Yang Data Model for Internet Protocol Security (IPSec)
draft-tran-ipecme-yang-ipsec-00.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Internet Protocol Security (IPSec).

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on November 14, 2015.

Copyright Notice

Copyright (c) 2015 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 (http://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
2. Conventions used in this document............................3
3. IPSec Configuration and Operation Model Overview............4
   3.1. IPSec Configuration Data Model............................4
   3.2. IKE Configuration Data Model..............................8
   3.3. IKEv2 Configuration Data Model............................9
   3.4. IPSec Operation Data Model...............................11
   3.5. IKE Operation Data Model.................................12
   3.6. IKEv2 Operation Data Model...............................13
   3.7. RPC Operation............................................13
4. IPSec YANG Module.............................................14
5. Security Considerations.......................................57
6. References....................................................57
   6.1. Normative References....................................57
   6.2. Informative References..................................58
1. Introduction

Internet Protocol Security (IPSec) is a suite of protocols that provides security to internet communications at the IP layer. This document defines a YANG data model that can be used to configure and manage the IPSec protocol.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.
3. IPSec Configuration and Operation Model Overview

Figure 1 illustrates the IPSec configuration and operation model which contains IPSec, IKE, and IKEv2 modules.

![Figure 1. Overview of IPSec configuration and operation model structure](image)

3.1. IPSec Configuration Data Model

The IPSec data model provides the appropriate leaves for configuring the IPSec protocol. The IPSec YANG data model has the following structure:

```Yang
module: ietf-ipsec
  +--rw ipsec
    |  +--rw access-list* [name sequence-number]
    |     |  +--rw name                        string
    |     |  +--rw description?                string
    |     |  +--rw sequence-number             uint32
    |     |  +--:(protocol)?
    |     |     |  +--:(number)
    |     |     |     |  +--rw number?                     uint16
    |     |     |     |  +--rw (argument)?
    |     |     |     |     |  +--:(source-ipv4-address)
    |     |     |     |     |     |  +--rw source-ipv4-address? inet:ipv4-address
    |     |     |     |     |     |  +--:(any)
    |     |     |     |     |     |     |  +--rw source-any? empty
    |     |     |     |     |  +--:(source-ipv4-address)
    |     |     |     |     |     |  +--rw source-ipv4-address? inet:ipv4-address
    |     |     |     |     |  +--:(any)
    |     |     |     |     |     |  +--rw any? empty
    |     |     |     |     |     |  +--:(tcp)
    |     |     |     |     |     |     |  +--rw tcp? empty
```

| +--:(udp) |
|    +--rw udp?                        empty |
|    +--rw (dest-address)? |
|        +--:(dest-ipv4-address) |
|            +--rw destination-ipv4-address? inet:ipv4-address |
|            +--:(dest-any) |
|            +--rw dest-any?          empty |
|    +--rw alarms |
|        +--rw hold-down?  uint8 |
|    +--rw qos |
|        +--rw policy* [name] |
|            +--rw name  string |
|            +--rw pq |
|                +--rw num-queues?  uint8 |
|    +--rw redundancy |
|        +--rw inter-chassis?  empty |
|    +--rw security-association |
|        +--rw ipsec-sa* [name] |
|            +--rw name  string |
|            +--rw anti-replay-window?  uint16 |
|            +--rw ip-comp?         empty |
|            +--rw in |
|                +--rw ah |
|                    +--rw spi?    uint32 |
|                    +--rw description?  string |
|                    +--rw (authentication-algorithm)? |
|                        +--:(hmac-aes-xcbc) |
|                            +--rw hmac-aes-xcbc |
|                                +--rw key-str?  union |
|                        +--:(hmac-md5-96) |
|                            +--rw hmac-md5-96 |
|                                +--rw key-str?  union |
|                        +--:(hmac-shal-96) |
|                            +--rw hmac-shal-96 |
|                                +--rw key-str?  union |
|                        +--:(key-string) |
|                            +--rw key-string |
|                                +--rw key-str?  union |
|    +--rw esp |
|        +--rw description?  string |
|        +--rw authentication |
|            +--rw (authentication-algorithm)? |
|                +--:(hmac-aes-xcbc) |
|                    +--rw hmac-aes-xcbc |
|                        +--rw key-str?  union |
|                +--:(hmac-md5-96) |
|                    +--rw hmac-md5-96 |
|                        +--rw key-str?  union |
|                +--:(hmac-shal-96) |
|                    +--rw hmac-shal-96 |
|                        +--rw key-str?  union |
|                +--:(key-string) |
|                    +--rw key-string |
|                        +--rw key-str?  union |
---:(hmac-md5-96)
  ---rw hmac-md5-96
  ---rw key-str?   union
---:(hmac-sha1-96)
  ---rw hmac-sha1-96
  ---rw key-str?   union
---:(key-string)
  ---rw key-string
  ---rw key-str?   union
---rw encryption
  ---rw (encryption-algorithm)?
    ---:(des3-cbc)
      ---rw des3-cbd
      ---rw key-str?   union
    ---:(aes-128-cbc)
      ---rw aes-128-cbc
      ---rw key-str?   union
    ---:(aes-192-cbc)
      ---rw aes-192-cbc
      ---rw key-str?   union
    ---:(aes-256-cbc)
      ---rw aes-256-cbc
      ---rw key-str?   union
    ---:(des-cbc)
      ---rw des-cbc
      ---rw key-str?   union
    ---:(key-string)
      ---rw key-string
      ---rw key-str?   union
---rw proposal
  ---rw ipsec-proposal* [name]
    ---rw name   string
    ---rw ah?     ike-integrity-algorithm-t
    ---rw esp
      ---rw authentication?     ike-integrity-algorithm-t
      ---rw encryption?         ike-encryption-algorithm-t
    ---rw ip-comp?   empty
    ---rw lifetime
      ---rw kbytes?   uint32
      ---rw seconds?  uint32
---rw policy
  ---rw ipsec-policy* [name]
    ---rw name   string
    ---rw description?  string
    ---rw anti-replay-window?   uint32
    ---rw perfect-forward-secrecy
      ---rw dh-group?    diffie-hellman-group-t
    ---rw seq* [seq-id]
      ---rw seq-id   uint32
### 3.2. IKE Configuration Data Model

The IKE data model provides the appropriate leaves for configuring the IKE protocol. The IKE YANG data model has the following structure:

```
+++rw ike
    ++rw proposal* [name]
        | ++rw name       string
        | ++rw description?   string
        | ++rw dh-group    diffie-hellman-group-t
        | ++rw encryption
            | ++rw algorithm?   ike-encryption-algorithm-t
            | ++rw lifetime    uint32
        | ++rw authentication
            | ++rw algorithm?   ike-integrity-algorithm-t
            | ++rw preshared-key?   empty
            | ++rw rsa-signature? empty
        | ++rw keepalive?   empty
    ++rw policy* [name]
        | ++rw name       string
        | ++rw mode
            | ++rw aggressive?   empty
            | ++rw main?         empty
        | ++rw connection-type   connection-type-t
        | ++rw pre-shared-key?   union
        | ++rw validate-certificate-identity?   empty
        | ++rw seq* [seq-id]
            | ++rw seq-id      uint32
            | ++rw proposal?   leafref
        | ++rw identity
            | ++rw local
                | ++rw (identity)?
                | | ++rw ipv4-address?   inet:ipv4-address
                | | ++rw ipv6-address?   inet:ipv6-address
                | | ++rw fqn-string?   inet:domain-name
                | | ++rw rfc822-address-string?   string
                | | ++rw rfc822-address-string?   string
                | | ++rw dnX509?   string
            | ++rw remote
```
3.3. IKEv2 Configuration Data Model

