:zero-based-counter64;
              description
                "Serial notify PDU count";
            }
            leaf cache-response {
              type yang:zero-based-counter64;
              description
                "Cache response PDU count";
            }
            leaf ipv4-prefix {
              type yang:zero-based-counter64;
              description
                "IPv4 prefix PDU count";
            }
            leaf ipv6-prefix {
              type yang:zero-based-counter64;
              description
                "Ipv6 prefix PDU count";
            }
            leaf end-of-data {
              type yang:zero-based-counter64;
              description
                "End of data PDU count";
            }
            leaf cache-reset {
              type yang:zero-based-counter64;
              description
                "Cache reset PDU count";
            }
            leaf reset-query {
              type yang:zero-based-counter64;
              description
                "Reset query PDU count";
            }
```

```
            leaf serial-query {
              type yang:zero-based-counter64;
              description
                "Serial query PDU count";
            }
          }
          container error-pdu-counters {
            config false;
            description
              "Counters of error PDUs that originate from router
        or cache server";
            leaf corrupt-data {
              type yang:zero-based-counter64;
              description
                "Corrupt data PDU count";
            }
            leaf internal-error {
              type yang:zero-based-counter64;
              description
                "Internal error PDU count";
            }
            leaf unsupported-protocol-version {
              type yang:zero-based-counter64;
              description
                "Unsupported protocol version PDU count";
            }
            leaf unsupported-pdu-type {
              type yang:zero-based-counter64;
              description
                "Unsupported PDU type count";
            }
            leaf unexpected-protocol-version {
              type yang:zero-based-counter64;
              description
                "Unexpected protocol version PDU count";
            }
            leaf no-data-available {
              type yang:zero-based-counter64;
              description
                "No data available PDU count";
            }
            leaf invalid-request {
              type yang:zero-based-counter64;
              description
                "Invalid request PDU count";
            }
            leaf withdrawal-unknown-record {
              type yang:zero-based-counter64;
```

```
                  description
                    "Withdrawal of unknown record PDU count";
                }
                leaf duplicate-announcement-received {
                  type yang:zero-based-counter64;
                  description
                    "Duplicate announcement received PDU count";
                }
              }
            }
          }
        }
      }
    }
```

<CODE ENDS>

4. RPKI Table YANG Module

4.1. Tree View

The complete tree of the ietf-rpki-table.yang data model is represented as following. See [RFC8340] for an explanation of the symbols used.

```
module: ietf-rpki-table

  augment /rt:routing:
    +--ro roa-tables
    |  +--ro roa-table* [name]
    |     +--ro name    string
    |     +--ro ipv4
    |     |  +--ro roas
    |     |  |  +--ro roa* [prefix max-len asn source]
    |     |  |     +--ro prefix     inet:ipv4-prefix
    |     |  |     +--ro max-len    ipv4-pfx-len
    |     |  |     +--ro asn        inet:as-number
    |     |  |     +--ro source     union
    |     |  +--ro total-records?     yang:gauge32
    |     |  +--ro records-added?     yang:counter64
    |     |  +--ro records-deleted?   yang:counter64
    |     +--ro ipv6
    |        +--ro roas
    |        |  +--ro roa* [prefix max-len asn source]
    |        |     +--ro prefix     inet:ipv6-prefix
    |        |     +--ro max-len    ipv6-pfx-len
    |        |     +--ro asn        inet:as-number
    |        |     +--ro source     union
    |        +--ro total-records?     yang:gauge32
    |        +--ro records-added?     yang:counter64
    |        +--ro records-deleted?   yang:counter64
    +--ro router-key-tables
    |  +--ro router-key-table* [name]
    |     +--ro name           string
    |     +--ro router-keys
    |        +--ro router-key* [ski asn key server-address]
    |           +--ro ski               subject-key-id
    |           +--ro asn               inet:as-number
    |           +--ro key               string
    |           +--ro server-address    inet:ip-address
    +--ro aspa-tables
       +--ro aspa-table* [name]
          +--ro name    string
          +--ro aspas* [customer-asn]
             +--ro customer-asn      inet:as-number
             +--ro server-address?   inet:ip-address
             +--ro provider-asns* [provider-asn]
                +--ro provider-asn    inet:as-number
```

4.2. Yang Module

<CODE BEGINS> file "ietf-rpki-table@2022-10-18.yang"

```
   module ietf-rpki-table {
     yang-version "1.1";
     namespace "urn:ietf:params:xml:ns:yang:ietf-rpki-table";
     prefix "rpki-table";
     import ietf-yang-types {
       prefix "yang";
       reference
         "RFC 6991: Common YANG Data Types.";
     }
     import ietf-inet-types {
       prefix "inet";
       reference
         "RFC 6991: Common YANG Data Types";
     }
     import ietf-routing {
       prefix rt;
       reference
         "RFC 8349: A YANG Data Model for Routing Management
          (NMDA Version).";
     }
     organization
       "IETF SIDROPS Working Group";
     contact
       "TBD";
     description
       "This module describes a YANG model for the Resource Public
        Key Infrastructure (RPKI) to Router Protocol configuration.
        This YANG model conforms to the Network Management
        Datastore Architecture (NMDA) as described in RFC 8342.
        Copyright (c) 2022 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 XXXX;
        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.";
     reference "RFC 8210";
     revision 2022-10-18 {
       description
         "Initial Version";
```

