Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 17, 2015

H. Wang
V. Nagaraj
X. Chen
Huawei Technologies

June 15, 2015

Yang Data Model for IPsec
draft-wang-ipsecme-ipsec-yang-00

Abstract

This document describes a YANG data model for the IPsec(Internet Protocol Security) protocol. The model covers the IPsec protocol operational state and remote procedural calls.

Requirements Language

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 [RFC2119].

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on December 17, 2015.

Copyright Notice

Table of Contents

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is a network management protocol that defines mechanisms to manage network devices. YANG [RFC6020] is a modular language that represents data structures in an XML tree format, and is used as a data modeling language for the NETCONF.

This document introduces a YANG data model for the IPsec(Internet Protocol Security) protocol[RFC4301]. The data model is defined for following constructs that are used for managing the IPsec protocol: operational state and remote procedural calls.

2. IPsec YANG Model Organization

2.1. Overview

The model discussed in this document covers IPsec[RFC4301] and other generic enhancements that pertain to the base protocol operation. The cryptographic algorithms are deliberately separated from ietf-ipsec model so that these algorithms can be updated or replaced

without affecting the standardization progress of the rest of the IPsec yang model.

^: import

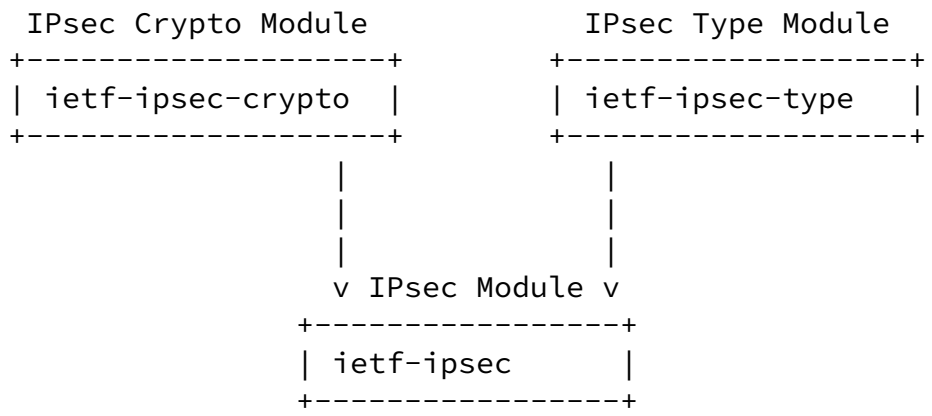

Figure 1: Relationship of IPsec module and other modules

This model aims to address only the core IPsec parameters as per [RFC4301]. This model does not cover any applications running on top of IPsec nor does it cover any OAM procedures for IPsec. Current revision only describes SAD and SPD, PAD will be covered in later revision.

Different IPsec implements may have different behaviors, e.g. a host may directly bind IPsec SA with socket, then SPD is not necessary; while a gateway may supply interfaces for IKE[RFC7296] to modify IPsec SPD entries. So we defined only the basic prototype of the data model, and all the databases are defined as read only. Any other extension and augment of the data model are left for implements.

The figure below describes the overall structure of the IPsec Yang model:

```
module: ietf-ipsec
   +--ro sad
   |  ...
   +--ro spd
   |  ...
   +--ro ipsec-global-statistics
      +--ro ipv4
      |  ...
      +--ro ipv6
      |  ...
      +--ro global
         ...
rpcs:
   +---x reset-ipv4
   |  ...
   +---x reset-ipv6
   |  ...
   +---x reset-global
      ...
```

2.2. Operational State

The Operational state of the IPsec can be queried and obtained from the respective container. All the attributes/items in this container are read-only attributes and they reflect the run-time information of IPsec database.

2.2.1. IPsec SAD State

The IPsec SAD(Security Association Database) container maintains information related to the IPSEC SAs established in a system. This is a run-time data structure that is created upon the first SA being established. The key for fetching SA in this database is the triplet: SPI, Protocol and Destination address of the SA to be fetched form the SA database.