The IKEv2 data model provides the appropriate leaves for configuring the IKEv2 protocol. The IKEv2 YANG data model has the following structure:

```yang
++-rw ikev2
  ++-rw proposal* [name]  
    |  ++-rw name                      string
    |  ++-rw description?              string
    |  ++-rw dh-group                  diffie-hellman-group-t
    |  ++-rw encryption
    |    |  ++-rw algorithm?   ike-encryption-algorithm-t
    |    |  ++-rw pseudo-random-function    pseudo-random-function-t
    |    |  ++-rw authentication
    |    |    |  ++-rw algorithm?   ike-integrity-algorithm-t
    |  ++-rw policy* [name]
    |    |  ++-rw name                      string
    |    |  ++-rw authentication
    |    |    |  ++-rw preshared-key?   empty
    |    |    |  ++-rw rsa-signature?   empty
    |    |  ++-rw lifetime                  uint32
    |    |  ++-rw address-allocation
    |    |    |  ++-rw aaa?   empty
    |    |  ++-rw connection-type            connection-type-t
    |    |  ++-rw pre-shared-key?           union
    |    |  ++-rw validate-certificate-identity?   empty
    |    |  ++-rw seq* [seq-id]
    |    |    |  ++-rw seq-id                  uint32
    |    |    |  ++-rw proposal?   leafref
    |    |  ++-rw identity
    |    |    |  ++-rw local
```

+-:(ipv4-address)
  |   +--rw ipv4-address?       inet:ipv4-address
+-:(ipv6-address)
  |   +--rw ipv6-address?       inet:ipv6-address
+-:(fqdn-string)
  |   +--rw fqdn-string?       inet:domain-name
+-:(rfc822-address-string)
  |   +--rw rfc822-address-string?   string
  +--:(dnX509)
     |   +--rw dnX509?       string
+-rw remote
  +--rw (identity)?
     |   +--:(ipv4-address)
     |       +--rw ipv4-address?       inet:ipv4-address
     |   +--:(ipv6-address)
     |       +--rw ipv6-address?       inet:ipv6-address
     |   +--:(fqdn-string)
     |       +--rw fqdn-string?       inet:domain-name
     +--:(rfc822-address-string)
        |   +--rw rfc822-address-string?   string
        |   +--:(dnX509)
        |       +--rw dnX509?       string
        +-rw description?       string
3.4. IPSec Operation Data Model

The IPSec data model provides the appropriate leaves for operational states of the IPSec protocol. The IPSec YANG data model has the following structure:

```
+--ro ipsec-state
  +--ro policy*
    |  +--ro name?                      string
    |  +--ro anti-replay-window?        uint32
    |  +--ro perfect-forward-secrecy?   diffie-hellman-group-t
    |  +--ro seq*
    |     +--ro seq-id?          uint32
    |     +--ro proposal-name? string
  +--ro proposal*
    |  +--ro name?       string
    |  +--ro ah?         ike-integrity-algorithm-t
    |  +--ro esp
    |     +--ro authentication?   ike-integrity-algorithm-t
    |     +--ro encryption?       ike-encryption-algorithm-t
    |  +--ro ip-comp?    empty
    |  +--ro lifetime
    |     +--ro kbytes?    uint32
    |     +--ro seconds?   uint32
  +--ro hold-down?   uint32
+--ro sa*
  +--ro name?                       string
  +--ro anti-replay-window?         uint16
  +--ro ip-comp?                    empty
  +--ro spi?                        uint32
  +--ro description?                string
  +--ro authentication-algorithm?   ike-integrity-algorithm-t
  +--ro encryption-algorithm?       ike-encryption-algorithm-t
```
3.5. IKE Operation Data Model

The IKE data model provides the appropriate leaves for operational states of the IKE protocol. The IKE YANG data model has the following structure:

```
+-ro ike-state
  |  +--ro proposal*
  |  |  +--ro name?    string
  |  |  +--ro lifetime? uint32
  |  |  +--ro encryption? ike-encryption-algorithm-t
  |  |  +--ro dh-group? diffie-hellman-group-t
  |  |  +--ro authentication? ike-integrity-algorithm-t
  |  +--ro policy*
  |     |  +--ro name?    string
  |     |  +--ro description? string
  |     |  +--ro mode?    enumeration
  |     |  +--ro connection-type? connection-type-t
  |     |  +--ro local-identity? inet:ipv4-address-no-zone
  |     |  +--ro remote-identity? inet:ipv4-address-no-zone
  |     |  +--ro pre-shared-key? string
  |     |  +--ro seq?    uint32
  |     +--ro proposal? string
```
3.6. IKEv2 Operation Data Model

The IKEv2 data model provides the appropriate leaves for operational states of the IKEv2 protocol. The IKEv2 YANG data model has the following structure:

```
---ro ikev2-state
  +---ro proposal*
    |   +---ro name?         string
    |   +---ro pseudo-random-function?  pseudo-random-function-t
    |   +---ro authentication?   ike-integrity-algorithm-t
    |   +---ro encryption?      ike-encryption-algorithm-t
    |   +---ro dh-group        diffie-hellman-group-t
    +---ro policy*
      |   +---ro name?         string
      |   +---ro description?   string
      |   +---ro mode?          enumeration
      |   +---ro connection-type? connection-type-t
      |   +---ro local-identity?  inet:ipv4-address-no-zone
      |   +---ro remote-identity?  inet:ipv4-address-no-zone
      |   +---ro pre-shared-key?   string
      |   +---ro seq?            uint32
      |   +---ro proposal?       string
```

3.7. RPC Operation

This section defines a list of RPC support for IPSec protocol.

```
rpcs:
  +---x clear-ipsec-group
    |   +---ro input
    |     +---ro alarm-hold-down? uint8
    |     +---ro ipsec-policy-name? leafref
  +---x clear-ike-group
    |   +---ro input
    |     +---ro proposal? leafref
  +---x clear-ikev2-group
    |   +---ro input
    |     +---ro proposal? leafref
```
4. IPSec YANG Module

```yaml
<CODE BEGINS> file "ietf-ipsec@2015-04-22.yang"

module ietf-ipsec {
    namespace "urn:ietf:params:xml:ns:yang:ietf-ipsec";
    prefix "eipsec";

    import ietf-inet-types {
        prefix inet;
    }

    import ietf-yang-types {
        prefix yang;
    }

    organization "Ericsson AB."
    contact "Web: <http://www.ericsson.com>";

    description "This YANG module defines the configuration and operational
    state data for Internet Protocol Security (IPSec) on
    IETF draft.
    Copyright (c) 2015 Ericsson AB.
    All rights reserved."

    revision 2015-04-22 {
        description "Initial revision.";
        reference "YANG Data model for Internet Protocol Security - IPSec";
    }

    typedef authentication-method-t {
        type enumeration {
            enum psk {
                value 0;
                description "Pre-Sharing Keys.";
            }
            enum certificate {
                value 1;
                description "Certificate.";
            }
            enum rsa {
                value 2;
                description "RSA.";
            }
        }
    }
```
typedef ikev2-exchange-type-t {
  type enumeration {
    enum ikev2-et-ike-sa-init {
      value 34;
      description
        "ikev2-et-ike-sa-init - RFC 7296.";
    }
    enum ikev2-et-ike-auth {
      value 35;
      description
        "ikev2-et-ike-auth - RFC 7296.";
    }
    enum ikev2-et-create-child-sa {
      value 36;
      description
        "ikev2-et-create-child-sa - RFC 7296.";
    }
    enum ikev2-et-informational {
      value 37;
      description
        "ikev2-et-informational - RFC 7296.";
    }
    enum ikev2-et-ike-session-resume {
      value 38;
      description
        "ikev2-et-ike-session-resume - RFC 7296.";
    }
    enum ikev2-et-gsa-auth {
      value 39;
      description
        "ikev2-et-gsa-auth - RFC 7296.";
    }
    enum ikev2-et-gsa-registration {
      value 40;
      description
        "ikev2-et-gsa-registration - RFC 7296.";
    }
    enum ikev2-et-gsa-rekey {
      value 41;
      description
        "Certificate.";
    }
  }
}

