module ieee802-dot1ae {
  yang-version 1.1;

  namespace "urn:ieee:std:802.1AE:yang:ieee802-dot1ae";
  prefix "dot1ae";

  import ieee802-dot1ae-types { prefix "dot1aetypes"; }
  import ietf-yang-types { prefix "yang"; }
  import ietf-interfaces { prefix "if"; }
  // import ietf-system { prefix "sys"; }
  import iana-if-type { prefix "ianaift"; }
  import ieee802-dot1x-types { prefix "dot1x-types"; }

  organization "Institute of Electrical and Electronics Engineers";
  contact
      "WG-URL: http://grouper.ieee.org/groups/802/1/
      WG-EMail: stds-802-1@ieee.org
      Contact: IEEE 802.1 Working Group Chair
      Postal: C/O IEEE 802.1 Working Group
      IEEE Standards Association
      445 Hoes Lane
      P.O. Box 1331
      Piscataway
      NJ 08855-1331
      USA

      E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG";
  description
      "The MAC security entity (SecY) MIB module. A SecY is a protocol
      shim providing MAC Security (MACsec) in an interface stack.

      Each SecY transmits MACsec protected frames on one or more Secure Channels
      (SCs) to each of the other SecYs attached to the same LAN and participating
      in the same Secure Connectivity Association (CA). The CA is a security
      relationship, that is established and maintained by key agreement protocols
      and supported by MACsec to provide full connectivity between its
      participants. Each SC provides unidirectional point to multipoint
      connectivity from one participant to all the others and is supported by a
      succession of similarly point to multipoint Secure Associations (SAs). The
      Secure Association Key (SAK) used to protect frames is changed as an SA is
      replaced by its (overlapping) successor so fresh keys can be used without
      disrupting a long lived SC and CA.

      Two different upper interfaces, a Controlled Port (for frames protected by
      MACsec, providing an instance of the secure MAC service) and an
      Uncontrolled Port (for frames not requiring protection, like the key
      agreement frames used to establish the CA and distribute keys) are
      associated with a SecY shim.

      | Controller Port Interface | Uncontrolled Port Interface |
      | ________________________ | ____________________________ |
      | Physical Interface |

      Example MACsec Interface Stack."

  revision 2019-03-26 {
    description
        "Updates based upon comment resolution on draft TBD ";
    reference
        "IEEE 802.1AE-2018, Media Access Control (MAC) Security.";"
/* ------------------------------------------
 * List of features that may be optionally
 * implemented/supported
 * ------------------------------------------
 */

/* ---------------------------------------------------
 * Group objects used by 802.1ae YANG module
 * ---------------------------------------------------
 */

grouping provided-interface-grouping {
  description
    "This holds statistics for the Provided interface ports both the
    controlled port and the uncontrolled port."
  leaf provided-interface {
    type dot1x-types:pae-if-index;
    //type if:interface-ref;
    config false;
    description
      "The controlled or uncontrolled Port for this
      Secy.";
    reference
      "IEEE 802.1AE-2018 Clause 10.7.4";
  }
  leaf mac-enabled {
    type boolean;
    config false;
    description
      "The mac-enabled parameter is TRUE if use of the service is
      permitted and is otherwise FALSE. The value of this parameter is
      determined by administrative controls specific to the entity
      providing the service.";
    reference
      "IEEE 802.1AE-2018 Clause 6.4";
  }
  leaf mac-operational {
    type boolean;
    config false;
    description
      "The mac-operational parameter is TRUE if, and only if, service
      requests can be made and service indications can occur.";
    reference
      "IEEE 802.1AE-2018 Clause 6.4";
  }
  leaf oper-point-to-point-mac {
    type boolean;
    config false;
    description
      "If the operPointToPointMAC parameter is TRUE, the service is used
      as if it provides connectivity to at most one other system; if
      FALSE, the service is used as if it can provide connectivity to a
      number of systems.";
    reference
      "IEEE 802.1AE-2018 Clause 6.5";
  }
  leaf admin-point-to-point-mac {
    type enumeration {
    enum force-true {
      value 1;
      description
        "If admin-point-to-point-mac is set to force-true
        oper-point-to-point-mac shall be TRUE, regardless of any