The SAD entries also contain information about the IPSEC tunnel like direction, SA-type (manual or VPN SA), sequence number, anti-replay window size, protocol mode, ipsec algorithm info, life time in Seconds/Bytes etc, NAT traversal info, path-mtu, dscp etc.


```
+--ro sad
   +--ro sad-entries* [spi security-protocol direction]
      +--ro spi                                ipsec-type:ipsec-spi
      +--ro security-protocol                  ipsec-type:ipsec-protocol
      +--ro direction                          ipsec-type:ipsec-traffic-direction
      +--ro sa-type?                           enumeration
      +--ro sequence-number?                   uint64
      +--ro sequence-number-overflow-flag?     boolean
      +--ro anti-replay-enable-flag?           boolean
      +--ro anti-replay-window-size?           uint64
      +--ro ah-auth-algorithm?                 ipsec-crypto:ipsec-authentication-
      +--ro esp-integrity-algorithm?           ipsec-crypto:ipsec-authentication-
      +--ro esp-encrypt-algorithm?             ipsec-crypto:ipsec-encryption-algo
      +--ro life-time
      |  +--ro life-time-in-seconds?            uint32
      |  +--ro remain-life-time-in-seconds?     uint32
      |  +--ro life-time-in-byte?               uint32
      |  +--ro remain-life-time-in-byte?        uint32
      +--ro protocol-mode?                     ipsec-type:ipsec-mode
      +--ro tunnel-mode-process-info
      |  +--ro local-address?                 string
      |  +--ro remote-address?                string
```

```
      |  +--ro bypass-df?                   enumeration
      |  +--ro dscp-flag?                   boolean
      |  +--ro stateful-frag-check-flag?    boolean
      +--ro dscp*                             uint8
      +--ro path-mtu?                         uint16
      +--ro nat-traversal-flag?               boolean
```

2.2.2. IPsec SPD State

The IPSEC SPD(Security Policy Database) container maintains policy information related to the IPSEC SAs established in a system. This is a run-time data structure that is created when the first IPSEC policy is created.

The SPD entries also contain information about the traffic selectors, protect action (permit, deny), protocol information etc as shown below. Based on these information the IPSEC module processes the outbound and inbound traffic.


```
+--ro spd
   +--ro spd-entries*
      +--ro name*
      |  +--ro name-type?     ipsec-type:ipsec-spd-name
      |  +--ro name-string?   string
      |  +--ro name-binary?   binary
      +--ro pfp-flag?            boolean
      +--ro traffic-selector*
      |  +--ro local-address-low?     inet:ip-address
      |  +--ro local-address-high?    inet:ip-address
      |  +--ro remote-address-low?    inet:ip-address
      |  +--ro remote-address-high?   inet:ip-address
      |  +--ro next-protocol-low?     uint16
      |  +--ro next-protocol-high?    uint16
      |  +--ro local-port-low?        inet:port-number
```

```
   |  +--ro local-port-high?       inet:port-number
   |  +--ro remote-port-high?      inet:port-number
   |  +--ro remote-port-low?       inet:port-number
   +--ro operation?           ipsec-type:ipsec-spd-operation
   +--ro protect-operation
      +--ro spd-ipsec-mode?           ipsec-type:ipsec-mode
      +--ro esn-flag?                 boolean
      +--ro spd-ipsec-protocol?       ipsec-type:ipsec-protocol
      +--ro tunnel-mode-additional
      |  +--ro local-address?              string
      |  +--ro remote-address?             string
      |  +--ro bypass-df?                  enumeration
      |  +--ro dscp-flag?                  boolean
      |  +--ro stateful-frag-check-flag?   boolean
      +--ro spd-algorithm*
         +--ro ah-auth-algorithm?         ipsec-crypto:ipsec-authentication-
         +--ro esp-integrity-algorithm?   ipsec-crypto:ipsec-authentication-
         +--ro esp-encrypt-algorithm?     ipsec-crypto:ipsec-encryption-algo
```

2.2.3. IPsec Global Statistics

The IPSEC Global Statistics container is used to maintain information related to all the IPSEC tunnels established in the system. These could be related to IPv4 IPSEC tunnels or IPv6 IPSEC tunnels.

