Network Working Group
Internet Draft
Intended status: Standard Track
Expires: September 18, 2016

K. Tran
D. Migault
Ericsson
H. Wang
V. Nagaraj
X. Chen
Huawei Technologies

March 18, 2016

Yang Data Model for IKEv2
draft-tran-ipsecme-ikev2-yang-00.txt

Abstract

This document defines a YANG data model that can be used to configure and manage Internet Key Exchange version 2 (IKEv2). The model covers the IKEv2 protocol configuration and operational state.

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 18, 2016.

Copyright Notice

Table of Contents

1. Introduction

This document introduces a YANG data model for the Internet Exchange Key version 2 (IKEv2) protocol. The model discussed in this document covers IKEv2 [RFC7296] and other generic enhancements that pertain to the base protocol operation. The YANG data model is defined for the following constructs that are used for managing the IKEv2 protocol including configuration and operational state.

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. IKEv2 protocol Overview

This section provides a high level overview of IKEv2 [RFC7296] to make the YANG model more comprehensive. The intent of this section is to fill the gap between the IKEv2 specifications and its associated YANG model. It is expected to clarify the YANG model, for those that are more familiar to the IKEv2 specifications, and provide some IKEv2 background for those that are more familiar to YANG models.

Note that the purpose of IKEv2 standard is to provide interoperability whereas the YANG model provides an implementation independent way to configure IKEv2 daemons. With these different goals application-dependent parameters or parameters that interoperability-independent (like the life time of the IKE SA for example) are not mentioned in the IKEv2 standard but needs to be specified in the YANG model.

IKEv2 can be designed as a single monolithic daemon that is configured in a single manner for all initiated and responding IKEv2 negotiation. On the other hand, IKEv2 can also be view as a daemon that can enable some specific configuration for each peer. This would mean for example that the IKE_SA could be set differently according to the peer. In addition to these different levels of configuration granularities, the IKEv2 daemon is not always aware of the peer identity. When it acts as a responder, for example, the peer ID is only known during the IKE_AUTH exchange, which means that during the previous exchange (IKE_INIT) the IKEv2 daemon is likely not to apply a per peer policy.

In order to address the multiple possible configurations the IKEv2 configuration and variables are subdivided into different modules. An IKEv2 daemon needs to have all these modules to be specified, however, each module may be specified at different level in the tree. More specifically, module may be set for the global implementation or for each peer.

3.1. IKEv2 Transport Attributes

This section provides the attributes used to enable the transport of the IKEv2 messages between the initiator and the peer. The transport often needs configuration attributes that define the behavior of the IKEv2 daemon according to operational attributes (or counters).

IKEv2 Header defines the attributes that identifies the IKE session between the peers. Although the configuration attributes may be common for the whole implementation, it is expected that the

operational attributes are defines from each session, that is for each IKE_SA. These attributes are provided in the header and are described in [RFC7296] section 3.1. Although the IKE header contains also attributes such as Message IDs, and flags for example that indicate if corresponds to a query or a response, these headers attributes are not considers as operational attributes of the IKE header, instead, these are considered as operational attribute of the Anti-Replay Mechanism. The attributes associated to the IKEv2 Header are thus:

- MjVer: defines the major version. As defined in [RFC7296] section 3.1 implementations that of [RFC7296] MUST set this attribute to 2.
- MnVer: defines the minor version. As defined in [RFC7296] section 3.1 implementations that of [RFC7296] MUST set this attribute to 0.
- SPI-generation-policies: defines how the SPI are expected to be generated. Most likely SPIs will randomly generate. On the other hand, it may be needed for some deployment such as clusters to be able to reduce the spectrum of these SPIs.
- Initiator SPI: defines the SPI assigned by the Initiator to index the inbound messages to the appropriated IKE_SA. The SPIs are agreed between the peers after the IKE_INIT exchange and are not part of the configuration parameters.
- Responder SPI: defines the SPI assigned by the Responder to index the inbound messages to the appropriated IKE_SA.

IKEv2 Header Configuration Attributes # [RFC7296] section 3.1
- MjVer: The IKEv2 Major version (set to 2)
- MnVer: The IKEv2 Minor version (set to 0)
- SPI-generation-policies

IKEv2 Header Operational Attributes (1 per IKE_SA)
- Initiator SPI
- Responder SPI

Anti-Replay Mechanism describes when message should be rejected or considered by the IKEv2 daemon. The anti-replay mechanism is defined for each session. Although the configuration attributed may be shared for the whole IKEv2 daemon, the operational attributes are expected to be duplicated for each IKE_SA. The following attributes are thus considered.

- Window Size defines how much parallel exchange can be performed between the peers. By default this value is set to 1. When greater than 1, as defined in [RFC7296] section 2.3, a

SET_WINDOW_SIZE Notify Payloads will be sent by the peer to agree with the other peer on the Window Size. After this exchange succeeds, the operational attribute that defines the Window Size used by the IKE_SA, will be updated with the value agreed by the peers.

. Optional Enable INVALID_MESSAGE_ID defines whether an optional INVALID_MESSAGE_ID Notify Payload is sent when the IKEv2 message received is outside the Operational Window Size.

. Operational Window Size defines the Window size considered by the IKE_SA. When the IKE_SA is created, it is set to 1. This value is updated only once the peers have agreed on another Window Size value with the SET_WINDOW_SIZE informational exchange.

. Peer Request MESSAGE ID stores the Message ID of the last request received by the peer.

. Peer Request MESSAGE ID stores the Message ID of the last response received by the peer.

. Local Request MESSAGE ID stores the Message ID of the last request received by the local host.

. Local Request MESSAGE ID stores the Message ID of the last response received by the local host.

```
Anti-Replay Mechanism Configuration Attributes
    - Window Size                        # [RFC7296] section 2.3
    - Optional Enable INVALID_MESSAGE_ID # [RFC7296] section 2.3

Anti-Replay Mechanism Operational Attributes (1 per IKE_SA)
    - Operational Window Size = 1 # [RFC7296] section 2.3
    - Peer Request MESSAGE ID     # [RFC7296] section 2.2
    - Peer Response MESSAGE_ID    # [RFC7296] section 2.2
    - Local Request MESSAGE_ID    # [RFC7296] section 2.2
    - Local Response MESSAGE_ID   # [RFC7296] section 2.2
```

IKEv2 Retransmission defines the necessary attributes to manage the retransmission of message by the IKEv2 daemon. Such attributes are not necessary for interoperability and as such are not defined in [RFC7296]. However, retransmission mechanism is described in [RFC7296] section 2.1. Although the configuration mechanism may be common to the IKEv2 daemon, the operational attributes are expected to be defined for each IKE_SA exchange. The number of parallel IKEv2 exchange is defined by Window Size.

. Max Retries: [RFC7256] section 2.1 mentions that when retransmission fails, all states associated to the IKE SA MUST be removed.
. Initial Retransmission Timeout: [RFC7256] section 2.1 mentions the retransmission timeout is not expected to be a fix value, but instead it should depend on the on number of retries. How the retransmission-timer value is set depends on the Retransmission Timer Policy.
. Retransmission Timer Policy: defines of the Retransmission Timer should be computed.
. Response Buffer Timeout: (section 2.1 of RFC7256). This timer set when the response buffer can be clean when the message ID is not being updated. It value is expected to be in the order of several minutes.
. Retries: Defines the number of retries for a given exchange. The number of exchange is defined by the Window Size.
. Retransmission Timeout: is an operational attribute that set how long the IKEv2 daemon should wait until a retransmission occurs. This attribute is derived from the Retransmission Timer Policy and the Initial Retransmission Timeout.
. Retransmission Timer: is an operational attribute that defines the time the response is being waited for. When its value reaches, Retransmission Timeout, a retransmission occurs. This Timer is set for each exchange.
. Response Buffer Timer: is an operational value that counts the time each Message ID is stored. There is a timer associated to each Message ID.

```
IKEv2 Retransmission Configuration Attributes
    - Max Retries                    # [RFC7296] section 2.1
    - Initial Retransmission Timeout # [RFC7296] section 2.1
    - Retransmission Timeout Policy
    - Max Response Buffer Timeout    # [RFC7296] section 2.1
    - Keep-Alive Timeout
    - NAT Keep-Alive Timeout
```

IKEv2 Retransmission Operational Attributes (Window Size per IKE_SA)
- Retries
- Retransmission Timeout
- Retransmission Timer
- Response Buffer Timer
- Keep-Alive Timer
- NAT Keep-Alive Timer

IKEv2 COOKIE MECHANISM Configuration Attributes
- COOKIE Lifetime
- Half Open IKE_SA Threshold

IKEv2 COOKIE MECHANISM Operational Attributes (Window Size per IKE_SA)
- Half Open IKE_SA Counter

IKEv2 VENDOR ID Configuration Attributes

- OPAQUE VALUES

3.2. IKEv2_INIT Exchange

This section provides the necessary configuration attributes so the IKE_INIT exchange can be performed.

Authorized DH is an ordered list that contains DH Transform. DH Transforms are ordered by preference. Such ordering avoids setting an additional preference field. The Initiator will choose the first and most preferred DH Transform to initiate the IKE_INIT. The DH public key will be generated and the chosen DH Transform will be included into the Transform Type 4 of the SAi1. If the DH Transform is not accepted by the Responder, the Initiator may check the acceptable DH Transform of the responder is acceptable by the initiator.

IKE_SA Proposals defines the proposals similarly to the proposals structure of SA1i. Note that the IKEv2 daemon is expected to place the appropriated Transform of Type 4, that it the chosen DH Transform. In addition, the IKEv2 daemons associates each transform to an ID to build SA1i.

Optional IKE_INIT Responder CERTREQ indicates whether the Certification authority supported by the responder should be added into the response.

Authorized Certification Authorities lists the CA considered by the responder.

Supported IKEv2 Options defines the option supported by the IKEv2 daemon. Some options should be considered in the IKE_INIT exchange, other should be considered in the IKE_AUTH exchange. To avoid duplication of the supported IKEv2 Options, they are all indicated here. Each Option may be associated some specific configuration and operational attributes detailed.

IKEv2_INIT Exchange Configuration Attributes:

```
## Attributes Model is common to object so it is defined as
## a preambule
Attributes [list]
    - Attribute
        - Attribute Type
        - Attribute Value

## Ordered list of the authorized DH
Authorized DH [list]
    - DH Transform
        - Name
        - Attributes

## Ordered list of proposals, the preference is indicated by the Num
IKE_SA Proposals [list]
    - IKE_SA Proposal
        - Proposal Num  # specify the order the proposals are sent.
                        # Need to check there are no two identical
                        # numbers
        - Protocol: IKE # It has a fix value
        - Transform Type 1: Encryption Algorithm [list]
            - ENCR Transform
                - Name
                - Attributes
        - Transform Type 2: PRF [list]
            - PRF Transform
                - Name
        - Transform Type 3: Integrity check Algorithm [list]
            - INTEG Transform
```

```
                - Name
                - Attributes
          ##- Transform Type 4: Diffie Hellman Group
          ## RFC7296 this MUST be the DH Transform used in the KEi


   ## lists the authorized Certification Authorities
   Authorized Certification_Authorities [list]
       - Certification Authority
           - Cert Encoding
           - Cert Value

   Optional IKE_INIT Responder CERTREQ

   ## IKEv2 options
   Supported IKEv2 Options
       ## sent during the IKE_INIT
       - NAT_DETECTION_SOURCE_IP
       - NAT_DETECTION_DESTINATION_IP
       - REDIRECT_SUPPORTED
       - IKEV2_FRAGMENTATION_SUPPORTED
       ## sent during the IKE_AUTH
       - MOBIKE_SUPPORTED
       - ROHC_SUPPORTED
       - CHILDLESS_IKEV2_SUPPORTED
       - IKEV2_MESSAGE_ID_SYNC_SUPPORTED
       - IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED
       - ERX_SUPPORTED
       - CLONE_IKE_SA_SUPPORTED
```

Section 1 of [RFC7296] provides a description of the IKEv2 exchanges. The purpose of the first exchange is that the initiator and the responder are able to set a IKE SA. The IKE SA can be seen as a control channel between the initiator and the responder that will be used for further negotiations. To reach an agreement on the IKE SA, the initiator and the responder must agree on the SKEYSEED (KEi, Ni KEr, Nr payloads) that is a Diffie Hellman value and nonces used to derived the cryptographic keys for the IKE SA and further IPsec SA or Child SA. In addition, the initiator and the respond must agree on how the IKE SA will use the cryptographic material (SAi1, SAr1).