description
  "Available authentication methods.";
}
description
"ikev2-et-gsa-rekey - RFC 7296."
}
description
"IKEv2 Exchange Types (ET)."
}

typedef transform-type-value-t {
type enumeration {
    enum ttv-reserved-0 {
        value 0;
description
"ttv-reserved-0 - Transform Type Value Reserved " +
"(RFC 7296).";
    }
    enum ttv-encr {
        value 1;
description
"ttv-encr - Transform Type Value 1, Encryption Algorithm " +
"(ENCR) used in IKE and ESP."
    }
    enum ttv-prf {
        value 2;
description
"ttv-prf - Transform Type Value 2, " +
"Pseudo-Random Function (PRF) used in IKE."
    }
    enum ttv-integ {
        value 3;
description
"ttv-integ - Transform Type Value 3, Integrity Algorithm" +
"(INTEG) used in IKE, AH, optional ESP."
    }
    enum ttv-dh {
        value 4;
description
"ttv-dh - Transform Type Value 4, Diffie-Hellman (DH) " +
"used in IKE, optional AH and ESP."
    }
    enum ttv-esn {
        value 5;
description
"ttv-esn - Transform Type Value 5, Extended Sequence " +
"Numbers (ESN) used in AH and ESP."
    }
}
description
"Transform Type Values (RFC 7296).";
}

/* IKEv2 Transform Attribute Types (TAT) */
typedef ikev2-transform-attribute-type-t {
type enumeration {
enum ikev2-tat-reserved-0 {
  value 0;
description
  "ikev2-tat-reserved-0 - IKEv2 Transform Attribute Type Reserved-0 (RFC 7296).";
}
enum ikev2-tat-reserved-1 {
  value 1;
description
  "ikev2-tat-reserved-1 - IKEv2 Transform Attribute Type Reserved-1 (RFC 7296).";
}
enum ikev2-tat-reserved-13 {
  value 13;
description
  "ikev2-tat-reserved-13 - IKEv2 Transform Attribute Type Reserved-13 (RFC 7296).";
}
enum ikev2-tat-key-length {
  value 41;
description
  "ikev2-tat-key-length - IKEv2 Transform Attribute Type KEY LENGTH (in bits) (RFC 7296).";
}
} description
"IKEv2 Transform Attribute Types (TAT) (RFC 7296).";
}

/* Transform Type 1 (Encryption Algorithm Transform IDs) */
typedef ike-encryption-algorithm-t {
type enumeration {
enum encr-reserved-0 {
  value 0;
description
  "encr-reserved-0 --> RFC_5996.";
}
enum encr-des-iv4 {
  value 1;
description
  "encr-des-iv4 --> RFC_5996.";
}
}
enum encr-des {
    value 2;
    description
    "encr-des --> RFC_5996.";
}
enum encr-3des {
    value 3;
    description
    "encr-3des --> RFC_5996.";
}
enum encr-rc5 {
    value 4;
    description
    "encr-rc5 --> RFC_5996.";
}
enum encr-idea {
    value 5;
    description
    "encr-idea --> RFC_5996.";
}
enum encr-cast {
    value 6;
    description
    "encr-cast --> RFC_5996.";
}
enum encr-blowfish {
    value 7;
    description
    "encr-blowfish --> RFC_5996.";
}
enum encr-3idea {
    value 8;
    description
    "encr-3idea --> RFC_5996.";
}
enum encr-des-iv32 {
    value 9;
    description
    "encr-des-iv32 --> RFC_5996.";
}
enum encr-reserved-10 {
    value 10;
    description
    "encr-reserved-10 --> RFC_5996.";
}
enum encr-null {
    value 11;
    description
    "encr-null --> RFC_5996.";
enum encr-aes-cbc {
    value 12;
    description
    "encr-aes-cbc --> RFC_5996.";
}

enum encr-aes-ctr {
    value 13;
    description
    "encr-aes-ctr --> RFC_5996.";
}

enum encr-aes-ccm-8 {
    value 14;
    description
    "encr-aes-ccm-8 --> RFC_5996.";
}

enum encr-aes-ccm-12 {
    value 15;
    description
    "encr-aes-ccm-12 --> RFC_5996.";
}

enum encr-aes-ccm-16 {
    value 16;
    description
    "encr-aes-ccm-16 --> RFC_5996.";
}

enum encr-reserved-17 {
    value 17;
    description
    "encr-reserved-17 --> RFC_5996.";
}

enum encr-aes-gcm-8-icv {
    value 18;
    description
    "encr-aes-gcm-8-icv --> RFC_5996.";
}

enum encr-aes-gcm-12-icv {
    value 19;
    description
    "encr-aes-gcm-12-icv --> RFC_5996.";
}

enum encr-aes-gcm-16-icv {
    value 20;
    description
    "encr-aes-gcm-16-icv --> RFC_5996.";
}

enum encr-null-auth-aes-gmac {
    value 21;
    description
}
"encr-null-auth-aes-gmac --> RFC_5996."
}
enum encr-ieee-p1619-xts-aes {
    value 22;
    description
    "encr-ieee-p1619-xts-aes --> Reserved for " +
    "IEEE P1619 XTS-AES."
}
enum encr-camellia-cbc {
    value 23;
    description
    "encr-camellia-cbc --> RFC_5996."
}
enum encr-camellia-ctr {
    value 24;
    description
    "encr-camellia-ctr --> RFC_5996."
}
enum encr-camellia-ccm-8-icv {
    value 25;
    description
    "encr-camellia-ccm-8-icv --> RFC_5996."
}
enum encr-camellia-ccm-12-icv {
    value 26;
    description
    "encr-camellia-ccm-12-icv --> RFC_5996."
}
enum encr-camellia-ccm-16-icv {
    value 27;
    description
    "encr-camellia-ccm-16-icv --> RFC_5996."
}
enum encr-aes-cbc-128 {
    value 1024;
    description
    "encr-aes-cbc-128 --> RFC_5996."
}
enum encr-aes-cbc-192 {
    value 1025;
    description
    "encr-aes-cbc-192 --> RFC_5996."
}
enum encr-aes-cbc-256 {
    value 1026;
    description
    "encr-aes-cbc-256 --> RFC_5996."
}
enum encr-blowfish-128 {
enum encr-blowfish-128 {
  value 1027;
  description
  "encr-blowfish-128 --> RFC_5996.";
}
enum encr-blowfish-192 {
  value 1028;
  description
  "encr-blowfish-192 --> RFC_5996.";
}
enum encr-blowfish-256 {
  value 1029;
  description
  "encr-blowfish-256 --> RFC_5996.";
}
enum encr-blowfish-448 {
  value 1030;
  description
  "encr-blowfish-448 --> RFC_5996.";
}
enum encr-camellia-128 {
  value 1031;
  description
  "encr-camellia-128 --> RFC_5996.";
}
enum encr-camellia-192 {
  value 1032;
  description
  "encr-camellia-192 --> RFC_5996.";
}
enum encr-camellia-256 {
  value 1033;
  description
  "encr-camellia-256 --> RFC_5996.";
}
);

description
"Transform Type 1 - Internet Key Exchange (IKE) " +
"encryption algorithms.";