The information maintained includes: traffic sent/received on an IPSEC tunnel like number of outbound/inbound packets, number of outbound/inbound bytes, number of packets dropped, number of replayed packets, number of packet authentication failures, number of packets dropped due to queue full, number of packets dropped due to deny policy, number of packet dropped due to being malformed, number of packets dropped due to being too large.


```
   +--ro ipsec-global-statistics
       +--ro ipv4
       |  +--ro inbound-packets?         uint64
       |  +--ro outbound-packets?        uint64
       |  +--ro inbound-bytes?           uint64
       |  +--ro outbound-bytes?          uint64
       |  +--ro inbound-drop-packets?    uint64
       |  +--ro outbound-drop-packets?   uint64
```

```
|  +--ro dropped-packet-detail
|     +--ro sa-non-exist?   uint64
|     +--ro queue-full?     uint64
|     +--ro auth-failure?   uint64
|     +--ro malform?        uint64
|     +--ro replay?         uint64
|     +--ro large-packet?   uint64
|     +--ro invalid-sa?     uint64
|     +--ro policy-deny?    uint64
|     +--ro other-reason?   uint64
+--ro ipv6
|  +--ro inbound-packets?         uint64
|  +--ro outbound-packets?        uint64
|  +--ro inbound-bytes?           uint64
|  +--ro outbound-bytes?          uint64
|  +--ro inbound-drop-packets?    uint64
|  +--ro outbound-drop-packets?   uint64
|  +--ro dropped-packet-detail
|     +--ro sa-non-exist?   uint64
|     +--ro queue-full?     uint64
|     +--ro auth-failure?   uint64
|     +--ro malform?        uint64
|     +--ro replay?         uint64
|     +--ro large-packet?   uint64
|     +--ro invalid-sa?     uint64
|     +--ro policy-deny?    uint64
|     +--ro other-reason?   uint64
+--ro global
   +--ro inbound-packets?         uint64
   +--ro outbound-packets?        uint64
   +--ro inbound-bytes?           uint64
   +--ro outbound-bytes?          uint64
   +--ro inbound-drop-packets?    uint64
   +--ro outbound-drop-packets?   uint64
   +--ro dropped-packet-detail
      +--ro sa-non-exist?   uint64
      +--ro queue-full?     uint64
      +--ro auth-failure?   uint64
      +--ro malform?        uint64
      +--ro replay?         uint64
```



```
      +--ro large-packet?   uint64
```

```
            +--ro invalid-sa?     uint64
            +--ro policy-deny?    uint64
            +--ro other-reason?   uint64
```

2.3. Actions

This model defines a list of RPCs that allow performing an action or executing a command on the protocol. In current version of this document, we only defined how to reset IPsec statistics, other actions are left for later version of this document.

2.3.1. IPsec statistics reset action

This operation type is executed when the user wants to reset IPSEC SA statistics. The operation will reset the global IPSEC4 statistics in the system.

```
rpcs:
   +---x reset-ipv4
   |  +---w input
   |  |  +---w ipv4?   empty
   |  +--ro output
   |     +--ro status?   string
   +---x reset-ipv6
   |  +---w input
   |  |  +---w ipv6?   empty
   |  +--ro output
   |     +--ro status?   string
   +---x reset-global
      +---w input
      |  +---w ipv6?   empty
      +--ro output
         +--ro status?   string
```

3. IPsec Yang Module

To support separately upgrade the algorithm part, the algorithm part is defined as separately part.