The IKE_INIT exchange is represented below:

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SAi1, KEi, Ni  -->
                              <--  HDR, SAr1, KEr, Nr, [CERTREQ]
```

All header of the IKEv2 payloads have a header which is built from the IKEv2 Header values as well as the IKE_SA for the SPI values.

KEi is derived from Authorized DH that is an ordered list of DH parameters. The public key is not stored into the model and is computed by the initiator. The chosen transform MUST be inserted in Transform 4 of IKE_SA Proposal in SA1i.

KEr is able to determine whether KEi is acceptable from the Authorized DH. In case the the KEi is not acceptable, the responder responds with an INVALID_KE_PAYLOAD.

SAi1 is derived from IKE_SA Proposals and KEi

SAr1: is derived by comparing the proposals from SA1i and the IKE_SA Proposals. The responder is able to chose the appropriated IKE proposal as well as to define whether none of the SAi1 is acceptable.

Optional IKE_INIT Responder CERTREQ indicates whether the responder sends CERTREQ payloads, the following attribute should be defined. When set to true, one CERTREQ payload is provided per Certification Authority in the Authorized Certification Authority.

When the NAT_DETECTION_SOURCE_IP, NAT_DETECTION_DESTINATION_IP, REDIRECT_SUPPORTED or IKEV2_FRAGMENTATION_SUPPORTED have been enabled, then additional notify payloads are added by the initiator. Unless not supported by the responder, the responder responds to them with an additional Notify payload.

3.3. Creation of the IKE_SA

In this model, it is assumed that the IKE_SA represents the relation between the initiator and the responder. It is expected that the IKE_SA model is created as soon as a peer initiates a IKE_INIT exchange as well as a peer receives a new IKE_INIT request. Of course this is implementation dependent, but the model relies on this assumption.

The IKE_SA information model is represented with the following attributes:

- Role: defines if the local peer acts as an initiator or as a responder.
- Local IP address: defines the IP address used by the local peer.
- Remote IP address: defines the IP address of the remote peer.
- Cryptographic material is derived after the IKE_INIT exchange. The IKE_SA may keep the original material SKEYSEED and Nonces Ni, Nr used to generate the necessary keys SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi, SK_pr. The following keys are used to protect the exchange.
- IKE SA Proposal: the agreed IKE_SA proposal.
- IKEv2 Header: the header with the agreed SPI values.
- IKEv2 Anti Replay Mechanism which contains the agreed (or to be agreed Window Size) and current Message IDs. According to RFC7296 section 2.2 Message IDs of the INKE_INIT exchange are set to 0 during the IKE_INIT exchange.
- IKEv2 Retransmission CTX that contains the element to enable retransmission for all ongoing exchange.
- IDi/IDr, Credentials are defined during the IKE_AUTH exchange.
- Vendor IDs.
- Supported IKEv2 Option CTX contains all necessary context associated to the different IKEv2 Options.

```
   IKE_SA Operational Attributes

   IKE_SA
       - Role
       - Local IP address
       - Remote IP address
       - Cryptographic material
           - SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi, SK_pr
           - SKEYSEED, Nonces
       - IKE_SA lifetime
       - IKE SA Proposal                ## cf IKE_INIT section
       - IKEv2 Header                   ## cf Transport section
       - IKEv2 Anti Replay Mechanism    ## cf Transport section
       - IKEv2 Retransmission CTX [list Window Size] ## cf Transport
                                                        section
           - IKEv2 Retransmission
       - IDi                            ## cf IKE_AUTH section
       - IDr                            ## cf IKE_AUTH section
       - Credentials                    ## fc IKE_AUTH section
       - Vendor ID
       - Supported IKEv2 Option CTX [list]
```

3.4. IKE_AUTH Exchange

This section provides the attributes associated to the IKE_AUTH exchange.

The IKE_AUTH and CREATE_CHILD_SA exchange is represented below. The IKE_AUTH exchange goal is to authenticate the respective peers and the CREATE_CHILD_SA exchange intends to creates the PIsec SA.

```
HDR, SK {IDi, [CERT,] [CERTREQ,]
  [IDr,] AUTH, SAi2,
  TSi, TSr}  -->
                          <--  HDR, SK {IDr, [CERT,] AUTH,
                                   SAr2, TSi, TSr}
```

Authentication is performed by providing an identity as well as a proof of ownership associated to that identity. The Initiator and Responder may have multiple identities and choose one. The Initiator may choose a specific identity according to the expected responder, and vise versa, the responder may choose a specific identity according to the initiator identity (IDi) as well as the acceptable Certificate Authorities of the initiator (CERTREQ) or the Certificate Authority of the initiator, that is the one used in its Certificate (CERT).

Available Signing Capabilities defines the signing capabilities of the IKEv2 daemon. A Signing capability is defined by a method and some Authentication Material such as a public key for example, or a certificate.

Available Hash Capabilities and and Available Signature Verification defines which are the acceptable authentication method provided by the remote peer. In other words, outside these Signature Verification and Hash Capabilities the peer will not be able to be authenticated. The difference with Available Signing Capabilities is that in this case, no credentials are required. For example a RSA signature may be checked without the peer own a RSA private key. Has and Signature are placed in different attributes as a signature

verification often results in a combination of these two structures. The authentication life time indicates when re-authentication needs to be performed. The minimum of the two values should be considered.

Local IDs lists the various IDs the Local IKEv2 daemon may use to identify itself. The Preference field indicates which one should be used preferably, but in most cases, it is expected that the Local Id to use will depend on teh remote peer.

Peer is the database of the Peer attributes. A Peer is defined by a list of IDr and a role. Once the Peer has been identified, it may be associated to some specific attributes to proceed the IKE_AUTH exchange. For example, suppose that the Local Peer want to set an IKE session with a Remote Peer, and both Peers have multiple IDs. When the Local Peer wants to reach the Remote Peer, it may use a specific IDi and request a specific IDr for that session. In addition, it can also redefine all configuration attributes previously defined for the IKE-Transport, IKE_INIT and IKE_AUTH.

Note that The definition of the Preferred IDr is only mandatory when the Local Peer initiates the exchange, so when the Remote Peer is a responder. In that case, the IDi and IDr will be use to provide the appropriated parameters for the CREATE_CHILD_SA exchange. As detailed in Section 4.4.3 of RFC4301, the PAD use used to provide such binding.

Optional attributes defines whether the optional payloads should be added or if an additional notification payload should be exchanged.

IKEv2_AUTH Configuration Attributes

Available Signing Capabilities [list]
- Authentication Method
- Authentication Method Name
- Authentication Material
  - Authentication Material Type
  - Authentication Material Data

CERT Authentication Material
- Authentication Material Type = CERT
- Authentication Material Data
  - Cert Encoding
  - Cert Value

```
   Available Hash Capabilities [list]
        - Hash Method
        - Authentication Life Time

   Available Signature Verification [list]
       - Authentication Method Name
       - Authentication Life Time

   Local IDs [list]
      - Local ID
          - preference
            - ID type
            - ID value


   Peers [list]
       - Peer
           - PeerIDs [list] # use to identify the peer
               - IDr
           - Role initiator / responder / any # this is only to make
   sure we can have different policies depending on who initiates the
   communication.
           - Sessions [list]
               - Session
                   - Session Label
                   - IDi
                   ## When initiating an IKEv2 exchange with Peer
                   - IDr
                   ## Can set (redefine) all configuration attributes
                   - IKE_Tranport Attributes
                   - IKE_INIT Attributes
                   - IKE_AUTH Attributes
                   - ...
                   - Optional Configuration Request
                       - INTERNAL_ADDRESS
                       - ...
                   - Optional Configuration Reply
                       - INTERNAL_ADDRESS

   Optional Enable INITIAL_CONTACT #[RFC7296] section 2.4
   Optional IKE_AUTH Initiator CERTREQ
   Optional IKE_AUTH Initiator CERT
   Optional IKE_AUTH Initiator-IDr
   Optional IKE_AUTH Responder-CERT
```

3.5. IKEv2 Configuration Data Model

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

```
module: ietf-ikev2
   +--rw ikev2 {ikev2}?
   |  +--rw transport {ikev2-transport}?
   |  +--rw init {ikev2-init}?
   |  +--rw sa {ikev2-sa}?
   |  +--rw peer* [peer-address] {ikev2-peer}?
```

The tree detail is:

```
   +--rw ikev2 {ikev2}?
   |  +--rw transport {ikev2-transport}?
   |  |  +--rw base-info
   |  |  |  +--rw major-version?          uint8
   |  |  |  +--rw minor-version?          uint8
   |  |  |  +--rw spi-generation-policy?  string
   |  |  +--rw anti-replay-mechanism
   |  |  |  +--rw window-size?                   uint32
   |  |  |  +--rw enable-notify-invalid-msg-id?  empty {ikev2-transport-
enable-notify-
   invalid-msg-id}?
   |  |  +--rw retransmision {ikev2-transport-retransmission}?
   |  |  |  +--rw max-retries?                     uint32
   |  |  |  +--rw initial-retransmission-timeout?  uint32
   |  |  |  +--rw retransmission-timeout-policy?   string
   |  |  |  +--rw max-response-buffer-timeout?     uint32
   |  |  |  +--rw keepalive-timeout?               uint32
   |  |  |  +--rw nat-keepalive-timeout?           uint32
   |  |  +--rw cookie-mechanism {ikev2-transport-cookie-mechanism}?
   |  |  |  +--rw cookie-lifetime?              uint32
   |  |  |  +--rw half-open-ike-sa-threshold?   uint32
   |  |  +--rw vendor-id?               uint64
   |  +--rw init {ikev2-init}?
   |  |  +--rw authorized-dh* [dhg key-length] {ikev2-init-authorized-dh}?
   |  |  |  +--rw dhg           ikev2-crypto:ikev2-diffie-hellman-group-t
   |  |  |  +--rw key-length    uint32
   |  |  +--rw proposal* [number]
   |  |  |  +--rw name?                           string
   |  |  |  +--rw description?                    string
```

```
      |  |  |  +--rw transform-encr-algorithm* [encr-algorithm key-length]
      |  |  |  |  +--rw encr-algorithm    ikev2-crypto:ikev2-encryption-
algorithm-t
      |  |  |  |  +--rw key-length        uint32
      |  |  |  +--rw transform-prf-algorithm* [prf-algorithm key-length]
      |  |  |  |  +--rw prf-algorithm    ikev2-crypto:ikev2-pseudo-random-
function-t
```

```
      |  |  |  |  +--rw key-length       uint32
      |  |  |  +--rw transform-integrity-algorithm* [integrity-algorithm key-
length]
      |  |  |  |  +--rw integrity-algorithm    ikev2-crypto:ikev2-integrity-
algorithm-t
      |  |  |  |  +--rw key-length             uint32
      |  |  |  +--rw transform-dh* [dh key-length]
      |  |  |  |  +--rw dh            ikev2-crypto:ikev2-diffie-hellman-group-t
      |  |  |  |  +--rw key-length    uint32
      |  |  |  +--rw number                          uint32
      |  |  |  +--rw protocol?                       ikev2-crypto:ikev2-
protocol-identifiers-
   t
      |  |  +--rw optional {ikev2-init-optional}?
      |  |  |  +--rw nat-detection-source-ip {ikev2-init-nat-detection-src-ip}?
      |  |  |  |  +--rw (ip-address)?
      |  |  |  |  |  +--:(ipv4-address)
      |  |  |  |  |  |  +--rw ipv4-address?            inet:ipv4-address
      |  |  |  |  |  +--:(ipv6-address)
      |  |  |  |  |     +--rw ipv6-address?            inet:ipv6-address
      |  |  |  |  +--rw nat-keepalive-interval?   uint16
      |  |  |  +--rw nat-detection-destination-ip {ikev2-init-nat-detection-
destination-ip}?
      |  |  |  |  +--rw (ip-address)?
      |  |  |  |  |  +--:(ipv4-address)
      |  |  |  |  |  |  +--rw ipv4-address?            inet:ipv4-address
      |  |  |  |  |  +--:(ipv6-address)
      |  |  |  |  |     +--rw ipv6-address?            inet:ipv6-address
      |  |  |  |  +--rw nat-keepalive-interval?   uint16
      |  |  |  +--rw redirect-supported?                   boolean {ikev2-
init-redirect-
   supported}?
      |  |  |  +--rw fragmentation-supported?              boolean {ikev2-
init-fragmentation-
   supported}?
      |  |  |  +--rw mobike-supported?                     boolean {ikev2-
auth-mobike-
   supported}?
      |  |  |  +--rw rohc-supported?                       boolean {ikev2-
auth-rohc-
   supported}?
      |  |  |  +--rw childless-ikev2-supported?            boolean {ikev2-
auth-childless-
   supported}?
      |  |  |  +--rw message-id-sync-supported?            boolean {ikev2-
auth-message-id-
   supported}?
      |  |  |  +--rw ipsec-replay-counter-sync-supported?   boolean {ikev2-
auth-ipsec-replay-
```