} /* Transform Type 2 (Pseudo-Random Function PRF) */
typedef pseudo-random-function-t {
  type enumeration {
    enum prf-reserved-0 {
      value 0;
      description
      "prf-reserved-0 --> RFC_2104.";
    }
    enum prf-hmac-md5 {

value 1;

description
"prf-hmac-md5 --> RFC_2104.";
}
enum prf-hmac-shal {
  value 2;
  description
  "prf-hmac-shal --> RFC2104.";
}
enum prf-hmac-tiger {
  value 3;
  description
  "prf-hmac-tiger --> RFC2104.";
}
enum prf-aes128-xcbc {
  value 4;
  description
  "prf-aes128-xcbc --> RFC4434.";
}
enum prf-hmac-sha2-256 {
  value 5;
  description
  "prf-hmac-sha2-256 --> RFC4434.";
}
enum prf-hmac-sha2-384 {
  value 6;
  description
  "prf-hmac-sha2-384 --> RFC4434.";
}
enum prf-hmac-sha2-512 {
  value 7;
  description
  "prf-hmac-sha2-512 --> RFC4434.";
}
enum prf-aes128-cmac {
  value 8;
  description
  "prf-aes128-cmac --> RFC4615.";
}
}

description
"Available Pseudo-Random Functions (PRF).";
}

/* Transform Type 3 (Integrity Algorithm) */
typedef ike-integrity-algorithm-t {
  type enumeration {
    enum auth-none {
      value 0;
      
description
"auth-none --> RFC_5996."
}
enum auth-hmac-md5-96 {
  value 1;
description
  "auth-hmac-md5-96 --> RFC_5996."
}
enum auth-hmac-sha1-96 {
  value 2;
description
  "auth-hmac-sha1-96 --> RFC_5996."
}
enum auth-des-mac {
  value 3;
description
  "auth-des-mac --> RFC_5996."
}
enum auth-kpdk-md5 {
  value 4;
description
  "auth-kpdk-md5 --> RFC_5996."
}
enum auth-aes-xcbc-96 {
  value 5;
description
  "auth-aes-xcbc-96 --> RFC_5996."
}
enum auth-hmac-md5-128 {
  value 6;
description
  "auth-hmac-md5-128 --> RFC_5996."
}
enum auth-hmac-sha1-160 {
  value 7;
description
  "auth-hmac-sha1-160 --> RFC_5996."
}
enum auth-aes-cmac-96 {
  value 8;
description
  "auth-aes-cmac-96 --> RFC_5996."
}
enum auth-aes-128-gmac {
  value 9;
description
  "auth-aes-128-gmac --> RFC_5996."
}
enum auth-aes-192-gmac {
value 10;
  description
 "auth-aes-192-gmac --> RFC_5996.";
}
enum auth-aes-256-gmac {
  value 11;
  description
 "auth-aes-256-gmac --> RFC_5996.";
}
enum auth-hmac-sha2-256-128 {
  value 12;
  description
 "auth-hmac-sha2-256-128 --> RFC_5996.";
}
enum auth-hmac-sha2-384-192 {
  value 13;
  description
 "auth-hmac-sha2-384-192 --> RFC_5996.";
}
enum auth-hmac-sha2-512-256 {
  value 14;
  description
 "auth-hmac-sha2-512-256 --> RFC_5996.";
}
enum auth-hmac-sha2-256-96 {
  value 1024;
  description
 "auth-hmac-sha2-256-96.";
}
}
description
 "Transform Type 3 - Internet Key Exchange (IKE) " +
 "Integrity Algorithms.";
}

/* Transform Type 4 (Diffie-Hellman Group) */
typedef diffie-hellman-group-t {
  type enumeration {
    enum group-none {
      value 0;
      description
 "group-none --> RFC_5996.";
    }
    enum modp-768-group-1 {
      value 1;
      description
 "modp-768-group-1 --> RFC_5996.";
    }
    enum modp-1024-group-2 {
value 2;
description
"modp-1024-group-2 --> RFC_5996.";
}
enum modp-1536-group-5 {
  value 5;
description
"modp-1536-group-5 --> RFC_3526.";
}
enum modp-2048-group-14 {
  value 14;
description
"modp-2048-group-14 --> RFC_3526.";
}
enum modp-3072-group-15 {
  value 15;
description
"modp-3072-group-15 --> RFC_3526.";
}
enum modp-4096-group-16 {
  value 16;
description
"modp-4096-group-16 --> RFC_3526.";
}
enum modp-6144-group-17 {
  value 17;
description
"modp-6144-group-17 --> RFC_3526.";
}
enum modp-8192-group-18 {
  value 18;
description
"modp-8192-group-18 --> RFC_3526.";
}
enum recp-256-group-19 {
  value 19;
description
"recep-256-group-19 --> RFC_6989. 256-bit"+
" Random ECP Group.";
}
enum recp-384-group-20 {
  value 20;
description
"recep-384-group-20 --> RFC_6989. 384-bit"+
" Random ECP Group.";
}
enum recp-521-group-21 {
  value 21;
description
"recp-521-group-21 --> RFC_6989. 521-bit"
" Random ECP Group."
}
enum modp-1024-160-pos-group-22 {
  value 22;
description
  "modp-1024-160-pos-group-22 --> RFC_6989."
  " 1024-bit MODP Group with"
  " 160-bit Prime Order Subgroup (POS)."
}
enum modp-2048-224-pos-group-23 {
  value 23;
description
  "modp-2048-224-pos-group-23 --> RFC_6989."
  " 2048-bit MODP Group with"
  " 224-bit Prime Order Subgroup (POS)."
}
enum modp-2048-256-pos-group-24 {
  value 24;
description
  "modp-2048-256-pos-group-24 --> RFC_6989."
  " 2048-bit MODP Group with"
  " 256-bit Prime Order Subgroup (POS)."
}
enum recp-192-group-25 {
  value 25;
description
  "recp-192-group-25 --> RFC_6989."
  " 192-bit Random ECP Group."
}
enum recp-224-group-26 {
  value 26;
description
  "recp-224-group-26 --> RFC_6989."
  " 224-bit Random ECP Group."
}

description
  "Diffie-Hellman Groups (RFC 5996)."
}

/* Transform Type 5 (Extended Sequence Numbers
Transform ESN IDs) */
typedef extended-sequence-number-t {
type enumeration {
enum esn-none {
  value 0;
description