3.1. IPsec Yang Module

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

  import ietf-ipsec-crypto {
    prefix ipsec-crypto;
```

```
  }
  import ietf-inet-types {
    prefix inet;
  }
  import ietf-ipsec-type {
    prefix ipsec-type;
  }

  organization "Huawei Technologies India Pvt Ltd";
  contact
    "stonewater.wang@huawei.com";
  description
    "IPsec Yang";

  revision 2015-04-18 {
    description
      "Initial revision.";
    reference "RFC XXX: IPsec Yang Modules";
  }

  grouping ipsec-tunnel-mode-info {
    description
      "common infomations when using IPsec tunnel mode";
    leaf local-address {
      type string;
      description
        "Local address of IPsec tunnel mode";
    }
    leaf remote-address {
      type string;
      description
        "Remote address of IPsec tunnel mode";
    }
    leaf bypass-df {
      type enumeration {
        enum "set" {
          description
            "Set the df bit";
        }
        enum "clear" {
          description
            "Clear the df bit";
        }
        enum "copy" {
          description
            "Copy the df bit from inner header";
```

```
        }
      }
```



```
    description
      "This flag indicates how to process tunnel mode df flag";
  }
  leaf dscp-flag {
    type boolean;
    description
      "This flag indicate whether bypass DSCP or map to unprotected DSCP valu
  }
  leaf stateful-frag-check-flag {
    type boolean;
    description
      "This flag indicates whether stateful fragment checking will be used.";
  }
}

grouping traffic-selector {
  description
    "IPsec traffic selector information";
  leaf local-address-low {
    type inet:ip-address;
    description
      "Low range of local address";
  }
  leaf local-address-high {
    type inet:ip-address;
    description
      "High range of local address";
  }
  leaf remote-address-low {
    type inet:ip-address;
    description
      "Low range of remote address";
  }
  leaf remote-address-high {
    type inet:ip-address;
    description
      "High range of remote address";
  }
  leaf next-protocol-low {
```

```
      type uint16;
      description
        "Low range of next protocol";
    }
    leaf next-protocol-high {
      type uint16;
      description
        "High range of next protocol";
    }
```



```
    leaf local-port-low {
      type inet:port-number;
      description
        "Low range of local port";
    }
    leaf local-port-high {
      type inet:port-number;
      description
        "High range of local port";
    }
    leaf remote-port-high {
      type inet:port-number;
      description
        "Low range of remote port";
    }
    leaf remote-port-low {
      type inet:port-number;
      description
        "High range of remote port";
    }
  }

  grouping ipsec-algorithm-info {
    description
      "IPsec algorithm information used by SPD and SAD";
    leaf ah-auth-algorithm {
      type ipsec-crypto:ipsec-authentication-algorithm;
      description
        "Authentication algorithm used by AH";
    }
    leaf esp-integrity-algorithm {
      type ipsec-crypto:ipsec-authentication-algorithm;
```

```
      description
        "Integrity algorithm used by ESP";
    }
    leaf esp-encrypt-algorithm {
      type ipsec-crypto:ipsec-encryption-algorithm;
      description
        "Encryption algorithm used by ESP";
    }
  }

  grouping ipsec-stat {
    leaf inbound-packets {

        type uint64;
        config false;
        description "Inbound Packet count";
```



```
    }
    leaf outbound-packets {
        type uint64;
        config false;
        description "Outbound Packet count";
    }
    leaf inbound-bytes {
        type uint64;
        config false;
        description "Inbound Packet bytes";
    }
    leaf outbound-bytes {
        type uint64;
        config false;
        description "Outbound Packet bytes";
    }

    leaf inbound-drop-packets {
        type uint64;
        config false;
        description "Inbound dropped packets count";
    }
    leaf outbound-drop-packets {
        type uint64;
        config false;
```