```
   counter-sync-supported}?
      |  |  |  +--rw erx-supported?                          boolean {ikev2-
auth-erx-
   supported}?
      |  |  |  +--rw clone-ike-sa-supported?                 boolean {ikev2-
auth-clone-ike-sa-
   supported}?
      |  |  +--rw auth-method?            ikev2-crypto:ikev2-authentication-
method-t
      |  |  +--rw responder-certreq {ikev2-init-responder-certreq}?
      |  |  |  +--rw cert-encoding?   ikev2-crypto:ikev2-cert-encoding-t
      |  |  |  +--rw cert-value?      uint32
      |  |  +--rw config-request
      |  |  |  +--rw (ip-address)?
      |  |  |     +--:(ipv4-address)
      |  |  |     |  +--rw ipv4-address?   inet:ipv4-address
      |  |  |     +--:(ipv6-address)
      |  |  |        +--rw ipv6-address?   inet:ipv6-address
      |  |  +--rw config-responder
      |  |  |  +--rw (ip-address)?
```

```
      |  |  |     +--:(ipv4-address)
      |  |  |     |  +--rw ipv4-address?   inet:ipv4-address
      |  |  |     +--:(ipv6-address)
      |  |  |        +--rw ipv6-address?   inet:ipv6-address
      |  |  +--rw authorized-cert-auth* [cert-encoding] {ikev2-init-authorized-
certification-
   auth}?
      |  |     +--rw cert-encoding    ikev2-crypto:ikev2-cert-encoding-t
      |  |     +--rw cert-value?      uint32
      |  +--rw sa {ikev2-sa}?
      |  |  +--rw role?                       role-t
      |  |  +--rw local-ip-address
      |  |  |  +--rw (ip-address)?
      |  |  |     +--:(ipv4-address)
      |  |  |     |  +--rw ipv4-address?   inet:ipv4-address
      |  |  |     +--:(ipv6-address)
      |  |  |        +--rw ipv6-address?   inet:ipv6-address
      |  |  +--rw remote-ip-address
      |  |  |  +--rw (ip-address)?
      |  |  |     +--:(ipv4-address)
      |  |  |     |  +--rw ipv4-address?   inet:ipv4-address
      |  |  |     +--:(ipv6-address)
      |  |  |        +--rw ipv6-address?   inet:ipv6-address
      |  |  +--rw cryptgraphic?               cryptographic-material-t
      |  |  +--rw lifetime?                   uint32
      |  |  +--rw proposal?                   ikev2-proposal-number-ref
      |  |  +--rw base-info
      |  |  |  +--rw major-version?           uint8
      |  |  |  +--rw minor-version?           uint8
      |  |  |  +--rw spi-generation-policy?   string
      |  |  +--rw anti-replay-mechanism
      |  |  |  +--rw window-size?                    uint32
      |  |  |  +--rw enable-notify-invalid-msg-id?   empty {ikev2-transport-
enable-notify-
   invalid-msg-id}?
      |  |  +--rw retransmistion-ctx* [window-id]
      |  |  |  +--rw window-id        uint32
      |  |  |  +--rw retransmision {ikev2-transport-retransmission}?
      |  |  |     +--rw max-retries?                      uint32
      |  |  |     +--rw initial-retransmission-timeout?   uint32
      |  |  |     +--rw retransmission-timeout-policy?    string
      |  |  |     +--rw max-response-buffer-timeout?      uint32
      |  |  |     +--rw keepalive-timeout?                uint32
      |  |  |     +--rw nat-keepalive-timeout?            uint32
      |  |  +--rw initiator-id
      |  |  |  +--rw initiator-id-type?   ikev2-crypto:pad-type-t
      |  |  |  +--rw initiator-id?        string
      |  |  +--rw responder-id
      |  |  |  +--rw responder-id-type?   ikev2-crypto:pad-type-t
```

```
|  |  |  +--rw responder-id?        string
|  |  +--rw cert-authentication-type?   string
|  |  +--rw cert-auth
|  |  |  +--rw cert-auth-encoding?   ikev2-crypto:ikev2-cert-encoding-t
|  |  |  +--rw cert-auth-value?      uint32
|  |  +--rw vendor-id?                   uint64
|  |  +--rw optional-ctx* [window-id]
|  |     +--rw window-id    uint32
```

```
      |  |     +--rw optional {ikev2-init-optional}?
      |  |        +--rw nat-detection-source-ip {ikev2-init-nat-detection-src-
ip}?
      |  |        |  +--rw (ip-address)?
      |  |        |  |  +--:(ipv4-address)
      |  |        |  |  |  +--rw ipv4-address?            inet:ipv4-address
      |  |        |  |  +--:(ipv6-address)
      |  |        |  |     +--rw ipv6-address?            inet:ipv6-address
      |  |        |  +--rw nat-keepalive-interval?   uint16
      |  |        +--rw nat-detection-destination-ip {ikev2-init-nat-detection-
destination-
   ip}?
      |  |        |  +--rw (ip-address)?
      |  |        |  |  +--:(ipv4-address)
      |  |        |  |  |  +--rw ipv4-address?            inet:ipv4-address
      |  |        |  |  +--:(ipv6-address)
      |  |        |  |     +--rw ipv6-address?            inet:ipv6-address
      |  |        |  +--rw nat-keepalive-interval?   uint16
      |  |        +--rw redirect-supported?                   boolean {ikev2-
init-redirect-
   supported}?
      |  |        +--rw fragmentation-supported?              boolean {ikev2-
init-
   fragmentation-supported}?
      |  |        +--rw mobike-supported?                     boolean {ikev2-
auth-mobike-
   supported}?
      |  |        +--rw rohc-supported?                       boolean {ikev2-
auth-rohc-
   supported}?
      |  |        +--rw childless-ikev2-supported?           boolean {ikev2-
auth-childless-
   supported}?
      |  |        +--rw message-id-sync-supported?            boolean {ikev2-
auth-message-id-
   supported}?
      |  |        +--rw ipsec-replay-counter-sync-supported?   boolean {ikev2-
auth-ipsec-
   replay-counter-sync-supported}?
      |  |        +--rw erx-supported?                        boolean {ikev2-
auth-erx-
   supported}?
      |  |        +--rw clone-ike-sa-supported?               boolean {ikev2-
auth-clone-ike-
   sa-supported}?
      |  +--rw peer* [peer-address] {ikev2-peer}?
      |     +--rw peer-address       string
      |     +--rw role?              role-t
      |     +--rw peer-id-entries* [peer-id peer-id-type]
```

```
|     |  +--rw peer-id-type    ikev2-crypto:pad-type-t
|     |  +--rw peer-id         string
|     +--rw session* [session-label]
|     |  +--rw session-label       string
|     |  +--rw initiator-id
|     |  |  +--rw initiator-id-type?   ikev2-crypto:pad-type-t
|     |  |  +--rw initiator-id?        string
|     |  +--rw responder-id
|     |  |  +--rw responder-id-type?   ikev2-crypto:pad-type-t
|     |  |  +--rw responder-id?        string
|     |  +--rw transport {ikev2-transport}?
|     |  |  +--rw base-info
|     |  |  |  +--rw major-version?           uint8
|     |  |  |  +--rw minor-version?           uint8
|     |  |  |  +--rw spi-generation-policy?   string
|     |  |  +--rw anti-replay-mechanism
|     |  |  |  +--rw window-size?                   uint32
```

```
      |     |  |  |  +--rw enable-notify-invalid-msg-id?   empty {ikev2-
transport-enable-
   notify-invalid-msg-id}?
      |     |  |  +--rw retransmision {ikev2-transport-retransmission}?
      |     |  |  |  +--rw max-retries?                    uint32
      |     |  |  |  +--rw initial-retransmission-timeout?   uint32
      |     |  |  |  +--rw retransmission-timeout-policy?    string
      |     |  |  |  +--rw max-response-buffer-timeout?      uint32
      |     |  |  |  +--rw keepalive-timeout?                uint32
      |     |  |  |  +--rw nat-keepalive-timeout?            uint32
      |     |  |  +--rw cookie-mechanism {ikev2-transport-cookie-mechanism}?
      |     |  |  |  +--rw cookie-lifetime?              uint32
      |     |  |  |  +--rw half-open-ike-sa-threshold?   uint32
      |     |  |  +--rw vendor-id?               uint64
      |     |  +--rw init {ikev2-init}?
      |     |  |  +--rw authorized-dh* [dhg key-length] {ikev2-init-authorized-
dh}?
      |     |  |  |  +--rw dhg           ikev2-crypto:ikev2-diffie-hellman-
group-t
      |     |  |  |  +--rw key-length    uint32
      |     |  |  +--rw proposal* [number]
      |     |  |  |  +--rw name?                            string
      |     |  |  |  +--rw description?                     string
      |     |  |  |  +--rw transform-encr-algorithm* [encr-algorithm key-
length]
      |     |  |  |  |  +--rw encr-algorithm    ikev2-crypto:ikev2-encryption-
algorithm-t
      |     |  |  |  |  +--rw key-length        uint32
      |     |  |  |  +--rw transform-prf-algorithm* [prf-algorithm key-length]
      |     |  |  |  |  +--rw prf-algorithm    ikev2-crypto:ikev2-pseudo-
random-function-t
      |     |  |  |  |  +--rw key-length       uint32
      |     |  |  |  +--rw transform-integrity-algorithm* [integrity-algorithm
key-length]
      |     |  |  |  |  +--rw integrity-algorithm    ikev2-crypto:ikev2-
integrity-algorithm-t
      |     |  |  |  |  +--rw key-length             uint32
      |     |  |  |  +--rw transform-dh* [dh key-length]
      |     |  |  |  |  +--rw dh            ikev2-crypto:ikev2-diffie-hellman-
group-t
      |     |  |  |  |  +--rw key-length    uint32
      |     |  |  |  +--rw number                           uint32
      |     |  |  |  +--rw protocol?                        ikev2-crypto:ikev2-
protocol-
   identifiers-t
      |     |  |  +--rw optional {ikev2-init-optional}?
      |     |  |  |  +--rw nat-detection-source-ip {ikev2-init-nat-detection-
src-ip}?
      |     |  |  |  |  +--rw (ip-address)?
```

```
      |     |  |  |  |  |  +--:(ipv4-address)
      |     |  |  |  |  |  |  +--rw ipv4-address?            inet:ipv4-address
      |     |  |  |  |  |  +--:(ipv6-address)
      |     |  |  |  |  |     +--rw ipv6-address?            inet:ipv6-address
      |     |  |  |  |  +--rw nat-keepalive-interval?   uint16
      |     |  |  |  +--rw nat-detection-destination-ip {ikev2-init-nat-
detection-destination-
   ip}?
      |     |  |  |  |  +--rw (ip-address)?
      |     |  |  |  |  |  +--:(ipv4-address)
      |     |  |  |  |  |  |  +--rw ipv4-address?            inet:ipv4-address
      |     |  |  |  |  |  +--:(ipv6-address)
      |     |  |  |  |  |     +--rw ipv6-address?            inet:ipv6-address
      |     |  |  |  |  +--rw nat-keepalive-interval?   uint16
      |     |  |  |  +--rw redirect-supported?                   boolean
{ikev2-init-
   redirect-supported}?
      |     |  |  |  +--rw fragmentation-supported?              boolean
{ikev2-init-
   fragmentation-supported}?
```