"esn-none - Extended Sequence Number None --> RFC_7296."
}
enum esn-1 {
  value 1;
  description
  "esn-1 - Extended Sequence Number --> RFC_7296."
}
}
description
"Extended Sequence Number (RFC 7296)."
}
typedef connection-type-t {
  type enumeration {
    enum initiator-only {
      value 0;
      description
      "initiator-only: ME will act as initiator for"+
      " bringing up IKEv2"+
      " session with its IKE peer."
    }
    enum responder-only {
      value 1;
      description
      "responder-only: ME will act as responder for"+
      " bringing up IKEv2"+
      " session with its IKE peer."
    }
    enum both {
      value 2;
      description
      "both: ME can act as initiator or responder."
    }
  }
  description
  "Connection type for IKE session."
}
typedef transport-protocol-name-t {
  type enumeration {
    enum tcp {
      value 1;
      description
    }
    enum udp {
      value 2;
      description
"User Datagram Protocol (UDP) Transport Protocol";

} enum sctp {
    value 3;
    description
    "Stream Control Transmission Protocol (SCTP) Transport "+
    "Protocol";
}

} enum icmp {
    value 4;
    description
    "Internet Control Message Protocol (ICMP) Transport "+
    "Protocol";
}

} description
"Enumeration of well known transport protocols.";

}

typedef preshared-key-t {
    type string;
    description
    "Derived string used as Pre-Shared Key.";
}


/*--------------------*/
/*   grouping */
/*--------------------*/

/* The following groupings are used in both configuration data */
and operational state data */

/*--------------------*/

grouping name-grouping {
    description
    "This grouping provides a leaf identifying the name.";
    leaf name {
        type string;
        description
        "Name of a identifying.";
    }
    leaf description {
        type string;
        description
        "Specify the description.";
    }
}

grouping sequence-number-grouping {

description
  "This grouping provides a leaf identifying a sequence number."
leaf sequence-number {
  type uint32 {
    range "1..4294967295"
  }
  description
  "Specify the sequence number."
}


grouping description-grouping {
  description
  "description for free use."
  leaf description {
    type string;
    description
    "description for free use."
  }
}

grouping traffic-selector-grouping {
  description
  "Traffic selector to be used for SA negotiation."
  leaf traffic-selector-id {
    type string;
    mandatory true;
    description
    "Traffic selector identifier."
  }
  leaf protocol-name {
    type transport-protocol-name-t;
    description
    "Specifies the protocol selector."
  }
  leaf address-range {
    type string;
    mandatory true;
    description
    "Specifies the IPv4 or IPv6 address range."
  }
}

grouping ike-general-proposal-grouping {
  description
  "IKE proposal."
  leaf name {

type string;
mandatory true;
description
"IKE Proposal identify.";
}
leaf description {
type string;
description
"Specify the description.";
}
leaf dh-group {
type diffie-hellman-group-t;
mandatory true;
description
"Specifies a Diffie-Hellman group.";
}
container encryption {
description
"Specify IKE Proposal encryption configuration";
leaf algorithm {
type ike-encryption-algorithm-t;
description
"Specifies an Encryption Algorithm.";
}
}
}
grouping ike-proposal-grouping {
description
"Configure the IKE Proposal";
uses ike-general-proposal-grouping;
leaf lifetime {
type uint32;
mandatory true;
description
"Configure lifetime for IKE SAs
0: for no timeout.
300 .. 99999999: IKE SA lifetime in seconds.";
}
container authentication {
description
"Specify IKE Proposal authentication configuration";
leaf algorithm {
type ike-integrity-algorithm-t;
description
"Specify the authentication algorithm";
}
leaf preshared-key {
  type empty;
  description
    "Use pre-shared key based authentication";
}
leaf rsa-signature {
  type empty;
  description
    "Use signature based authentication by using
    PKI certificates";
}


grouping ikev2-proposal-grouping {
  description
    "Holds an IKEv2 transform proposal used during "+
    "IKEv2 SA negotiation. Multiple IKEv2 Transforms "+
    "can be proposed during an IKEv2 session initiation "+
    "in an ordered list.";
  uses ike-general-proposal-grouping;

  leaf pseudo-random-function {
    type pseudo-random-function-t;
    mandatory true;
    description
      "Specifies Pseudo Random Function for IKEv2 key exchange";
  }

  container authentication {
    description
      "Specify IKEv2 Proposal authentication configuration";
    leaf algorithm {
      type ike-integrity-algorithm-t;
      description
        "Specify the authentication algorithm";
    }
  }
}


grouping ipsec-proposal-grouping {
  description
    "Configure IPSec Proposal";

  leaf name {
    type string;
    mandatory true;
    description
      "IPSec proposal identifier.";
  }

  leaf ah {

type ike-integrity-algorithm-t;
description
"Configure Authentication Header (AH).";
}
container esp {
description
"Configure Encapsulating Security Payload (ESP).";
leaf authentication {
type ike-integrity-algorithm-t;
description
"Configure ESP authentication";
}
leaf encryption {
type ike-encryption-algorithm-t;
description
"Configure ESP encryption";
}
}
leaf ip-comp{
type empty;
description
"Enable IPSec proposal IP-COMP which uses the IP Payload " +
"compression protocol to compress IP Security (IPSec) " +
"packets before encryption";
}
container lifetime {
description
"Configure lifetime for IPSEC SAs";
leaf kbytes {
type uint32 {
    range "128..2147483647";
}
description
"Enter lifetime kbytes for IPSEC SAs";
}
leaf seconds {
type uint32 {
    range "300..99999999";
}
description
"Enter lifetime seconds for IPSEC SAs 
0: lifetime of 0 for no timeout 
300..99999999: IPSec SA lifetime in seconds";
}
}

grouping identity-grouping {
description

"Identification type. It is an union identity, " +
  "possible type as follows: " +
  "a) ID_FQDN: A fully-qualified domain name string. " +
    "An example of a ID_FQDN is, example.com. " +
    "The string MUST not contain any terminators " +
    "(e.g., NULL, CR, etc.). " +
  "b) ID_RFC822_ADDR: A fully-qualified RFC822 email " +
    "address string, An example of a ID_RFC822_ADDR is, " +
    "jsmith@example.com. The string MUST not contain " +
    "any terminators. " +
  "c) ID_IPV4_ADDR: A single four (4) octet IPv4 address. " +
  "d) ID_IPV6_ADDR: A single sixteen (16) octet IPv6 address. " +
  "e) DN_X509: Distinguished name in the X.509 tradition."
choice identity {
  "Choice of identity."
  leaf ipv4-address {
    type inet:ipv4-address;
    description
    "Specifies the identity as a single four (4) octet IPv4 address."
    "An example is, 10.10.10.10."
  }
  leaf ipv6-address {
    type inet:ipv6-address;
    description
    "Specifies the identity as a single sixteen (16) octet IPv6 address."
    "An example is, FF01::101, 2001:DB8:0:0:800:200C:417A."
  }
  leaf fqdn-string {
    type inet:domain-name;
    description
    "Specifies the identity as a Fully-Qualified Domain Name (FQDN) string."
    "An example is: example.com."
    "The string MUST not contain any terminators (e.g., NULL, CR, etc.)."
  }
  leaf rfc822-address-string {
    type string;
    description
    "Specifies the identity as a fully-qualified RFC822 email address string."
    "An example is, jsmith@example.com."
    "The string MUST not contain any terminators (e.g., NULL, CR, etc.)."
  }
leaf dnX509 {
  type string;
  description
    "Specifies the identity as a distinguished name
    in the X.509 tradition."
}
} /* grouping identity-grouping */