```
        description "Outbound dropped packets count";
    }
    container dropped-packet-detail {
        description "The detail information of dropped packets";
        leaf sa-non-exist {
            type uint64;
            config false;
            description "The dropped packets counts caused by SA non-exist.";
        }
        leaf queue-full {
            type uint64;
            config false;
            description "The dropped packets counts caused by full processing q
        }

        leaf auth-failure {
            type uint64;
            config false;
            description "The dropped packets counts caused by authentication fa
        }

        leaf malform {
            type uint64;
```



```
            config false;
            description "The dropped packets counts of malform";
        }
        leaf replay {
            type uint64;
            config false;
            description "The dropped packets counts of replay";
        }
        leaf large-packet {
            type uint64;
            config false;
            description "The dropped packets counts of too large";
        }
        leaf invalid-sa {
            type uint64;
            config false;
            description "The dropped packets counts of invalid SA";
        }
```

```
        leaf policy-deny {
            type uint64;
            config false;
            description "The dropped packets counts of denyed by policy";
        }
        leaf other-reason {
            type uint64;
            config false;
            description "The dropped packets counts of other reason";
        }
    }
    description "IPsec statistics information";

  }

  container sad {

    config false;

    description
      "The IPsec SA database";

    list sad-entries {
      key "spi security-protocol direction";
      description
        "The SA entries information";
      leaf spi {
        type ipsec-type:ipsec-spi;
        description
          "Security parameter index of SA entry.";
```



```
      }
      leaf security-protocol {
        type ipsec-type:ipsec-protocol;
        description
          "Security protocol of IPsec SA.";
      }
      leaf direction {
        type ipsec-type:ipsec-traffic-direction;
        description
          "It indicates whether the SA is inbound SA or out bound SA.";
      }
```

```
      leaf sa-type {
        type enumeration {
          enum "manual" {
            description
              "Manual IPsec SA";
          }
          enum "isakmp" {
            description
              "ISAKMP IPsec SA";
          }
        }
        description
          "It indicates whether the SA is created by manual or by dynamic proto
      }
      leaf sequence-number {
        type uint64;
        description
          "Current sequence number of IPsec packet.";
      }
      leaf sequence-number-overflow-flag {
        type boolean;
        description
          "The flag indicating whether overflow of the sequence number counter
      }
      leaf anti-replay-enable-flag {
        type boolean;
        description
          "It indicates whether anti-replay is enable or disable.";
      }
      leaf anti-replay-window-size {
        type uint64;
        description
          "The size of anti-replay window.";
      }
      uses ipsec-algorithm-info;
      container life-time {
        leaf life-time-in-seconds {
```



```
          type uint32;
          description
            "SA life time in seconds";
        }
```

```
      leaf remain-life-time-in-seconds {
        type uint32;
        description
          "Remain SA life time in seconds";
      }
      leaf life-time-in-byte {
        type uint32;
        description
          "SA life time in bytes";
      }
      leaf remain-life-time-in-byte {
        type uint32;
        description
          "Remain SA life time in bytes";
      }
      description
        "SA life time information";
    }
    leaf protocol-mode {
      type ipsec-type:ipsec-mode;
      description
        "It indicates whether tunnel mode or transport mode will be used.";
    }
    container tunnel-mode-process-info {
      when "protocol-mode = 'tunnel'" {
        description
          "External information of SA when SA works in tunnel mode.";
      }
      uses ipsec-tunnel-mode-info;
      description
        "External information of SA when SA works in tunnel mode.";
    }
    leaf-list dscp {
      type uint8 {
        range "0..63";
      }
      description
        "When traffic matchs SPD, the DSCP values used to filter traffic";
    }
    leaf path-mtu {
      type uint16;
      description
        "Path MTU valie";
    }
```