```
      |     |  |  |  +--rw mobike-supported?                      boolean
{ikev2-auth-mobike-
   supported}?
      |     |  |  |  +--rw rohc-supported?                        boolean
{ikev2-auth-rohc-
   supported}?
      |     |  |  |  +--rw childless-ikev2-supported?             boolean
{ikev2-auth-
   childless-supported}?
      |     |  |  |  +--rw message-id-sync-supported?             boolean
{ikev2-auth-message-
   id-supported}?
      |     |  |  |  +--rw ipsec-replay-counter-sync-supported?   boolean
{ikev2-auth-ipsec-
   replay-counter-sync-supported}?
      |     |  |  |  +--rw erx-supported?                         boolean
{ikev2-auth-erx-
   supported}?
      |     |  |  |  +--rw clone-ike-sa-supported?                boolean
{ikev2-auth-clone-
   ike-sa-supported}?
      |     |  |  +--rw auth-method?            ikev2-crypto:ikev2-
authentication-method-t
      |     |  |  +--rw responder-certreq {ikev2-init-responder-certreq}?
      |     |  |  |  +--rw cert-encoding?   ikev2-crypto:ikev2-cert-encoding-t
      |     |  |  |  +--rw cert-value?      uint32
      |     |  |  +--rw config-request
      |     |  |  |  +--rw (ip-address)?
      |     |  |  |     +--:(ipv4-address)
      |     |  |  |     |  +--rw ipv4-address?   inet:ipv4-address
      |     |  |  |     +--:(ipv6-address)
      |     |  |  |        +--rw ipv6-address?   inet:ipv6-address
      |     |  |  +--rw config-responder
      |     |  |  |  +--rw (ip-address)?
      |     |  |  |     +--:(ipv4-address)
      |     |  |  |     |  +--rw ipv4-address?   inet:ipv4-address
      |     |  |  |     +--:(ipv6-address)
      |     |  |  |        +--rw ipv6-address?   inet:ipv6-address
      |     |  |  +--rw authorized-cert-auth* [cert-encoding] {ikev2-init-
authorized-
   certification-auth}?
      |     |  |     +--rw cert-encoding    ikev2-crypto:ikev2-cert-encoding-t
      |     |  |     +--rw cert-value?      uint32
      |     |  +--rw auth {ikev2-auth}?
      |     |  |  +--rw avail-signing-capabilities* [auth-method-name]
      |     |  |  |  +--rw auth-method-name      string
      |     |  |  |  +--rw auth-method?          ikev2-crypto:ikev2-
authentication-method-t
      |     |  |  |  +--rw auth-material-data?   string
```

```
      |     |  |  +--rw cert-auth
      |     |  |  |  +--rw cert-auth-encoding?   ikev2-crypto:ikev2-cert-
encoding-t
      |     |  |  |  +--rw cert-auth-value?      uint32
      |     |  |  +--rw avail-hash* [hash-method]
      |     |  |  |  +--rw hash-method           string
      |     |  |  |  +--rw auth-hash-lifetime?   uint32
      |     |  |  +--rw avail-signature-verify* [signature-id]
      |     |  |  |  +--rw signature-id          string
      |     |  |  |  +--rw signature-lifetime?   uint32
      |     |  |  +--rw local-id* [host-id]
      |     |  |  |  +--rw host-id       string
      |     |  |  |  +--rw preference?   string
      |     |  |  |  +--rw id-type?      string
      |     |  |  |  +--rw id-value?     string
      |     |  |  +--rw authorized-certificate-authority
      |     |  |     +--rw cert-encoding?   ikev2-crypto:ikev2-cert-encoding-t
```

```
|     |  |     +--rw cert-value?      uint32
|     |  +--rw config-request
|     |  |  +--rw (ip-address)?
|     |  |     +--:(ipv4-address)
|     |  |     |  +--rw ipv4-address?   inet:ipv4-address
|     |  |     +--:(ipv6-address)
|     |  |        +--rw ipv6-address?   inet:ipv6-address
|     |  +--rw config-responder
|     |     +--rw (ip-address)?
|     |        +--:(ipv4-address)
|     |        |  +--rw ipv4-address?   inet:ipv4-address
|     |        +--:(ipv6-address)
|     |           +--rw ipv6-address?   inet:ipv6-address
|     +--rw preshared-key?     string
|     +--rw nat-traversal?     boolean
```

3.6. IKEv2 Operation Data Model

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

```
+--ro ikev2-state {ikev2-state}?
   +--ro transport-state {ikev2-transport-state}?
   +--ro ike-sa-state* [initiator-spi responder-spi]
```

The tree detail is:

```
   +--ro ikev2-state {ikev2-state}?
   +--ro ikev2-state {ikev2-state}?
      +--ro transport-state {ikev2-transport-state}?
      |  +--ro major-version?           uint8
      |  +--ro minor-version?           uint8
      |  +--ro spi-generation-policy?   string
      |  +--ro exchange-type?           ikev2-crypto:ikev2-exchange-type-t
      |  +--ro flags?                   uint8
      +--ro sa-state* [initiator-spi responder-spi]
         +--ro initiator-spi               ipsec-spi
         +--ro responder-spi               ipsec-spi
         +--ro retransmistion-ctx* [window-id]
         |  +--ro window-id        uint32
         |  +--ro retransmision {ikev2-transport-retransmission}?
         |     +--ro max-retries?                      uint32
         |     +--ro initial-retransmission-timeout?   uint32
         |     +--ro retransmission-timeout-policy?    string
         |     +--ro max-response-buffer-timeout?      uint32
         |     +--ro keepalive-timeout?                uint32
         |     +--ro nat-keepalive-timeout?            uint32
         +--ro anti-replay-mechanism
         |  +--ro window-size?             uint32
         |  +--ro peer-request-msg-id?     uint32
         |  +--ro peer-response-msg-id?    uint32
         |  +--ro local-request-msg-id?    uint32
         |  +--ro local-response-msg-id?   uint32
         +--ro vendor-id?                  uint64
         +--ro initiator-id
         |  +--ro initiator-id-type?   ikev2-crypto:pad-type-t
         |  +--ro initiator-id?        string
         +--ro responder-id
         |  +--ro responder-id-type?   ikev2-crypto:pad-type-t
         |  +--ro responder-id?        string
         +--ro auth {ikev2-auth}?
         |  +--ro avail-signing-capabilities* [auth-method-name]
```

```
            |  |  +--ro auth-method-name      string
            |  |  +--ro auth-method?          ikev2-crypto:ikev2-
authentication-method-t
            |  |  +--ro auth-material-data?   string
            |  +--ro cert-auth
```

```
            |  |  +--ro cert-auth-encoding?   ikev2-crypto:ikev2-cert-encoding-
t
            |  |  +--ro cert-auth-value?      uint32
            |  +--ro avail-hash* [hash-method]
            |  |  +--ro hash-method           string
            |  |  +--ro auth-hash-lifetime?   uint32
            |  +--ro avail-signature-verify* [signature-id]
            |  |  +--ro signature-id          string
            |  |  +--ro signature-lifetime?   uint32
            |  +--ro local-id* [host-id]
            |  |  +--ro host-id       string
            |  |  +--ro preference?   string
            |  |  +--ro id-type?      string
            |  |  +--ro id-value?     string
            |  +--ro authorized-certificate-authority
            |     +--ro cert-encoding?   ikev2-crypto:ikev2-cert-encoding-t
            |     +--ro cert-value?      uint32
            +--ro half-open-ike-sa-counter?   uint32
            +--ro optional-ctx* [window-id]
               +--ro window-id    uint32
               +--ro optional {ikev2-init-optional}?
                  +--ro nat-detection-source-ip {ikev2-init-nat-detection-src-
ip}?
                  |  +--ro (ip-address)?
                  |  |  +--:(ipv4-address)
                  |  |  |  +--ro ipv4-address?            inet:ipv4-address
                  |  |  +--:(ipv6-address)
                  |  |     +--ro ipv6-address?            inet:ipv6-address
                  |  +--ro nat-keepalive-interval?   uint16
                  +--ro nat-detection-destination-ip {ikev2-init-nat-detection-
destination-
   ip}?
                  |  +--ro (ip-address)?
                  |  |  +--:(ipv4-address)
                  |  |  |  +--ro ipv4-address?            inet:ipv4-address
                  |  |  +--:(ipv6-address)
                  |  |     +--ro ipv6-address?            inet:ipv6-address
                  |  +--ro nat-keepalive-interval?   uint16
                  +--ro redirect-supported?                    boolean {ikev2-
init-redirect-
   supported}?
                  +--ro fragmentation-supported?               boolean {ikev2-
init-
   fragmentation-supported}?
                  +--ro mobike-supported?                      boolean {ikev2-
auth-mobike-
   supported}?
                  +--ro rohc-supported?                        boolean {ikev2-
auth-rohc-
```

```
   supported}?
                +--ro childless-ikev2-supported?            boolean {ikev2-
auth-childless-
   supported}?
                +--ro message-id-sync-supported?            boolean {ikev2-
auth-message-id-
   supported}?
                +--ro ipsec-replay-counter-sync-supported?  boolean {ikev2-
auth-ipsec-
   replay-counter-sync-supported}?
                +--ro erx-supported?                        boolean {ikev2-
auth-erx-
   supported}?
                +--ro clone-ike-sa-supported?               boolean {ikev2-
auth-clone-ike-
   sa-supported}?
```

4. IKEv2 Crypto YANG Module

This section will present the YANG data model for IKEv2 Crypto.

<CODE BEGINS> file "ietf-ikev2-crypto@2016-02-26.yang"

```
module ietf-ikev2-crypto {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ikev2-crypto";
  prefix ikev2-crypto;

  organization "Ericsson AB.
                Huawei Technologies India Pvt Ltd.";

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

  description
    "This YANG module defines the parameters"+
    " for IANA, Internet Key Exchange Version 2 (IKEv2)"+
    " Parameters."+
    " <http://www.rfc-editor.org/info/rfc4301>"+
    " Copyright (c) 2016 Ericsson AB."+
    " All rights reserved.";

  revision 2016-02-26 {
    description
      "First revision.";
    reference
      "RFC 7296: Internet Key Exchange Protocol Version 2.";
  }

  /*--------------------*/
  /* Typedefs           */
  /*--------------------*/

  /* IKEv2 Exchange Types (ET) */
  typedef ikev2-exchange-type-t {
    type enumeration {
      enum et-ike-sa-init {
        value 34;
        description
          "et-ike-sa-init - IKEv2 Exchange Types (ET)";
      }
      enum et-ike-auth {
        value 35;
        description
          "et-ike-auth - IKEv2 Exchange Types (ET)";
      }
      enum et-create-child-sa {
        value 36;
```

```
          description
            "et-create-child-sa - IKEv2 Exchange Types (ET)";
        }
        enum et-informational {
          value 37;
          description
            "et-informational - IKEv2 Exchange Types (ET)";
        }
        enum et-ike-session-resume {
          value 38;
          description
            "et-ike-session-resume - IKEv2 Exchange Types (ET)";
        }
        enum et-gsa-auth {
          value 39;
          description
            "et-gsa-auth - IKEv2 Exchange Types (ET)";
        }
        enum et-gsa-registration {
          value 40;
          description
            "et-gsa-registration - IKEv2 Exchange Types (ET)";
        }
        enum et-gsa-rekey {
          value 41;
          description
            "et-gsa-rekey - IKEv2 Exchange Types (ET)";
        }
      }
      description
        "IKEv2 Exchange Types (ET).";
    }

     /* Transform Type Values (TTV), RFC 7296 */
     typedef ikev2-transform-type-value-t {
      type enumeration {
        enum ttv-reserved-0 {
          value 0;
          description
            "ttv-reserved-0 - Transform Type Value (TTV)"+
            " Reserved ";
        }
        enum ttv-encr {
          value 1;
          description
            "ttv-encr - Transform Type Value 1 (TTV),"+
            " Encryption Algorithm "+
            "(ENCR) used in IKE and ESP.";
```

```
        }
```

```
        enum ttv-prf {
          value 2;
          description
            "ttv-prf - Transform Type Value 2 (TTV),"+
            " Pseudo-Random Function(PRF) used in IKE.";
        }
        enum ttv-integ {
          value 3;
          description
            "ttv-integ - Transform Type Value 3 (TTV),"+
            " Integrity Algorithm"+
            " (INTEG) used in IKE, AH, optional ESP.";
        }
        enum ttv-dh {
          value 4;
          description
            "ttv-dh - Transform Type Value 4 (TTV),"+
            " Diffie-Hellman (DH)"+
            " used in IKE, optional AH and ESP.";
        }
        enum ttv-esn {
          value 5;
          description
            "ttv-esn - Transform Type Value 5 (TTV),"+
            " Extended Sequence"+
            " Numbers (ESN) used in AH and ESP.";
        }
      }
      description
        "IKEv2 Transform Type Values ((TTV).";
    }

     /* IKEv2 Transform Attribute Types (TAT) */
     typedef ikev2-transform-attribute-type-t {
      type enumeration {
        enum tat-reserved-0 {
          value 0;
          description
            "tat-reserved-0 - IKEv2 Transform Attribute "+
            "Type (TAT) Reserved-0";
        }
        enum tat-reserved-1 {
          value 1;
          description
            "tat-reserved-1 - IKEv2 Transform Attribute "+
            "Type (TAT) Reserved-1";
        }
        enum tat-reserved-13 {
```