grouping ike-general-policy-profile-grouping {
  description
    "IKE policy."
  leaf connection-type {
    type connection-type-t;
    mandatory true;
    description
      "Specify the IKE connection type"
  }
  leaf pre-shared-key {
    type union {
      type string {
        length "16";
      }
      type yang:hex-string {
        length "40";
      }
    }
    description
      "Specify IKE pre-shared-key value"
  }
  leaf validate-certificate-identity {
    type empty;
    description
      "Validate Remote-ID payload against the
      ID's available in the certificate"
  }
  list seq {
    key seq-id;
    description
      "list of sequence of policy."
    leaf seq-id {
      type uint32 {
        range "1..4294967299"
      }
      description
        "Sequence Number"
    }
    leaf proposal {
      type leafref {

path "/eipsec:ike/eipsec:proposal" + "/eipsec:name";
}
}
}
}
}

container identity {
    description "Specify IKE identity value";
    container local {
        description "Specify the identity of the local IP Security (IPSec)
tunnel endpoint in an Internet Key Exchange (IKE)
policy to use when negotiating IKE request with a
remote peer.";
        uses identity-grouping;
    }  
    container remote {
        description "Specify the identity of the remote IP Security (IPSec)
tunnel endpoint in an
Internet Key Exchange (IKE) policy to use when
negotiating IKE request with a remote peer.";
        uses identity-grouping;
    }
}

grouping ike-policy-mode-grouping {
    description "IKE Policy Mode";
    container mode {
        description "Specify IKE mode configuration";
        leaf aggressive {
            type empty;
            description "Set IKE Aggressive mode";
        }
        leaf main {
            type empty;
            description "Set IKE Main mode";
        }
    }
}

grouping ike-policy-profile-grouping {
description
"Configure IKE policy";
leaf name {
  type string;
  mandatory true;
  description
    "Specify an IKE policy name";
}
uses ike-policy-mode-grouping;
uses ike-general-policy-profile-grouping;
}

grouping ikev2-policy-profile-grouping {
  description
    "Common information for multiple IKE sessions to be instantiated on a managed element.
    One or more Ikev2Session instances might refer to this instance."
  leaf name {
    type string;
    mandatory true;
    description
      "Value component of the RDN.";
  }
  container authentication {
    description
      "Specify IKE Proposal authentication configuration";
    leaf preshared-key {
      type empty;
      description
        "Use pre-shared key based authentication";
    }
    leaf rsa-signature {
      type empty;
      description
        "Use signature based authentication by using PKI certificates";
    }
  }
  leaf lifetime {
    type uint32;
    mandatory true;
    description
      "Configure lifetime for IKE SAs
       0: for no timeout.
       300 .. 99999999: IKE SA lifetime in seconds.";
  }
  container address-allocation {

must "./connection-type == 'responder-only'" {
    description
    "address-allocation can be configured only with
    responder-only in ike2 policy";
}
leaf aaa {
    type empty;
    description
    "IRAC address allocation by AAA";
} description
"Specify IKE IRAS address allocation option";
} uses ike-general-policy-profile-grouping;

leaf description {
    type string;
    description
    "Specify the description.";
}

grouping ipsec-policy-grouping {
    description
    "Holds configuration information for IPSec policies.";
    leaf name {
        type string;
        mandatory true;
        description
        "IPSec Policy Identification";
    }
    leaf description {
        type string;
        description
        "Specify the description.";
    }
    leaf anti-replay-window {
        type uint32 {
            range "0 | 32..1024";
        }
        description
        "Configure replay window size
        0: to disable anti-replay-window
        32..1024: IPSec anti-replay-window size in multiple of 32";
    }
    container perfect-forward-secrecy {
        description
        "Configure Perfect Forward Secrecy (PFS) for IPSec Policy";
leaf dh-group {
  type diffie-hellman-group-t;
  description
    "Configure Diffie-Hellman group for
     perfect-forward-secrecy";
}
}

list seq {
  key seq-id;
  description
    "Specify IPSEC proposal sequence number";
  leaf seq-id {
    type uint32;
    description
      "Sequence ID";
  }
  leaf description {
    type string;
    description
      "Specify the description.";
  }

  leaf proposal {
    type leafref {
      path "/ipsec/ipsec/"+
        "ipsec:proposal/ipsec-proposal/ipsec:name";
    }
    description
      "IKE proposal reference.";
  }
}

grouping key-string-grouping {
  description
    "Configure key for authentication algorithm";
  leaf key-str {
    type union {
      type string {
        length "16";
      }
      type yang:hex-string {
        length "40";
      }
    }
    description
      "Key string input is either string value (length of 16)
       or hexadecimal (length of 40)";
  }
}
grouping ipsec-sa-ah-grouping {
  description
  "Configure Authentication Header (AH) for Security Association (SA)";
  container ah {
    description
    "Configure Authentication Header (AH) for SA";
    leaf spi {
      type uint32 {
        range "256..131071";
      }
    }
    leaf description {
      type string;
      description
      "Specify the description.";
    }
  }
  choice authentication-algorithm {
    description
    "choice for authentication algorithm to set for AH";
    case hmac-aes-xcbc {
      container hmac-aes-xcbc {
        description
        "Set the authentication algorithm to hmac-aes-xcbc";
        uses key-string-grouping;
      }
    }
    case hmac-md5-96 {
      container hmac-md5-96 {
        description
        "Set the authentication algorithm to hmac-md5-96";
        uses key-string-grouping;
      }
    }
    case hmac-shal-96 {
      container hmac-shal-96 {
        description
        "Set the authentication algorithm to hmac-shal-96";
        uses key-string-grouping;
      }
    }
    case key-string {
      container key-string {
        description
        "Set the authentication algorithm to key-string";
      }
    }
  }
}
"Configure key for authentication algorithm";
uses key-string-grouping;
}
}
}
}
}

grouping ipsec-sa-esp-grouping {

description
"Configure IPSec Encapsulation Security Payload (ESP)";
}
}
}

container esp {  

description
"Set IPSec Encapsulation Security Payloer (ESP)";
leaf description {  
type string;

description
"Specify the description.";
}
}

container authentication {  

description
"Configure authentication for IPSec Encapsulation Security Payload (ESP)";
choice authentication-algorithm {  

description
"choice for authentication algorithm to set";
}
}

case hmac-aes-xcbc {  

carntainer hmac-aes-xcbc {  

description
"Set the authentication algorithm to hmac-aes-xcbc";
uses key-string-grouping;
}
}

case hmac-md5-96 {  

carntainer hmac-md5-96 {  

description
"Set the authentication algorithm to hmac-md5-96";
uses key-string-grouping;
}
}

case hmac-sha1-96 {  

carntainer hmac-sha1-96 {  

description
"Set the authentication algorithm to hmac-sha1-96";
uses key-string-grouping;
}
}

case key-string {  

container key-string {
  description
  "Configure key for authentication algorithm";
  uses key-string-grouping;
}
}
}

container encryption {
  description
  "Configure encryption for IPSec Encapsulation Security Payload (ESP)";
  choice encryption-algorithm {
    description
    "type of encryption";
    case des3-cbc {
      container des3-cbd {
        description
        "Set the encryption algorithm to des3-cbc";
        uses key-string-grouping;
      }
    }
    case aes-128-cbc {
      container aes-128-cbc {
        description
        "Set the encryption algorithm to aes-128-cbc";
        uses key-string-grouping;
      }
    }
    case aes-192-cbc {
      container aes-192-cbc {
        description
        "Set the encryption algorithm to aes-192-cbc";
        uses key-string-grouping;
      }
    }
    case aes-256-cbc {
      container aes-256-cbc {
        description
        "Set the encryption algorithm to aes-256-cbc";
        uses key-string-grouping;
      }
    }
    case des-cbc {
      container des-cbc {
        description
        "Set the encryption algorithm to des-cbc";
        uses key-string-grouping;
      }
    }
  }
}
case key-string {
  container key-string {
    description
    "Configure key for encryption algorithm";
    uses key-string-grouping;
  }
}