```
      leaf nat-traversal-flag {
        type boolean;
        description
          "Whethe the SA is used to protect traffic that nedds nat traversal";
      }
    }
  }
  container spd {
    config false;
    description
      "IPsec security policy database information";

    list spd-entries {
      description
        "IPsec SPD entry information";
      list name {
        description
          "SPD name information.";
        leaf name-type {
          type ipsec-type:ipsec-spd-name;
          description
            "SPD name type.";
        }
        leaf name-string {
          when "name-type = 'id_rfc_822_addr' or name-type = 'id_fqdn'" {
            description
              "when name type is id_rfc_822_addr or id_fqdn, the name are saved
          }
          type string;
          description
            "SPD name content";
        }
        leaf name-binary {
          when "name-type = 'id_der_asn1_dn' or name-type = 'id_key'" {
            description
              "when name type is id_der_asn1_dn or id_key, the name are saved i
          }
          type binary;
          description
            "SPD name content";
        }
      }
      leaf pfp-flag {
        type boolean;
        description
          "populate from packet flag";
      }
```

```
      list traffic-selector {
```



```
        min-elements 1;
        uses traffic-selector;
        description
          "Traffic selectors of SAD entry";
      }
      leaf operation {
        type ipsec-type:ipsec-spd-operation;
        description
          "It indicates how to process the traffic when it matches the security
      }
      container protect-operation {
        when "operation = 'protect'" {
          description
            "How to protect the traffic when the SPD operation is protect";
        }
        leaf spd-ipsec-mode {
          type ipsec-type:ipsec-mode;
          description
            "It indicates which mode is chosen when the traffic need be protect
        }
        leaf esn-flag {
          type boolean;
          description
            "It indicates whether ESN is used.";
        }
        leaf spd-ipsec-protocol {
          type ipsec-type:ipsec-protocol;
          description
            "It indicates which protocol (AH or ESP) is chosen.";
        }
        container tunnel-mode-additional {
          when "spd-ipsec-mode = 'tunnel'" {
            description
              "Additional informations when choose tunnel mode";
          }
          uses ipsec-tunnel-mode-info;
          description
            "When use tunnel mode, the additional information of SPD.";
        }
        list spd-algorithm {
```

```
          min-elements 1;
          uses ipsec-algorithm-info;
          description
            "Algorithms defined in SPD, ordered by decreasing priority.";
        }
        description
          "How to protect the traffic when the SPD operation is protect";
      }
```



```
    }
  }

  container ipsec-global-statistics {
    config false;
    description "IPsec global statistics";

    container ipv4 {
        description "IPsec statistics of IPv4";
        uses ipsec-stat;
    }

    container ipv6 {
        description "IPsec statistics of IPv6";
        uses ipsec-stat;
    }

    container global {
        description "IPsec statistics of global";
        uses ipsec-stat;
    }
  }


    rpc reset-ipv4 {
        description "Reset IPsec IPv4 statistics";
        input {
            leaf ipv4 {
                type empty;
                description "Reset IPsec IPv4 statistics";
            }
        }
        output {
```

```
            leaf status {
                type string;
                description "Operation status";
            }
        }

    }
    rpc reset-ipv6 {
        description "Reset IPsec IPv6 statistics";
        input {
            leaf ipv6 {
                type empty;
                description "Reset IPsec IPv6 statistics";
            }
        }
```



```
        output {
            leaf status {
                type string;
                description "Operation status";
            }
        }

    }
    rpc reset-global {
        description "Reset IPsec global statistics";
        input {
            leaf ipv6 {
                type empty;
                description "Reset IPsec global statistics";
            }
        }
        output {
            leaf status {
                type string;
                description "Operation status";
            }
        }