```
            value 13;
```

```
          description
            "ikev2-tat-reserved-13 - IKEv2 Transform Attribute "+
            "Type (TAT) Reserved-13";
        }
        enum tat-key-length {
          value 41;
          description
            "ikev2-tat-key-length - IKEv2 Transform Attribute "+
            "Type (TAT) KEY LENGTH (in bits)";
        }
      }
      description
        "IKEv2 Transform Attribute Types (TAT)";
    }

     /* Transform Type 1 (Encryption Algorithm) Transform IDs */
    typedef ikev2-encryption-algorithm-t {
      type enumeration {
        enum encr-reserved-0 {
          value 0;
          description
            "encr-reserved-0 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-des-iv4 {
          value 1;
          description
            "encr-des-iv4 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-des {
          value 2;
          description
            "encr-des - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-3des {
          value 3;
          description
            "encr-3des - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-rc5 {
          value 4;
          description
            "encr-rc5 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-idea {
          value 5;
          description
            "encr-idea - IKEv2 Encryption Algorithm Transform";
        }
```

```
          enum encr-cast {
```

```
          value 6;
          description
            "encr-cast - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-blowfish {
          value 7;
          description
            "encr-blowfish - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-3idea {
          value 8;
          description
            "encr-3idea - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-des-iv32 {
          value 9;
          description
            "encr-des-iv32 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-reserved-10 {
          value 10;
          description
            "encr-reserved-10 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-null {
          value 11;
          description
            "encr-null - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-aes-cbc {
          value 12;
          description
            "encr-aes-cbc - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-aes-ctr {
          value 13;
          description
            "encr-aes-ctr - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-aes-ccm-8 {
          value 14;
          description
            "encr-aes-ccm-8 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-aes-ccm-12 {
          value 15;
          description
```

"encr-aes-ccm-12 - IKEv2 Encryption Algorithm"+

```
            " Transform";
          }
          enum encr-aes-ccm-16 {
            value 16;
            description
              "encr-aes-ccm-16 - IKEv2 Encryption Algorithm"+
              " Transform";
          }
          enum encr-reserved-17 {
            value 17;
            description
              "encr-reserved-17 - IKEv2 Encryption Algorithm"+
              " Transform";
          }
          enum encr-aes-gcm-8-icv {
            value 18;
            description
              "encr-aes-gcm-8-icv - IKEv2 Encryption Algorithm"+
              " Transform";
          }
          enum encr-aes-gcm-12-icv {
            value 19;
            description
              "encr-aes-gcm-12-icv - IKEv2 Encryption Algorithm"+
              " Transform";
          }
          enum encr-aes-gcm-16-icv {
            value 20;
            description
              "encr-aes-gcm-16-icv - IKEv2 Encryption Algorithm"+
              " Transform";
          }
          enum encr-null-auth-aes-gmac {
            value 21;
            description
              "encr-null-auth-aes-gmac - IKEv2 Encryption Algorithm"+
              " Transform";
          }
          enum encr-ieee-p1619-xts-aes {
            value 22;
            description
              "encr-ieee-p1619-xts-aes - IKEv2 Encryption Algorithm"+
              " Transform IEEE P1619 XTS-AES.";
          }
          enum encr-camellia-cbc {
            value 23;
            description
              "encr-camellia-cbc - IKEv2 Encryption Algorithm"+
```

```
            " Transform";
```

```
        }
        enum encr-camellia-ctr {
          value 24;
          description
            "encr-camellia-ctr - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-camellia-ccm-8-icv {
          value 25;
          description
            "encr-camellia-ccm-8-icv - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-camellia-ccm-12-icv {
          value 26;
          description
            "encr-camellia-ccm-12-icv - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-camellia-ccm-16-icv {
          value 27;
          description
            "encr-camellia-ccm-16-icv - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-chacha20-poly1305 {
          value 28;
          description
            "encr-chacha20-poly1305 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-aes-cbc-128 {
          value 1024;
          description
            "encr-aes-cbc-128 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-aes-cbc-192 {
          value 1025;
          description
            "encr-aes-cbc-192 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-aes-cbc-256 {
          value 1026;
          description
            "encr-aes-cbc-256 - IKEv2 Encryption Algorithm Transform";
        }
        enum encr-blowfish-128 {
          value 1027;
```

description

```
            "encr-blowfish-128 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-blowfish-192 {
          value 1028;
          description
            "encr-blowfish-192 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-blowfish-256 {
          value 1029;
          description
            "encr-blowfish-256 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-blowfish-448 {
          value 1030;
          description
            "encr-blowfish-448 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-camellia-128 {
          value 1031;
          description
            "encr-camellia-128 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-camellia-192 {
          value 1032;
          description
            "encr-camellia-192 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
        enum encr-camellia-256 {
          value 1033;
          description
            "encr-camellia-256 - IKEv2 Encryption Algorithm"+
            " Transform";
        }
      }
      description
        "Transform Type 1 - IKEv2 Encryption Algorithm Transformm"+
        " IDs";
    }

    /* Transform Type 2 (Pseudo-Random Function PRF) Transform IDs */
    typedef ikev2-pseudo-random-function-t {
      type enumeration {
```

```
         enum prf-reserved-0 {
```

```
          value 0;
          description
            "prf-reserved-0 - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-hmac-md5 {
          value 1;
          description
            "prf-hmac-md5 - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-hmac-sha1 {
          value 2;
          description
            "prf-hmac-sha1 - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-hmac-tiger {
          value 3;
          description
            "prf-hmac-tiger - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-aes128-xcbc {
          value 4;
          description
            "prf-aes128-xcbc - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-hmac-sha2-256 {
          value 5;
          description
            "prf-hmac-sha2-256 - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-hmac-sha2-384 {
          value 6;
          description
            "prf-hmac-sha2-384 - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-hmac-sha2-512 {
          value 7;
          description
            "prf-hmac-sha2-512 - IKEv2 Pseudo-Random Function (PRF)";
        }
        enum prf-aes128-cmac {
          value 8;
          description
            "prf-aes128-cmac - IKEv2 Pseudo-Random Function (PRF)";
        }
      }
      description
        "Transform Type 2 - IKEv2 Pseudo-Random Function (PRF)"+
        " Transform IDs";
```

```
    }
```

```
     /* Transform Type 3 (Integrity Algorithm) Transform IDs */
    typedef ikev2-integrity-algorithm-t {
      type enumeration {
        enum auth-none {
          value 0;
          description
            "auth-none - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-md5-96 {
          value 1;
          description
            "auth-hmac-md5-96 - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-sha1-96 {
          value 2;
          description
            "auth-hmac-sha1-96 - IKEv2 Integrity Algorithm";
        }
        enum auth-des-mac {
          value 3;
          description
            "auth-des-mac - IKEv2 Integrity Algorithm";
        }
        enum auth-kpdk-md5 {
          value 4;
          description
            "auth-kpdk-md5 - IKEv2 Integrity Algorithm";
        }
        enum auth-aes-xcbc-96 {
          value 5;
          description
            "auth-aes-xcbc-96 - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-md5-128 {
          value 6;
          description
            "auth-hmac-md5-128 - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-sha1-160 {
          value 7;
          description
            "auth-hmac-sha1-160 - IKEv2 Integrity Algorithm";
        }
        enum auth-aes-cmac-96 {
          value 8;
          description
            "auth-aes-cmac-96 - IKEv2 Integrity Algorithm";
```

```
        }
```

```
        enum auth-aes-128-gmac {
          value 9;
          description
            "auth-aes-128-gmac - IKEv2 Integrity Algorithm";
        }
        enum auth-aes-192-gmac {
          value 10;
          description
            "auth-aes-192-gmac - IKEv2 Integrity Algorithm";
        }
        enum auth-aes-256-gmac {
          value 11;
          description
            "auth-aes-256-gmac - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-sha2-256-128 {
          value 12;
          description
            "auth-hmac-sha2-256-128 - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-sha2-384-192 {
          value 13;
          description
            "auth-hmac-sha2-384-192 - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-sha2-512-256 {
          value 14;
          description
            "auth-hmac-sha2-512-256 - IKEv2 Integrity Algorithm";
        }
        enum auth-hmac-sha2-256-96 {
          value 1024;
          description
            "auth-hmac-sha2-256-96 - IKEv2 Integrity Algorithm";
        }
      }
      description
        "Transform Type 3 - IKEv2"+
        " Integrity Algorithms Transform IDs";
    }

    /* Transform Type 4 (Diffie-Hellman Group) Transform IDs */
    typedef ikev2-diffie-hellman-group-t {
      type enumeration {
        enum dh-group-none {
          value 0;
          description
            "dh-group-none - IKEv2 Diffie-Hellman Group (DH)";
```

```
        }
```

```
          enum dh-modp-768-group-1 {
            value 1;
            description
              "dh-modp-768-group-1 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-modp-1024-group-2 {
            value 2;
            description
              "dh-modp-1024-group-2 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-modp-1536-group-5 {
            value 5;
            description
              "dh-modp-1536-group-5 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-modp-2048-group-14 {
            value 14;
            description
              "dh-modp-2048-group-14 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-modp-3072-group-15 {
            value 15;
            description
              "dh-modp-3072-group-15 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-modp-4096-group-16 {
            value 16;
            description
              "dh-modp-4096-group-16 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-modp-6144-group-17 {
            value 17;
            description
              "dh-modp-6144-group-17 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-modp-8192-group-18 {
            value 18;
            description
              "dh-modp-8192-group-18 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-recp-256-group-19 {
            value 19;
            description
              "dh-recp-256-group-19 - IKEv2 Diffie-Hellman Group (DH)";
          }
          enum dh-recp-384-group-20 {
            value 20;
            description
```

"dh-recp-384-group-20 - IKEv2 Diffie-Hellman Group (DH)";

```
        }
        enum dh-recp-521-group-21 {
          value 21;
          description
            "dh-recp-521-group-21 - IKEv2 Diffie-Hellman Group (DH)";
        }
        enum dh-modp-1024-160-pos-group-22 {
          value 22;
          description
            "dh-modp-1024-160-pos-group-22 - IKEv2 Diffie-Hellman"+
            " Group (DH)";
        }
        enum dh-modp-2048-224-pos-group-23 {
          value 23;
          description
            "dh-modp-2048-224-pos-group-23 - IKEv2 Diffie-Hellman"+
            " Group (DH)";
        }
        enum dh-modp-2048-256-pos-group-24 {
          value 24;
          description
            "dh-modp-2048-256-pos-group-24 - IKEv2 Diffie-Hellman"+
            " Group (DH)";
        }
        enum dh-recp-192-group-25 {
          value 25;
          description
            "dh-recp-192-group-25 - IKEv2 Diffie-Hellman Group (DH)";
        }
        enum dh-recp-224-group-26 {
          value 26;
          description
            "dh-recp-224-group-26 - IKEv2 Diffie-Hellman Group (DH)";
        }
        enum dh-brainpool-ip-224-r1 {
          value 27;
          description
            "dh-brainpool-ip-224-r1 - IKEv2 Diffie-Hellman Group"+
            " (DH)";
        }
        enum dh-brainpool-ip-256-r1 {
          value 28;
          description
            "dh-brainpool-ip-256-r1 - IKEv2 Diffie-Hellman Group"+
            " (DH)";
        }
        enum dh-brainpool-ip-384-r1 {
          value 29;
```

description

```
            "dh-brainpool-ip-384-r1 - IKEv2 Diffie-Hellman Group"+
            " (DH)";
        }
        enum dh-brainpool-ip-512-r1 {
          value 30;
          description
            "dh-brainpool-ip-512-r1 - IKEv2 Diffie-Hellman Group"+
            " (DH)";
        }
      }
      description
        "Transform Type 4 - IKEv2"+
        " Diffie-Hellman Groups (DH) Transform IDs";
    }
    /* Transform Type 5 (Extended Sequence Numbers ESN
       Transform IDs) */
    typedef ikev2-extended-sequence-number-t {
      type enumeration {
        enum esn-none {
          value 0;
          description
            "esn-none - IKEv2 Extended Sequence Number";
        }
        enum esn-1 {
          value 1;
          description
            "esn-1 - IKEv2 Extended Sequence Number";
        }
      }
      description
        "Transform Type 5 - IKEv2 Extended Sequence Number (ESN)";
    }
    typedef ikev2-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
        "IKEv2 Connection type for IKE session.";
    }
    typedef ikev2-transport-protocol-name-t {
      type enumeration {
        enum tcp {
          value 1;
          description
            "Transmission Control Protocol (TCP) Transport Protocol.";
        }
        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.";
    }

    typedef pad-type-t {
      type enumeration {
        enum id-ipv4-addr {
          value 1;
          description
```