grouping ipsec-acl-dest-grouping {
  description
  "IPSEC ACL destination.";
  /* For destination */
  choice dest-address {
    description
    "destination address.";
    case dest-ipv4-address {
      leaf destination-ipv4-address {
        type inet:ipv4-address;
        description
        "Destination IPv4 Address A.B.C.D/0..32.";
      }
    }
    case dest-any {
      leaf dest-any {
        type empty;
        description
        "Match Any Destination IPv4 Address.";
      }
    }
  }
}

grouping ipsec-acl-seq-protocol-number-grouping {
  description
  "IPSec ACL Sequence protocol number.";
  leaf number {
    type uint16 {
      range "0..255";
    }
    description
    "Specify protocol number.";
  }
  choice argument {
    description
    ""
"Source IPv4 address."
  case source-ipv4-address {
    leaf source-ipv4-address {
      type inet:ipv4-address;
      description "Source IPv4 Address A.B.C.D/0..32."
    }
  }
  case any {
    /* For source */
    leaf source-any {
      type empty;
      description "Match Any Source IPv4 Address.";
    }
  }
}

grouping ipsec-acl-seq-ip-address-grouping {
  description "IPSec ACL Sequence IP Address."
  leaf source-ipv4-address {
    type inet:ipv4-address;
    description "Source is IPv4 Address A.B.C.D/0..32."
  }
}

grouping ipsec-acl-seq-any-grouping {
  description "IPSec ACL Sequence Any."
  leaf any {
    type empty;
    description "Source is Any."
  }
}

grouping ipsec-acl-seq-tcp-grouping {
  description "IPSec ACL Sequence TCP."
  leaf tcp {
    type empty;
    description "Source is TCP protocol."
  }
}
grouping ipsec-acl-seq-udp-grouping {
    description
    "IPSec ACL Sequence for UDP.";
    leaf udp {
        type empty;
        description
        "Source is UDP protocol."
    }
}

grouping ipsec-acl-grouping {
    description
    "IPSec ACL";
    list access-list {
        key "name sequence-number";
        uses name-grouping;
        uses sequence-number-grouping;
        description
        "Configure the IPSec access-list."
        choice protocol {
            description
            "IPSec ACL protocol.";
            case number {
                uses ipsec-acl-seq-protocol-number-grouping;
            }
            case source-ipv4-address {
                uses ipsec-acl-seq-ip-address-grouping;
            }
            case any {
                uses ipsec-acl-seq-any-grouping;
            }
            case tcp {
                uses ipsec-acl-seq-tcp-grouping;
            }
            case udp {
                uses ipsec-acl-seq-udp-grouping;
            }
        }
        uses ipsec-acl-dest-grouping;
    }
}

grouping ipsec-df-bit-grouping {
    description
    "IPSec Don’t Fragment (DF) bit for IP header.";
    container df-bit {
        description
        "Configure Don’t Fragment (DF) bit for IP Header."
        leaf clear {
            type boolean;
            description
            "Clear DF bit in IP header."
        }
    }
}
type empty;
  description
  "Clear DF bit for outer IP header.";
}
leaf propagate {
  type empty;
  description
  "Propagate DF bit for outer IP header.";
}
leaf set {
  type empty;
  description
  "Set DF bit for outer IP header.";
}
}

grouping ipsec-profile-grouping {
  description
  "IPSec profile.";
  list profile {
    key "name";
    uses name-grouping;
    uses ipsec-df-bit-grouping;
    description
    "Configure the IPSec Profile.";
    leaf mtu {
      type uint32 {
        range "256..1600";
      }
      description
      "Set the MTU.";
    }
    list seq {
      key "sequence-number";
      uses sequence-number-grouping;
      description
      "IPSec Access List sequence number.";
      leaf policy {
        type leafref {
          path "/eipsec:ipsec/eipsec:policy="/ + eipsec:ipsec-policy/eipsec:name";
        }
        description
        "Specify IPSec policy name.";
      }
      leaf access-list {
        type leafref {
          path "/econtext:contexts/econtext:context="/ +
"econtext:name/econtext:ipsec"+
"/econtext:access-list/econtext:name";
}

description
"Specify IPSec access-list name.";
}
}

 geological...

/*--------------------*/
/* Configuration Data */
/*--------------------*/

container ike {

description
"Configuration IPSec IKE";
/* The following is for <configure> */

list proposal {

key "name";

uses ike-proposal-grouping;

description
"Configure IKE proposal";
}

leaf keepalive {

type empty;

description
"Enables sending Dead Peer Detection (DPD) messages +
to Internet Key Exchange (IKE) peers.";
}

list policy {

key "name";

uses ike-policy-profile-grouping;

description
"Configure IKE Policy Profile.";
}
}

container ikev2 {

description
"Configuration IPSec IKEv2";
/* The following is for <configure> */

list proposal {

key "name";

uses ikev2-proposal-grouping;

description
"Configure IKEv2 proposal";
}

list policy {

key "name";


uses ikev2-policy-profile-grouping;
description
"IKEv2 Policy Profile";
}
}

container ipsec {
description
"Configuration IPSec";
uses ipsec-acl-grouping;
container alarms {
description
"Configure the IPSec alarm for tunnels";
leaf hold-down {
  type uint8 {
    range "1..120";
  }
description
  "Hold-down time (in seconds) before tunnel alarms are generated";
}
}

container qos {
description
"Configure the IPSec QoS priority queuing policy";
list policy {
  key "name";
  leaf name {
    type string;
    description
    "Specify IPSec QoS priority queuing name";
  }
description
  "Configure IPSec QoS priority queuing name";
  container pq {
    description
    "Configure IPSec QoS priority queuing policy";
    leaf num-queues {
      type uint8 {
        range "1 | 4";
      }
description
        "IPSec QoS Number of queues is either 1 or 4";
    }
  }
}
}

container redundancy {
  description

"Configure redundancy for IPSec";
leaf inter-chassis {
    type empty;
    description
    "Set redundancy at chassis level";
}
}
container security-association {
    description
    "Configure the IPSec Security Association (SA)";
    list ipsec-sa {
        key "name";
        leaf name {
            type string;
            description
            "Specify IPSec Security Association (SA) name";
        }
        description
        "Configure IPSec Security Association (SA)";
        leaf anti-replay-window {
            type uint16 {
                range "0 | 32..1024";
            }
            description
            "Specify replay window size";
        }
        leaf ip-comp {
            type empty;
            description
            "Enables IPCOMP, which uses the IP payload compression
             protocol to compress IP security (IPsec) packets
             before encryption";
        }
        container in {
            description
            "Configure inbound SA";
            uses ipsec-sa-ah-grouping;
            uses ipsec-sa-esp-grouping;
        }
        container out {
            uses ipsec-sa-ah-grouping;
            uses ipsec-sa-esp-grouping;
            description
            "Configure outbound SA";
        }
    }
}
container proposal {
    description
"IPSec Proposal Profile";
list ipsec-proposal {
    key "name";
    uses ipsec-proposal-grouping;
    description
        "Configure the IP Security (IPSec) proposal";
}
}
container policy {
    description
        "Configure the IPSec policy";
list ipsec-policy {
    key "name";
    uses ipsec-policy-grouping;
    description
        "Specify an IPSec policy name";
}
}