    }

}
```

3.2. IPsec Algorithm Yang Module

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

  organization "Huawei Technologies India Pvt Ltd";
  contact
    "stonewater.wang@huawei.com";
  description
    "IPsec Crypto Yang";
  reference
    "RFC 4301: Security Architecture for the Internet Protocol";

  revision 2015-04-18 {
    description
      "Initial revision.";
    reference
      "RFC 4301: Security Architecture for the Internet Protocol";
  }
```



```
  typedef ipsec-authentication-algorithm {
    type enumeration {
      enum "null" {
        value 0;
        description
          "null";
      }
      enum "md5" {
        value 1;
        description
          "MD5 authentication algorithm";
      }
      enum "sha1" {
        value 2;
        description
          "SHA1 authentication algorithm";
      }
      enum "sha2-256" {
        value 3;
```

```
          description
            "SHA2-256 authentication algorithm";
        }
        enum "sha2-384" {
          value 4;
          description
            "SHA2-384 authentication algorithm";
        }
        enum "sha2-512" {
          value 5;
          description
            "SHA2-512 authentication algorithm";
        }
      }
      description
        "typedef for ipsec authentication algorithm";
    }

    typedef ipsec-encryption-algorithm {
      type enumeration {
        enum "null" {
          description
            "null";
        }
        enum "des" {
          description
            "DES encryption algorithm";
        }
        enum "3des" {
```



```
          description
            "3DES encryption algorithm";
        }
        enum "aes-128" {
          description
            "AES-128 encryption algorithm";
        }
        enum "aes-192" {
          description
            "AES-192 encryption algorithm";
        }
        enum "aes-256" {
```

```
            description
              "AES-256 encryption algorithm";
          }
        }
        description
          "typedef for ipsec encryption algorithm";
      }
    }
```

3.3. IPsec Type Yang Module

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

  organization "Huawei Technologies India Pvt Ltd";
  contact
    "stonewater.wang@huawei.com";
  description
    "common type define for ipsec protocol Yang";
  reference "RFC 4301: Security Architecture for the Internet Protocol";

  revision 2015-04-18 {
    description
      "Initial revision.";
    reference "RFC 4301: Security Architecture for the Internet Protocol";
  }

  typedef ipsec-mode {
    type enumeration {
      enum "transport" {
        description
          "Transport mode";
      }
      enum "tunnel" {
        description
```



```
          "Tunnel mode";
      }
    }
    description
      "type define of ipsec mode";
```

```
  }

  typedef ipsec-protocol {
    type enumeration {
      enum "ah" {
        description
          "AH Protocol";
      }
      enum "esp" {
        description
          "ESP Protocol";
      }
    }
    description
      "type define of ipsec security protocol";
  }

  typedef ipsec-spi {
    type uint32 {
      range "1..max";
    }
    description
      "SPI";
  }

  typedef ipsec-spd-name {
   type enumeration {
    enum id_rfc_822_addr {
        description
          "Fully qualified user name string.";
      }
      enum id_fqdn {
        description
          "Fully qualified DNS name.";
      }
      enum id_der_asn1_dn {
        description
          "X.500 distinguished name.";
      }
      enum id_key {
        description
          "IKEv2 Key ID.";
      }
```

```
    }
    description
      "IPsec SPD name type";
  }

  typedef ipsec-traffic-direction {
    type enumeration {
      enum inbound {
        description
          "Inbound traffic";
      }
      enum outbound {
        description
          "Outbound traffic";
      }
    }
    description
      "IPsec traffic direction";
  }

  typedef ipsec-spd-operation {
    type enumeration {
      enum protect {
        description
          "PROTECT the traffic with IPsec";
      }
      enum bypass {
        description
          "BYPASS the traffic";
      }
      enum discard {
        description
          "DISCARD the traffic";
      }
    }
    description
      "The operation when traffic matches IPsec security policy";
  }

}
```

4. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-ipsec XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-ipsec-crypto XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-ipsec-type XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-ipsec namespace: urn:ietf:params:xml:ns:yang:ietf-ipsec prefix: ipsec reference: [RFC4301]

5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol[RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content. There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

6. Acknowledgements

7. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, October 2014.

Authors' Addresses

Honglei Wang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: stonewater.wang@huawei.com

Vijay Kumar Nagaraj
Huawei Technologies
Huawei Technologies India Pvt Ltd
Bangalore 560008
India

Email: vijay.kn@huawei.com

Xia Chen
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: xiachen@huawei.com