```
            "A single four (4) octet IPv4 address";
```

```
        }
        enum id-fdqn {
          value 2;
          description
            "A fully-qualified domain name string.";
        }
        enum id-rfc822-addr {
          value 3;
          description
            "A fully-qualified RFC 822 email address string";
        }
        enum id-ipv6-addr {
          value 5;
          description
            "A single sixteen (16) octet IPv6 address";
        }
        enum id-der-asn1-dn {
          value 9;
          description
            "The binary Distinguished Encoding Rules (DER) encoding"+
            " of an ASN.1 X.500 Distinguished Name";
        }
        enum id-der-asn1-gn {
          value 10;
          description
            "The binary Distinguished Encoding Rules (DER) encoding"+
            " of an ASN.1 X.509 General Name";
        }
        enum id-key {
          value 11;
          description
            "Key ID (exact match only). An opaque octet stream that"+
            " may be used to pass vendor-specific information"+
            " necessary to do certain proprietary types of"+
            " identification";
        }
        enum id-any {
          value 100;
          description
            "Optional: openIKEv2.conf";
        }
      }
      description
        "Peer Authorization  Database (PAD) Type";
    }

    typedef ikev2-protocol-identifiers-t {
      type enumeration {
```

```
          enum "reserved-0" {
```

```
          value 0;
          description
            "Reserved IKEv2 Security Protocol Identifier";
        }
        enum "ike" {
          value 1;
          description
            "Internet Key Exchange (IKE) Protocol Identifier";
        }
        enum "ah" {
          value 2;
          description
            "Authentication Header (AH) Protocol Identifier";
        }
        enum "esp" {
          value 3;
          description
            "Encapsulating Security Payload (ESP) Protocol"+
            " Identifier";
        }
        enum "fc_esp_header" {
          value 4;
          description
            "Fibre Channel Encapsulating Security Payload Header";
        }
        enum "fc_ct_authentication" {
          value 5;
          description
            "Fibre Channel Common Transport Authentication";
        }
      }
      description
        "IKEv2 Security Protocol Identifiers";
    }

    typedef ikev2-authentication-method-t {
      type enumeration {
        enum auth-preshared {
          value 0;
          description
            "authorization preshared - IKEv2 Authentication Method";
        }
        enum rsa-digital-signature {
          value 1;
          description
            "rsa-digital-signature - IKEv2 Authentication Method";
        }
        enum shared-key-msg-integrity-code {
```

```
            value 2;
```

```
          description
            "shared-key-msg-integrity-code - IKEv2 Authentication"+
            " Method";
        }
        enum dss-digital-signature {
          value 3;
          description
            "dss-digital-signature - IKEv2 Authentication Method";
        }
        enum ecdsa-sha-256-p256-curve {
          value 9;
          description
            "ecdsa-sha-256-p256-curve - IKEv2 Authentication Method";
        }
        enum ecdsa-sha-384-p384-curve {
          value 10;
          description
            "ecdsa-sha-384-p384-curve - IKEv2 Authentication Method";
        }
        enum ecdsa-sha-512-p512-curve {
          value 11;
          description
            "ecdsa-sha-512-p512-curve - IKEv2 Authentication Method";
        }
        enum generic-secure-passwd-auth-method {
          value 12;
          description
            "generic-secure-passwd-auth-method - IKEv2"+
            " Authentication Method";
        }
        enum null-auth-method {
          value 13;
          description
            "null-auth-method - IKEv2 Authentication Method";
        }
        enum digital-signature {
          value 14;
          description
            "digital-signature - IKEv2 Authentication Method";
        }
      }
      description "IKEv2 Authentication Methods";
    }

    typedef ikev2-traffic-selector-types-t {
      type enumeration {
        enum "ts-ipv4-addr-range" {
          value 7;
```

description

```
            "ts-ipv4-addr-range - IKEv2 Traffic Selector Type (TS)";
        }
        enum "ts-ipv6-addr-range" {
          value 8;
          description
            "ts-ipv6-addr-range - IKEv2 Traffic Selector Type (TS)";
        }
        enum "ts-fc-addr-range" {
          value 9;
          description
            "ts-fc-addr-range - IKEv2 Traffic Selector Type (TS)";
        }
      }
      description
        "IKEv2 Traffic Selector Types";
    }

    typedef ikev2-cert-encoding-t {
      type enumeration {
        enum cert-pkcs-7-wrapped-x509 {
          value 1;
          description
            "PKCS #7 wrapped X.509 certificate";
        }
        enum cert-pgp {
          value 2;
          description
            "PGP Certificate";
        }
        enum cert-dns-signed-key {
          value 3;
          description
            "DNS Signed Key";
        }
        enum cert-x509-signature {
          value 4;
          description
            "X.509 Certificate - Signature";
        }
        enum cert-kerberos-token {
          value 6;
          description
            "Kerberos Token";
        }
        enum cert-revocation-list {
          value 7;
          description
            "Certificate Revocation List (CRL)";
```

```
        }
```

```
        enum cert-authority-revocation-list {
          value 8;
          description
            "Authority Revocation List (ARL)";
        }
        enum cert-spki {
          value 9;
          description
            "SPKI Certificate";
        }
        enum cert-x509-attribute {
          value 10;
          description
            "X.509 Certificate - Attribute";
        }
        enum cert-raw-rsa-key {
          value 11;
          description
            "Raw RSA Key";
        }
        enum cert-hash-url-x509 {
          value 12;
          description
            "Hash and URL of X.509 certificate";
        }
        enum cert-hash-url-x509-bundle {
          value 13;
          description
            "Hash and URL of X.509 bundle";
        }
        enum cert-ocsp-content {
          value 14;
          description
            "OCSP Content";
        }
        enum cert-raw-public-key {
          value 15;
          description
            "Raw Public Key";
        }
      }
      description
        "Type of Certificate Encoding";
    }
  }
```

<CODE ENDS>

5. IKEv2 YANG Module

This section will present the YANG data model for IKEv2.

<CODE BEGINS> file "ietf-ikev2@2016-03-10.yang"

```
module ietf-ikev2 {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ikev2";
  prefix "ikev2";

  import "ietf-ikev2-crypto" {
    prefix "ikev2-crypto";
  }

  import ietf-inet-types {
    prefix inet;
  }

  organization "Ericsson AB.
                Huawei Technologies India Pvt Ltd.";

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

  description
    "This YANG module defines the configuration and operational
     state data for Internet Key Exchange version 2 (IKEv2) on
     IETF draft.
     Copyright (c) 2016 Ericsson AB.
     All rights reserved.";

  revision 2016-03-10 {
    description
      "First revision.";
    reference
      "YANG Data model for Internet Protocol Security - IPSec.
       draft-tran-ipecme-yang-ipsec-00.
       draft-wang-ipsecme-ike-yang-00.
       draft-wang-ipsecme-ipsec-yang-00.";
  }

  /*--------------------*/
  /* Feature            */
  /*--------------------*/

  feature ikev2 {
    description
      "Feature IKEv2";
```

```
    }
    feature ikev2-transport {
      description
        "Common IKEv2 Transport attributes";
    }
    feature ikev2-transport-anti-replay-mechanism {
      description
        "Optional: Enable INVALID_MESSAGE_ID defines whether an"+
        " optional INVALID_MESSAGE_ID  Notify Payload is sent when"+
        " the IKEv2 message received is outside the Operational"+
        " Window Size";
    }
    feature ikev2-transport-enable-notify-invalid-msg-id {
      description
        "Feature IKEv2 Transport enable notify of invalid message id";

    }
    feature ikev2-transport-retransmission {
      description
        "Feature IKEv2 Transport retransmission";

    }
    feature ikev2-transport-cookie-mechanism {
      description
        "Feature IKEv2 Transport Cookie mechanism";

    }
    feature ikev2-init {
      description
        "Feature IKEv2 INIT";

    }
    feature ikev2-init-authorized-dh {
      description
        "Feature IKEv2 INIT authorized Diffie-Hellman (DH)";

    }
    feature ikev2-init-authorized-certification-auth {
      description
        "Feature IKEv2 INIT authorized certification author";

    }
    feature ikev2-init-nat-detection-src-ip {
      description
        "Feature IKEv2 INIT NAT Detection Source IP Address";

    }
    feature ikev2-init-nat-detection-destination-ip {
```

description

```
        "Feature IKEv2 INIT Detection Destination IP Address";

    }
    feature ikev2-init-redirect-supported {
      description
        "Feature IKEv2 INIT Redirect Supported";

    }
    feature ikev2-init-fragmentation-supported {
      description
        "Feature IKEv2 INIT Fragmentation Supported";

    }
    feature ikev2-init-responder-certreq {
      description
        "Feature IKEv2 INIT Responder CERTREQ";
    }
    feature ikev2-init-optional {
      description
        "Feature IKEv2 INIT Optional Attributes";
    }
    feature ikev2-auth-mobike-supported {
      description
        "Feature IKEv2 AUTH Mobike Supported";

    }
    feature ikev2-auth-rohc-supported {
      description
        "Feature IKEv2 AUTH RObust Header Compression ROHC Supported";

    }
    feature ikev2-auth-childless-supported {
      description
        "Feature IKEv2 AUTH Childless Supported";

    }
    feature ikev2-auth-message-id-supported {
      description
        "Feature IKEv2 AUTH Message ID supported";

    }
    feature ikev2-auth-ipsec-replay-counter-sync-supported {
      description
        "Feature IKEv2 AUTH IPSec Replay Counter Sync Supported";

    }
    feature ikev2-auth-erx-supported {
      description
        "Feature IKEv2 AUTH ERX Supported";
```

```
    }
    feature ikev2-auth-clone-ike-sa-supported {
      description
        "Feature IKEv2 AUTH Clone IKE-SA Supported";

    }

    feature ikev2-sa {
      description
        "Feature IKEv2 Security Association (SA)";
    }

    feature ikev2-auth {
      description
        "Feature IKEv2 AUTH";
    }

    feature ikev2-peer {
      description
        "Feature IKEv2 Peer";
    }

    feature ikev2-state {
      description
        "IKEv2 Operational State";
    }

    feature ikev2-proposal-state {
      description
        "IKEv2 Proposal Operational State";
    }

    feature ikev2-transport-state {
      description
        "IKEv2 Transport State";
    }


    /*--------------------*/
    /* Typedefs           */
    /*--------------------*/
    typedef ipsec-spi {
      type uint64 {
        range "1..max";
      }
      description
        "Security Parameter Index SPI";
    }
```

```
    typedef transport-protocol-name-t {
      type enumeration {
        enum tcp {
          value 1;
          description
            "Transmission Control Protocol (TCP) Transport Protocol.";
        }
        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 role-t {
      type enumeration {
        enum any {
          value 0;
          description
            "Role: Any";
        }
        enum initiator {
          value 1;
          description
            "Role: Initiator";
        }
        enum responder {
          value 2;
          description
            "Role: Responder";
        }
      }
```

description

```
        "Role Type";
    }

    typedef cryptographic-material-t {
      type enumeration {
        enum sk-d {
          value 0;
          description
            "SK_d";
        }
        enum sk-ai {
          value 1;
          description
            "SK_ai";
        }
        enum sk-ar {
          value 2;
          description
            "SK_ar";
        }
        enum sk-ei {
          value 3;
          description
            "SK_ei";
        }
        enum sk-er {
          value 4;
          description
            "SK_er";
        }
        enum sk-pi {
          value 5;
          description
            "SK_pi";
        }
        enum sk-pr {
          value 6;
          description
            "SK_pr";
        }
        enum skeyseed {
          value 7;
          description
            "SKEYSEED";
        }
        enum nonces {
          value 8;
          description
```

```
            "Nonces";
```

```
        }
      }
      description
        "Cryptographic Material Type";
    }

    typedef ikev2-proposal-number-ref {
      type leafref {
        path "/ikev2/init/proposal/number";
      }
      description
        "reference to IKEv2 proposal number";
    }

    typedef ikev2-transport-base-mjver-ref {
      type leafref {
        path "/ikev2/transport/base-info/major-version";
      }
      description
        "reference to IKEv2 Transport Base Information
        Major Version";
    }

    typedef ikev2-transport-base-mnver-ref {
      type leafref {
        path "/ikev2/transport/base-info/minor-version";
      }
      description
        "reference to IKEv2 Transport Base Information
        Minor Version";
    }

    typedef ikev2-transport-base-spi-gen-policy-ref {
      type leafref {
        path "/ikev2/transport/base-info/spi-generation-policy";
      }
      description
        "reference to IKEv2 Transport Base Information
        SPI Generation Policy";
    }

    typedef ikev2-transport-anti-replay-mechanism-window-size-ref {
      type leafref {
        path "/ikev2/transport/anti-replay-mechanism/window-size";
      }
      description
        "reference to IKEv2 Transport Anti Replay Mechanism
        Window Size";
```