```
      reference
        "RFC 8210, YANG Data Model for RPKI to Router Protocol";
    }
    typedef ipv4-pfx-len {
      type uint8 {
        range "0 .. 32";
      }
      description
        "IPv4 Prefix Length.";
    }
    typedef ipv6-pfx-len {
      type uint8 {
        range "0 .. 128";
      }
      description
        "IPv6 Prefix Length.";
    }
    typedef subject-key-id {
      type binary {
        length 20;
      }
      description
        "Subject Key Identifier.";
    }
    grouping aspa-overall-records {
      description
        "ASPAs received from all RPKI cache servers.";
      list aspas {
        key "customer-asn";
        description
          "An entry of ASPA.";
        leaf customer-asn {
          type inet:as-number;
          description
            "The AS number of a customer.";
        }
        leaf server-address {
          type inet:ip-address;
          description
            "IP address of the RPKI cache server.";
        }
        list provider-asns {
          key "provider-asn";
          description
            "Providers of the customer.";
          leaf provider-asn {
            type inet:as-number;
            description
```

```
              "The AS number of a provider.";
          }
        }
      }
    }

    augment "/rt:routing" {
      description
        "RPKI tables augmentation of ietf-routing module.";
      container roa-tables {
        config false;
        description
          "List of tables containing ROAs received from all RPKI
           cache servers.";
        list roa-table {
          key "name";
          description
            "Table of ROAs received from all RPKI cache servers.";
          leaf name {
            type string;
            description
              "Name of the ROA table.";
          }
          container ipv4 {
            config false;
            description
              "Container for IPv4 ROAs table.";
            container roas {
              config false;
              description
                "ROAs received from the RPKI cache server.";
              list roa {
                key "prefix max-len asn source";
                description
                  "An entry of ROA.";
                leaf prefix {
                  type inet:ipv4-prefix;
                  description
                    "The IPv4 prefix of the ROA.";
                }
                leaf max-len {
                  type ipv4-pfx-len;
                  description
                    "Denotes the longest prefix allowed. This
                     MUST NOT be less than the prefix length.";
                }
                leaf asn {
                  type inet:as-number;
```

```
                description
                  "The origin AS number of the ROA.";
              }
              leaf source {
                type union {
                  type string;
                  type inet:ip-address;
                }
                description
                  "String representing the source of the records
                   in this record-set.";
              }
            }
          }
          leaf total-records {
            type yang:gauge32;
            description
              "Number of prefix policy records.";
          }
          leaf records-added {
            type yang:counter64;
            description
              "Number of prefix policy records cumulatively added.";
          }
          leaf records-deleted {
            type yang:counter64;
            description
              "Number of prefix policy records cumulatively
               deleted.";
          }
        }
        container ipv6 {
          config false;
          description
            "Container for IPv6 ROAs table.";
          container roas {
            config false;
            description
              "ROAs received from the RPKI cache server.";
            list roa {
              key "prefix max-len asn source";
              description
                "An entry of ROA.";
              leaf prefix {
                type inet:ipv6-prefix;
                description
                  "The IPv6 prefix of the ROA.";
              }
```

```
                leaf max-len {
                  type ipv6-pfx-len;
                  description
                    "Denotes the longest prefix allowed. This
                     MUST NOT be less than the prefix length.";
                }
                leaf asn {
                  type inet:as-number;
                  description
                    "The origin AS number of the ROA.";
                }
                leaf source {
                  type union {
                    type string;
                    type inet:ip-address;
                  }
                  description
                    "Representing the source of the records in this
                     record-set. Either a server IP or a source file
                     of static records.";
                }
              }
            }
            leaf total-records {
              type yang:gauge32;
              description
                "Number of prefix policy records.";
            }
            leaf records-added {
              type yang:counter64;
              description
                "Number of prefix policy records cumulatively added.";
            }
            leaf records-deleted {
              type yang:counter64;
              description
                "Number of prefix policy records cumulatively
                 deleted.";
            }
          }
        }
      }
      container router-key-tables {
        config false;
        description
          "List of router key table received from all RPKI cache
           servers.";
        list router-key-table {
```