/*================================================================------------*/
/* Operational State Data */
/*================================================================------------*/
grouping ike-proposal-state-components {
    description
        "IKE Proposal operational state";
list proposal {
    description
        "Operational data for IKE Proposal";
    leaf name {
        type string {
            length "1..50";
        }
        description
            "Name of the IKE proposal."
    }
    leaf lifetime {
        type uint32;
        units "seconds";
        description
            "lifetime";
    }
    leaf encryption {
        type ike-encryption-algorithm-t;
        description
            "Encryption algorithm";
    }
}
leaf dh-group {
    type diffie-hellman-group-t;
    description
        "Diffie-Hellman group.";
}
leaf authentication {
    type ike-integrity-algorithm-t;
    description
        "authentication";
}

grouping ike-policy-state-grouping {
    description
        "IKE Policy State.";
    list policy {
        description
            "Operational data for IKE policy";
        leaf name {
            type string {
                length "1..50";
            }
            description
                "Name of the IKE Policy.";
        }
        leaf description {
            type string;
            description
                "Description for IKE Policy.";
        }
        leaf mode {
            type enumeration {
                enum aggressive {
                    description
                        "Aggressive mode.";
                }
                enum main {
                    description
                        "Main mode.";
                }
            }
            description
                "IKE policy mode.";
        }
        leaf connection-type {
            type connection-type-t;
            description
                "IKE policy connection type.";
        }
    }
}
leaf local-identity {
    type inet:ipv4-address-no-zone;
    description
    "IP address of the local identity.";
}

leaf remote-identity {
    type inet:ipv4-address-no-zone;
    description
    "IP address of the remote identity.";
}

leaf pre-shared-key {
    type string;
    description
    "Pre-shared key";
}

leaf seq {
    type uint32;
    description
    "sequence number";
}

leaf proposal {
    type string;
    description
    "proposal name";
}
leaf encryption {
    type ike-encryption-algorithm-t;
    description
        "Encryption algorithm";
}
leaf dh-group {
    type diffie-hellman-group-t;
    mandatory true;
    description
        "Diffie-Hellman group.";
}

grouping ipsec-policy-state-grouping {
    description
        "IPSec operational state";
    list policy {
        description
            "IPSec policy operational data";
        leaf name {
            type string;
            description
                "IPSec Policy name.";
        }
        leaf anti-replay-window {
            type uint32;
            description
                "replay window size";
        }
        leaf perfect-forward-secrecy {
            type diffie-hellman-group-t;
            description
                "Diffie-Hellman group for perfect-forward-secrecy";
        }
    list seq {
        description
            "Sequence number";
        leaf seq-id {
            type uint32;
            description
                "Sequence number";
        }
        leaf proposal-name {
            type string;
            description
                "IPSec proposal name";
        }
    }
}
grouping ipsec-proposal-state-grouping {
  description
  "IPSec proposal operational data";
  list proposal {
    description
    "IPSec proposal operational data";
    leaf name {
      type string;
      description
      "IPSec Proposal name";
    }
    leaf ah {
      type ike-integrity-algorithm-t;
      description
      "Authentication Header (AH).";
    }
    container esp {
      description
      "Encapsulating Security Payload (ESP).";
      leaf authentication {
        type ike-integrity-algorithm-t;
        description
        "ESP authentication";
      }
      leaf encryption {
        type ike-encryption-algorithm-t;
        description
        "ESP encryption";
      }
    }
    leaf ip-comp {
      type empty;
      description
      "IPSec proposal IP-COMP which uses the IP Payload" +
      "compression protocol to compress IP Security (IPSec)" +
      "packets before encryption";
    }
    container lifetime {
      description
      "lifetime for IPSEC SAs";
      leaf kbytes {
        type uint32;
        description
        "lifetime kbytes for IPSEC SAs";
      }
    }
  }
}
leaf seconds {
  type uint32;
  description  
    "lifetime seconds for IPSEC SAs";
}
}

  grouping ipsec-alarms-state-grouping {
    description 
      "IPSec alarms operational data";
    leaf hold-down {
      type uint32;
      description 
        "Hold-down value";
    }
  }

  grouping ipsec-sa-ah-state-grouping {
    description 
      "IPSec SA’s AH operational data";
    leaf spi {
      type uint32;
      description 
        "Security Parameter Index (SPI) value";
    }
    leaf description {
      type string;
      description 
        "the description.";
    }
    leaf authentication-algorithm {
      type ike-integrity-algorithm-t;
      description 
        "Authentication algorithm";
    }
    leaf encryption-algorithm {
      type ike-encryption-algorithm-t;
      description 
        "Encryption algorithm";
    }
  }

  grouping ipsec-sa-state-grouping {
    description 
      "IPSec Security Association Operational data";
  }
list sa {
  description "IPSec SA operational data";
  leaf name {
    type string;
    description "Specify IPSec Security Association (SA) name";
  }
  leaf anti-replay-window {
    type uint16;
    description "replay window size";
  }
  leaf ip-comp {
    type empty;
    description "Enables IPCOMP, which uses the IP payload compression protocol to compress IP security (IPsec) packets before encryption";
  }
  uses ipsec-sa-ah-state-grouping;
}

container ike-state {
  config "false";
  uses ike-proposal-state-components;
  uses ike-policy-state-grouping;
  description "Contain the operational data for IKE.";
}

container ikev2-state {
  config "false";
  uses ikev2-proposal-state-components;
  uses ikev2-policy-state-grouping;
  description "Contain the operational data for IKEv2.";
}

container ipsec-state {
  config "false";
  uses ipsec-policy-state-grouping;
  uses ipsec-proposal-state-grouping;
  uses ipsec-alarms-state-grouping;
  uses ipsec-sa-state-grouping;
  description "Contain the operational data for IPsec.";
/*--------------------*/
/* RPC                */
/*--------------------*/

rpc clear-ipsec-group {
  description
    "RPC for clear ipsec states";
  input {
    leaf alarm-hold-down {
      type uint8;
      description
        "IPSec alarm hold-down";
    }
    leaf ipsec-policy-name {
      type leafref {
        path "/eipsec:ipsec/eipsec:policy/+
          "eipsec:ipsec-policy/eipsec:name";
      }
      description
        "IPSec Policy name.";
    }
  }
}

rpc clear-ike-group {
  description
    "RPC for clear IKE states";
  input {
    leaf proposal {
      type leafref {
        path "/eipsec:ike/eipsec:proposal/+
          "eipsec:ike Commonwealth of Australia
  "eipsec:name";
      }
      description
        "IPSec IKE Proposal name.";
    }
  }
}

rpc clear-ikev2-group {
  description
    "RPC for clear IKEv2 states";
  input {
    leaf proposal {
      type leafref {
        path "/eipsec:ikev2/eipsec:proposal/+
          "eipsec:name";
      }
      description
        "IPSec IKEv2 Proposal name.";
    }
  }
}
5. Security Considerations

The configuration, state, and action data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

6. References

6.1. Normative References


6.2. Informative References


Authors' Addresses

Khanh Tran
Ericsson
300 Holger Way
San Jose, CA 95134
USA

Email: khanh.x.tran@ericsson.com