```
    }
```

```
    typedef ikev2-transport-anti-replay-mechanism-enable-notify-ref {
      type leafref {
        path "/ikev2/transport/anti-replay-mechanism/"+
             "enable-notify-invalid-msg-id";
      }
      description
        "reference to IKEv2 Transport Anti Replay Mechanism
        Enable Notify Invalid Message ID";
    }

    /*--------------------*/
    /*   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 ip-address-grouping {
      description
        "IP Address grouping";

      choice ip-address {
        description
          "Choice of IPv4 or IPv6.";
        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:DB8:0:0:8:800:200C:417A .";
        }
      }
    }

    grouping certificate-auth-grouping {
      description
        "Certificate Authority";
      leaf cert-encoding {
        type ikev2-crypto:ikev2-cert-encoding-t;
        description
          "Certificate Authority Encoding";
      }
      leaf cert-value {
        type uint32;
        description
          "Certificate Authority value";
      }
    }

    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 transform-encr-algorithm-grouping {
      description
```

```
          "Transform Type 1, Encryption Algorithm";
```

```
      list transform-encr-algorithm {
        key "encr-algorithm key-length";
        leaf encr-algorithm {
          type ikev2-crypto:ikev2-encryption-algorithm-t;
          description
            "IKEv2 Transform Type 1, Encryption Algorithm";
        }
        leaf key-length {
          type uint32;
          description
            "IKEv2 Transform Type 1, key length for Encryption"+
            " Algorithm";
        }
        description
          "IKEv2 Transform Type 1, Encryption Algorithm";
      }
    }

    grouping transform-prf-algorithm-grouping {
      description
        "IKEv2 Transform Type 2, Pseudo-Random Function PRF";
      list transform-prf-algorithm {
        key "prf-algorithm key-length";
        leaf prf-algorithm {
          type ikev2-crypto:ikev2-pseudo-random-function-t;
          description
            "IKEv2 Transform Type 2, Pseudo-Random Function"+
            " (PRF) Algorithm";
        }
        leaf key-length {
          type uint32;
          description
            "IKEv2 Transform Type 2, key length for PRF";
        }
        description
          "IKEv2 Transform Type 2, Pseudo-Random Function PRF";
      }
    }

    grouping transform-integrity-algorithm-grouping {
      description
        "IKEv2 Transform Type 3, Integrity Algorithm";
      list transform-integrity-algorithm {
        key "integrity-algorithm key-length";
        leaf integrity-algorithm {
          type ikev2-crypto:ikev2-integrity-algorithm-t;
          description
```

```
            "IKEv2 Transform Type 3, Integrity Algorithm";
```

```
        }
        leaf key-length {
          type uint32;
          description
            "IKEv2 Transform Type 3, key length for Integrity"+
            " Algorithm";
        }
        description
          "IKEv2 Transform Type 3, Integrity Algorithm";
      }
    }

    grouping transform-dh-grouping {
      description
        "IKEv2 Transform Type 4, Diffie-Hellman Group (DH)";
      list transform-dh {
        key "dh key-length";
        leaf dh {
          type ikev2-crypto:ikev2-diffie-hellman-group-t;
          description
            "IKEv2 Transform Type 4, Diffie-Hellman Group (DH)";
        }
        leaf key-length {
          type uint32;
          description
            "IKEv2 Transform Type 4, key length for Diffie-Hellman"+
            " Group (DH)";
        }
        description
          "IKEv2 Transform Type 4, Diffie-Hellman Group (DH)";
      }
    }

    grouping ikev2-proposal-grouping {
      description
        "IKEv2 Proposal";
      list proposal {
        key "number";
        description
          "Configure IKEv2 proposal";
        uses name-grouping;
        uses transform-encr-algorithm-grouping;
        uses transform-prf-algorithm-grouping;
        uses transform-integrity-algorithm-grouping;
        uses transform-dh-grouping;
        leaf number {
          type uint32;
          description
```

```
            "specify the order the proposals are sent";
```

```
        }
        leaf protocol {
          type ikev2-crypto:ikev2-protocol-identifiers-t;
          description
            "IKEv2 Proposal Protocol Identifier";
        }
      }
    }

    grouping ikev2-retransmission-grouping {
      description
        "IKEv2 retransmission policy configuration";
      container retransmision {
        if-feature ikev2-transport-retransmission;
        leaf max-retries {
          type uint32;
          description
            "maximum retry when retransmission failed";
        }
        leaf initial-retransmission-timeout {
          type uint32;
          description
            "initial retransmission timeout value";
        }
        leaf retransmission-timeout-policy {
          type string;
          description
            "defines of the Retransmission Timeout should be"+
            " computed";
        }
        leaf max-response-buffer-timeout {
          type uint32;
          description
            "This timer set when the response buffer can be clean"+
            " when the message ID is not being updated. It value"+
            " is expected to be in the order of several minutes";
        }
        leaf keepalive-timeout {
          type uint32;
          description
            "Keep-alive timeout";
        }
        leaf nat-keepalive-timeout {
          type uint32;
          description
            "Network Address Translation (NAT) Keep-alive timeout";
        }
        description
```

```
            "IKEv2 retransmission policy configuration";
```

```
      }
    }

    grouping ikev2-cookie-mechanism-grouping {
      description
        "IKEv2 Cookie Mechanism";
      container cookie-mechanism {
        if-feature ikev2-transport-cookie-mechanism;
        leaf cookie-lifetime {
          type uint32;
          description
            "Cookie Lifetime";
        }
        leaf half-open-ike-sa-threshold {
          type uint32;
          description
            "Half-open IKE-SA Threshold";
        }
        description
          "IKEv2 Cookie Mechanism";
      }
    }

    grouping ikev2-auth-avail-signing-capabilities-grouping {
      description
        "IKEv2 AUTH Available Signing Capabilities";
      list avail-signing-capabilities {
        key "auth-method-name";
        description
          "availiable signing capabilities";
        leaf auth-method-name {
          type string;
          description
            "Authentication method name";
        }
        leaf auth-method {
          type ikev2-crypto:ikev2-authentication-method-t;
          description
            "type of authentication method";
        }
        leaf auth-material-data {
          type string;
          description
            "authentication material data";
        }
      }
    }
```

```
     grouping ikev2-cert-auth-grouping {
```

```
      description
        "IKEv2 AUTH Certificate Authentication";
      container cert-auth {
        description
          "Certificate authentication";
        leaf cert-auth-encoding {
          type ikev2-crypto:ikev2-cert-encoding-t;
          description
            "certificate authentication encoding";
        }
        leaf cert-auth-value {
          type uint32;
          description
            "certificate authentication value";
        }
      }
    }

    grouping ikev2-cert-authentication-material-grouping {
      description
        "IKEv2 CERT Authentication Material";
      leaf cert-authentication-type {
        type string;
        default "cert";
        description
          "CERT Authentication Type";
      }
      uses ikev2-cert-auth-grouping;
    }

    grouping ikev2-auth-avail-hash-capabilities-grouping {
      description
        "IKEv2 AUTH Available Hash Capabilities";
      list avail-hash {
        key "hash-method";
        description
          "available hash";
        leaf hash-method {
          type string;
          description
            "hash method";
        }
        leaf auth-hash-lifetime {
          type uint32;
          description
            "Authentication Hash lifetime";
        }
      }
```

```
    }
```

```
    grouping ikev2-auth-avail-signature-verification-grouping {
      description
        "IKEv2 AUTH Available Signature Verification";
      list avail-signature-verify {
        key "signature-id";
        description
          "available signature verification";
        leaf signature-id {
          type string;
          description
            "signature ID";
        }
        leaf signature-lifetime {
          type uint32;
          description
            "signature lifetime";
        }
      }
    }

    grouping local-id-grouping {
      description
        "IKEv2 AUTH Local ID";
      list local-id {
        key "host-id";
        description
          "list of Local ID";
        leaf host-id {
          type string;
          description
            "Local Host ID";
        }
        leaf preference {
          type string;
          description
            "Local Preference";
        }
        leaf id-type {
          type string;
          description
            "Local ID type";
        }
        leaf id-value {
          type string;
          description
            "ID value";
        }
```

```
        }
```

```
    }

    grouping ikev2-vendor-id-grouping {
      description
        "IKEv2 Vendor ID";
      leaf vendor-id {
        type uint64;
        description
          "IKEv2 Vendor ID";
      }
    }

    grouping ikev2-base-info-grouping {
      description
        "IKEv2 Base Information";
      container base-info {
        description
          "IKEv2 basic information";
        leaf major-version {
          type uint8;
          default 2;
          description
            "IKEv2 Major Version";
        }
        leaf minor-version {
          type uint8;
          default 0;
          description
            "IKEv2 Minor Version";
        }
        leaf spi-generation-policy {
          type string;
          description
            "SPI genration policy";
        }
      }
    }

    grouping ikev2-anti-replay-mechanism-grouping {
      description
        "IKEv2 Anti Replay Mechanism";
      container anti-replay-mechanism {
        leaf window-size {
          type uint32;
          default 1;
          description
            "Window Size defines how much parallel exchange can"+
            " be performed between the peers. By default this"+
```

```
" value is set to 1. When greater than 1, as defined"+
```

```
            " in [RFC7296] section 2.3, a SET_WINDOW_SIZE Notify"+
            " Payloads will be sent by the peer to agree withe the"+
            " other peer on the Window Size. After this exchange"+
            " succeeds, the operational attribute that defines"+
            " the Window Size used by the IKE_SA, will be updated"+
            " with the value agreed by the peers.";
        }
        leaf enable-notify-invalid-msg-id {
          if-feature ikev2-transport-enable-notify-invalid-msg-id;
          type empty;
          description
            "Optional Enable INVALID_MESSAGE_ID defines whether an"+
            " optional INVALID_MESSAGE_ID  Notify Payload is sent"+
            " when the IKEv2 message received is outside the"+
            " Operational Window Size.";
        }
        description
          "Anti Replay Mechanism describes when message should be"+
          " rejected or considered by the IKEv2 daemon. The anti"+
          " reply mechanism is defined for each session.";
      }
    }


    grouping ikev2-init-optional-grouping {
      description
        "IKEv2 INIT Optional";
      container optional {
        if-feature ikev2-init-optional;
        container nat-detection-source-ip {
          if-feature ikev2-init-nat-detection-src-ip;
          description
            "Optional support: for Network Address Translation (NAT)"+
            " Destination Source IP Address, sent during the"+
            " IKE_INIT";
          uses ip-address-grouping;
          leaf nat-keepalive-interval {
            type uint16 {
              range "5..300";
            }
            units "Seconds";
            default 20;
            description "NAT detected and keepalive interval";
          }
        }

        container nat-detection-destination-ip {
          if-feature ikev2-init-nat-detection-destination-ip;
```

description

```
            "Optional support: for Network Address Translation (NAT)"+
            " Detecttion Destination IP Address, sent during the"+
            " IKE_INIT";
          uses ip-address-grouping;
          leaf nat-keepalive-interval {
            type uint16 {
              range "5..300";
            }
            units "Seconds";
            default 20;
            description "NAT detected and keepalive interval";
          }
        }

        leaf redirect-supported {
          if-feature ikev2-init-redirect-supported;
          type boolean;
          default true;
          description
            "Optional support: for redirect supported, sent"+
            " during the IKE_INIT";
        }
        leaf fragmentation-supported {
          if-feature ikev2-init-fragmentation-supported;
          type boolean;
          default true;
          description
            "Optional support: for fragmentation supported"+
            " sent during the IKE_INIT";
        }
        leaf mobike-supported {
          if-feature ikev2-auth-mobike-supported;
          type boolean;
          default true;
          description
            "Optional support: for mobike supported, sent during"+
            " IKE-AUTH";
        }
        leaf rohc-supported {
          if-feature ikev2-auth-rohc-supported;
          type boolean;
          default true;
          description
            "Optional support: for RObust Header Compression (ROHC)"+
            " supported, sent during IKE-AUTH";
        }
        leaf childless-ikev2-supported {
          if-feature ikev2-auth-childless-supported;
```