```
          key "name";
          description
            "Table of router keys received from all RPKI cache
             servers.";
          leaf name {
            type string;
            description
              "Name of the router-key-table.";
          }
          container router-keys {
            config false;
            description
              "Router keys received from the RPKI cache server.";
            list router-key {
              key "ski asn key server-address";
              description
                "An entry of router key.";
              leaf ski {
                type subject-key-id;
                description
                  "A router key's Subject Key Identifier (SKI).";
                reference
                  "RFC 6487: A Profile for X.509 PKIX Resource
                   Certificates";
              }
              leaf asn {
                type inet:as-number;
                description
                  "The AS number of the router which the key
                   belongs to.";
              }
              leaf key {
                type string;
                description
                  "A router key's subjectPublicKeyInfo value.";
                reference
                  "RFC 8608: BGPsec Algorithms, Key Formats, and
                   Signature Formats";
              }
              leaf server-address {
                type inet:ip-address;
                description
                  "IP address of the RPKI cache server.";
              }
            }
          }
        }
      }
```

```
      container aspa-tables {
        config false;
        description
          "List of tables of ASPAs received from all RPKI cache
           servers.";
       list aspa-table {
          key "name";
          description
            "Table of ASPAs received from all RPKI cache servers.";
          leaf name {
            type string;
            description
              "Name of the ASPA-table.";
          }
          uses aspa-overall-records;
        }
      }
    }
  }
```

<CODE ENDS>

5. Security Considerations

This section is modeled after the template described in Section 3.7 of [YANG-GUIDE].

The "ietf-rpki-rtr.yang" YANG module and "ietf-rpki-table.yang" YANG module define data models that are designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and have to use mutual authentication.

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.

There are a number of data nodes defined in these YANG modules that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes can 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. Specifically, the following subtrees and data nodes have particular sensitivities/vulnerabilities:

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:server-address

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:server-port

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:local-address

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:local-port

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:enabled

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:preference

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:description

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:enable-authentication

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:authentication

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:roa-limit

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:aspa-limit

Some of the readable data nodes in these YANG modules 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. Specifically, the following subtrees and data nodes have particular sensitivities/vulnerabilities:

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:session-state

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:statistics

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:connection-data

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:protocol-data

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:pdu-counters

rpki-rtr:rpki-rtr/rpki-rtr:sessions/rpki-rtr:session/rpki-rtr:error-pdu-counters

rt:routing/rpki-table:roa-tables/rpki-table:roa-table

rt:routing/rpki-table:router-key-tables/rpki-table:router-key-table

rt:routing/rpki-table:aspa-tables/rpki-table:aspa-table

There are no particularly sensitive RPC or action operations.

6. IANA Considerations

6.1. RPKI to Router YANG Module Registry

IANA is requested to register the following URI in the "ns" registry within the "IETF XML Registry" group ([RFC3688]):

URI: urn:ietf:params:xml:ns:yang:ietf-rpki-rtr

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace

IANA is requested to register the following YANG modules in the "YANG Module Names" registry ([RFC6020]) within the "YANG Parameters" registry group.

name: ietf-rpki-rtr

Maintained by IANA? N

namespace: urn:ietf:params:xml:ns:yang:ietf-rpki-rtr

prefix: rpki-rtr

reference: RFC XXXX

6.2. RPKI Table YANG Module Registry

IANA is requested to register the following URI in the "ns" registry within the "IETF XML Registry" group ([RFC3688]):

URI: urn:ietf:params:xml:ns:yang:ietf-rpki-table

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace

IANA is requested to register the following YANG module in the "YANG Module Names" registry ([RFC6020]) within the "YANG Parameters" registry group.

name: ietf-rpki-table

Maintained by IANA? N

namespace: urn:ietf:params:xml:ns:yang:ietf-rpki-table

prefix: rpki-table

reference: RFC XXXX

7. References

7.1. Normative References

[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>.

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <https://www.rfc-editor.org/info/rfc6810>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <https://www.rfc-editor.org/info/rfc6811>.

[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>.

[RFC8210] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", RFC 8210, DOI 10.17487/RFC8210, September 2017, <https://www.rfc-editor.org/info/rfc8210>.

[I-D.ietf-sidrops-8210bis] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 2", Work in Progress, Internet-Draft, draft-ietf-sidrops-8210bis-21, 21 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-8210bis-21>.

7.2. Informative References

[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>.

[YANG-GUIDE] Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>.

Contributors

Mengxiao Chen
H3C
China
Email: chen.mengxiao@h3c.com

Authors' Addresses

Yisong Liu
China Mobile
China
Email: liuyisong@chinamobile.com

Changwang Lin
New H3C Technologies
China
Email: linchangwang.04414@h3c.com

Haibo Wang
Huawei Technologies
China
Email: rainsword.wang@huawei.com

Jishnu Roy
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: jishnur@juniper.net

Jeffrey Haas
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: jhaas@juniper.net

Hongwei Liu
ZTE Corporation
China
Email: liu.hongwei3@zte.com.cn

Di Ma
ZDNS
Floor 21, Block B, Greenland Center
Chaoyang Beijing, 100102
China
Email: madi@zdns.cn