```
            type boolean;
```

```
          default true;
          description
            "Optional support: for CHILDLESS_IKEV2_SUPPORTED,"+
            " sent during IKE-AUTH";
        }
        leaf message-id-sync-supported {
          if-feature ikev2-auth-message-id-supported;
          type boolean;
          default true;
          description
            "Optional support: for IKEV2_MESSAGE_ID_SYNC_SUPPORTED,"+
            " sent during IKE-AUTH";
        }
        leaf ipsec-replay-counter-sync-supported {
          if-feature ikev2-auth-ipsec-replay-counter-sync-supported;
          type boolean;
          default true;
          description
            "Optional support: for"+
            " IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED,"+
            " sent during IKE-AUTH";
        }
        leaf erx-supported {
          if-feature ikev2-auth-erx-supported;
          type boolean;
          default true;
          description
            "Optional support: for ERX_SUPPORTED,"+
            " sent during IKE-AUTH";
        }
        leaf clone-ike-sa-supported {
          if-feature ikev2-auth-clone-ike-sa-supported;
          type boolean;
          default true;
          description
            "Optional support: for CLONE_IKE_SA_SUPPORTED,"+
            " sent during IKE-AUTH";
        }
        description
          "IKEv2 INIT Optional Attributes";
      }
    }

    grouping ikev2-initiator-id-grouping {
      container initiator-id {
        leaf initiator-id-type {
          type ikev2-crypto:pad-type-t;
          description
```

```
            "Initiator ID Type";
```

```
        }
        leaf initiator-id {
          type string;
          description
            "Initiator ID";
        }
        description
          "Initiator ID";
      }
      description
        "Initiator ID";
    }

    grouping ikev2-responder-id-grouping {
      container responder-id {
        leaf responder-id-type {
          type ikev2-crypto:pad-type-t;
          description
            "Responder ID Type";
        }
        leaf responder-id {
          type string;
          description
            "Responder ID";
        }
        description
          "Responder ID";
      }
      description
        "Responder ID";
    }

    grouping ikev2-transport-grouping {
      description
        "IKEv2 Transport Attributes";
      container transport {
        if-feature ikev2-transport;
        description
          "Common IKEv2 transport attributes";

        uses ikev2-base-info-grouping;
        uses ikev2-anti-replay-mechanism-grouping;
        uses ikev2-retransmission-grouping;
        uses ikev2-cookie-mechanism-grouping;
        uses ikev2-vendor-id-grouping;
      } //  End of container transport
    }
```

```
     grouping ikev2-config-request-grouping {
```

```
      description
        "Optional Configuration Request";
      container config-request {
        uses ip-address-grouping;
        description
          "Optional Configuration Requester";
      }
    }

    grouping ikev2-config-responder-grouping {
      description
        "Optional Configuration Responder";
      container config-responder {
        uses ip-address-grouping;
        description
          "Optional Configuration Responder";
      }
    }

    grouping ikev2-init-grouping {
      description
        "IKEv2 INIT Attributes";
      container init {
        if-feature ikev2-init;
        description
          "configuration attributes for the IKE_INIT exchange";

        list authorized-dh {
          if-feature ikev2-init-authorized-dh;
          key "dhg key-length";
          leaf dhg {
            type ikev2-crypto:ikev2-diffie-hellman-group-t;
            description
              "IKEv2 Transform Type 4, Diffie-Hellman Group (DH)";
          }
          leaf key-length {
            type uint32;
            description
              "IKEv2 Transform Type 4, key length for Diffie-Hellman"+
              " Group (DH)";
          }
          description
            "IKEv2 INIT Authorized Diffie-Hellman";
        }

        uses ikev2-proposal-grouping;
        uses ikev2-init-optional-grouping;
```

```
          leaf auth-method {
```

```
            type ikev2-crypto:ikev2-authentication-method-t;
            default auth-preshared;
            description
              "The authentication method of IKEv2 peer";
          }

          container responder-certreq {
            if-feature ikev2-init-responder-certreq;
            uses certificate-auth-grouping;
            description
              "IKEv2 INIT Responder CERTREQ";
          }

          uses ikev2-config-request-grouping;
          uses ikev2-config-responder-grouping;

          list authorized-cert-auth {
            if-feature ikev2-init-authorized-certification-auth;
            key "cert-encoding";
            uses certificate-auth-grouping;
            description
              "IKEv2 Initiator authorized certification authorities";
          }
        } // end of container init
      }

      grouping ikev2-auth-grouping {
        description
          "IKEv2 AUTH Attributes";
        container auth {
          if-feature ikev2-auth;
          description
            "IKEv2 AUTH Exchange";
          uses ikev2-auth-avail-signing-capabilities-grouping;
          uses ikev2-cert-auth-grouping;
          uses ikev2-auth-avail-hash-capabilities-grouping;
          uses ikev2-auth-avail-signature-verification-grouping;
          uses local-id-grouping;
          container authorized-certificate-authority {
            uses certificate-auth-grouping;
            description
              "IKEv2 AUTH Authorized Certificate Authority";
          }
        } // End of container auth
      }
      grouping ikev2-proposal-state-components {
        description
          "IKEv2 Operational state";
```

```
       list proposal {
```

```
        if-feature ikev2-proposal-state;
        key "name";
        description
          "IKEv2 proposal operational data";
        uses name-grouping;

        leaf encryption-algorithm {
          type ikev2-crypto:ikev2-encryption-algorithm-t;
          description
            "Transform Type 1 - IKEv2 Encryption Algorithm";
        }
        leaf prf-algorithm {
          type ikev2-crypto:ikev2-pseudo-random-function-t;
          description
            "Transform Type 2 - IKEv2 Pseudo-Random Function (PRF)";
        }
        leaf integrity-algorithm {
          type ikev2-crypto:ikev2-integrity-algorithm-t;
          description
            "Transform Type 3 - IKEv2 Integrity Algorithms";
        }
        leaf dh-group {
          type ikev2-crypto:ikev2-diffie-hellman-group-t;
          mandatory true;
          description
            "Transform Type 4 - IKEv2 Diffie-Hellman group.";
        }
        leaf esn {
          type ikev2-crypto:ikev2-extended-sequence-number-t;
          description
            "Transform Type 5 - IKEv2 Extended Sequence Number (ESN)";
        }
      }
      leaf connection-type {
        type ikev2-crypto:ikev2-connection-type-t;
        description
          "define whether the corresponding IKEv2 SA is being used"+
          " as an initiator or as a responder or both";
      }
    }


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

    /* ------------------- */
    /* IKEv2 configuration */
```

```
    /* ------------------- */
```

```
     container ikev2 {
       if-feature ikev2;
       description
         "Configuration IPSec IKEv2";

       uses ikev2-transport-grouping;
       uses ikev2-init-grouping;

       container sa {
         if-feature ikev2-sa;
         description
           "IKEv2 Security Association";
         leaf role {
           type role-t;
           description
             "IKEv2 SA Role [any | initiator | responder]";
         }
         container local-ip-address {
           description
             "IKEv2 SA Local IP Address";
           uses ip-address-grouping;
         }
         container remote-ip-address {
           description
             "IKEv2 SA Remote IP Address";
           uses ip-address-grouping;
         }
         leaf cryptgraphic {
           type cryptographic-material-t;
           description
             "Cryptographic Material Type";
         }
         leaf lifetime {
           type uint32;
           description
             "lifetime for IKEv2 SAs
              0: for no timeout.
              300 .. 99999999:  IKEv2 SA lifetime in seconds.";
         }
         leaf proposal {
           type ikev2-proposal-number-ref;
           description
             "IKE proposal number referenced by IKE peer";
         }
         uses ikev2-base-info-grouping;
         uses ikev2-anti-replay-mechanism-grouping;

         list retransmistion-ctx {
```

```
            key "window-id";
```

```
            leaf window-id {
              type uint32;
              description
                "Window ID";
            }
            uses ikev2-retransmission-grouping;
            description
              "IKEv2 Security Association Retransmission CTX
              that contains the element to enable retransmission
              for all ongoing exchange";
          }
          uses ikev2-initiator-id-grouping;
          uses ikev2-responder-id-grouping;
          uses ikev2-cert-authentication-material-grouping;
          uses ikev2-vendor-id-grouping;
          list optional-ctx {
            key "window-id";
            description
              "Optional Security Association CTX";
            leaf window-id {
              type uint32;
              description
                "Window ID";
            }
            uses ikev2-init-optional-grouping;
          }
        } // end of container sa

        list peer {
          if-feature ikev2-peer;
          key "peer-address";
          description "IKEv2 peer information";
          leaf peer-address {
            type string;
            description
              "Peer address";
          }
          leaf role {
            type role-t;
            default any;
            description
              "Peer Role [any | initiator | responder]";
          }

          list peer-id-entries {
            key "peer-id peer-id-type";
            description "IKE peer information";
            leaf peer-id-type {
```

```
            type ikev2-crypto:pad-type-t;
```

```
            description
              "Peer ID Type";
          }
          leaf peer-id {
            type string;
            description
              "Peer ID";
          }
        } // End of peer-entries

        list session {
          key "session-label";
          description
            "List of session";
          leaf session-label {
            type string;
            description
              "Session Label";
          }
          uses ikev2-initiator-id-grouping;
          uses ikev2-responder-id-grouping;
          uses ikev2-transport-grouping;
          uses ikev2-init-grouping;
          uses ikev2-auth-grouping;
          uses ikev2-config-request-grouping;
          uses ikev2-config-responder-grouping;
        }

        leaf preshared-key {
          type string;
          description "Preshare key";
        }
        leaf nat-traversal {
          type boolean;
          default false;
          description
            "Enable/Disable Network Address Translation"+
            " (NAT) traversal";
        }
      } //End of peer


    } // End of ikev2



    /*-------------------------------------------------------------*/
    /*************         Operational State       *************/
```

```
    /*-------------------------------------------------------------*/
```

```
    /*--------------------------*/
    /* IKEv2 Operational State  */
    /*--------------------------*/
    container ikev2-state {
      if-feature ikev2-state;
      config "false";

      container transport-state {
        if-feature ikev2-transport-state;
        description
          "Common IKEv2 operational transport state";
        leaf major-version {
          type uint8;
          default 2;
          description
            "IKEv2 Major Version";
        }
        leaf minor-version {
          type uint8;
          default 0;
          description
            "IKEv2 Minor Version";
        }
        leaf spi-generation-policy {
          type string;
          description
            "SPI genration policy";
        }
        leaf exchange-type {
          type ikev2-crypto:ikev2-exchange-type-t;
          description
            "IKEv2 Exchange Type";
        }
        leaf flags {
          type uint8;
          description
            "indicate specific options that are set for message";
        }
      }

      list sa-state {
        key "initiator-spi responder-spi";
        description
          "IKEv2 Security Association (SA) Operational State";

        leaf initiator-spi {
          type ipsec-spi;
```

description

```
            "initiator Security Parameter Index (SPI)";
          }
          leaf responder-spi {
            type ipsec-spi;
            description
              "initiator Security Parameter Index (SPI)";
          }
          list retransmistion-ctx {
            key "window-id";
            leaf window-id {
              type uint32;
              description
                "Window ID";
            }
            uses ikev2-retransmission-grouping;
            description
              "IKEv2 Security Association Retransmission CTX
              that contains the element to enable retransmission
              for all ongoing exchange";
          }
          container anti-replay-mechanism {
            leaf window-size {
              type uint32;
              description
                "window size";
            }
            leaf peer-request-msg-id {
              type uint32;
              description
                "Peer Request Message ID";
            }
            leaf peer-response-msg-id {
              type uint32;
              description
                "Peer Response Message ID";
            }
            leaf local-request-msg-id {
              type uint32;
              description
                "Local Request Message ID";
            }
            leaf local-response-msg-id {
              type uint32;
              description
                "Local Response Message ID";
            }
            description
              "IKEv2 Anti Replay Mechanism Operational State";
```

```
        }
```

```
        uses ikev2-vendor-id-grouping;
        uses ikev2-initiator-id-grouping;
        uses ikev2-responder-id-grouping;
        uses ikev2-auth-grouping;
        leaf half-open-ike-sa-counter {
          type uint32;
          description
            "IKEv2 Cookie Mechanism Half-Open IKE-SA counter";
        }
        list optional-ctx {
          key "window-id";
          description
            "Optional Security Association CTX";
          leaf window-id {
            type uint32;
            description
              "Window ID";
          }
          uses ikev2-init-optional-grouping;
        }
      }
      description
        "Contain the operational data for IKEv2";
    }
  } /* module ietf-ikev2 */


  <CODE ENDS>
```

6. 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.

7. References

7.1. Normative References

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

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

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

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

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

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., Kivinen, T., "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, October 2014.

[RFC6071] Frankel, S., Krishnan, S., "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", February 2011.

7.2. Informative References

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, January 2011.

Authors' Addresses

Khanh Tran
Ericsson
300 Holger Way
San Jose, CA 95134
USA
Email: khanh.x.tran@ericsson.com

Daniel Migault
Ericsson
8500 Decarie Blvd
Montreal, Quebec H4P 2N2
CANADA
Email: daniel.migault@ericsson.com

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