indications to the contrary generated by the service providing
entity.";
reference
"IEEE 802.1AE-2018 Clause 6.5";
}
enum force-false {
  value 2;
  description
  "If admin-point-to-point-mac is set to force-false
  oper-point-to-point-mac shall be FALSE."
  reference
  "IEEE 802.1AE-2018 Clause 6.5";
}
enum auto {
  value 3;
  description
  "If admin-point-to-point-mac is set to auto
  oper-point-to-point-mac is as currently determined by the
  service providing entity.";
  reference
  "IEEE 802.1AE-2018 Clause 6.5";
}
}
default "auto";
description
"Each service access point can make available status parameters that
reflect the point-to-point status for the service instance provided,
and that allow administrative control over the use of that
information. The adminPointToPointMAC parameter can take one of three
values.";
reference
"IEEE 802.1AE-2018 Clause 6.5";
}
/* common SC items */
grouping secy-secure-channel-grouping {
  description
  "The secy-secure-channel grouping contains configuration and
  state common to both transmit and receive SCs.";
  leaf sci {
    type dot1aetypes:sec-sci-type;
    description
    "Each SecY transmits frames conveying secure MAC Service requests of
    any given priority on a single SC. Each SC provides unidirectional
    point-to-multipoint communication, and it can be long lived, persisting
    through SAK changes. Each SC is identified by a Secure Channel
    Identifier (SCI) comprising a 48-bit MAC address concatenated with a
    16-bit Port Identifier.";
    reference
    "IEEE 802.1AE Clause 7.1.2 and figure 7.7";
  }
  leaf created-time {
    type yang:date-and-time;
    config false;
    description
    "This is the system time when the SC was created.";
    reference
    "IEEE 802.1AE-2018 Clause 10.7.12";
  }
  leaf started-time {
    type yang:date-and-time;
    config false;
    description
    "This is the system time when receiving last became True for the SC.";
leaf stopped-time {
  type yang:date-and-time;
  config false;
  description "This is the system time when receiving last became False for the SC."
  reference "IEEE 802.1AE-2018 Clause 10.7.12";
}

/* common SA items */
grouping secy-secure-association-grouping {
  description "The secy-secure-association grouping contains configuration and state common to both transmit and receive Security Associations(SAs).";
  leaf in-use {
    type boolean;
    config false;
    description "If inUse is True, and MAC_Operational is True for the Common Port, the SA can receive and transmit frames."
    reference "IEEE 802.1AE-2018 Clause 10.7.14, 10.7.23";
  }
  leaf ssci {
    type uint32;
    config false;
    description "Short Secure Channel Identifier for the Send and Transmit SA";
    reference "IEEE 802.1AE-2018 Clause 10.7.14, 10.7.23";
  }
  leaf next-pn {
    type dot1aetypes:sec-pn-type;
    config false;
    description "The Next Packet Number, one more than the highest PN conveyed in the SecTAG of successfully validates frames received on this SA."
    reference "IEEE 802.1AE-2018 Clause 10.7.14, 10.7.23";
  }
  leaf created-time {
    type yang:date-and-time;
    config false;
    description "This is the system time when the SA was created."
    reference "IEEE 802.1AE-2018 Clause 10.7.14, 10.7.23";
  }
  leaf started-time {
    type yang:date-and-time;
    config false;
    description "This is the system time when inUse last became True for the SA."
    reference "IEEE 802.1AE-2018 Clause 10.7.14";
  }
  leaf stopped-time {
    type yang:date-and-time;
    config false;
    description "This is the system time when inUse last became False for the SA.";
reference
    "IEEE 802.1AE-2018 Clause 10.7.14";
}

grouping set-def {
    leaf tc {
        type int8;
    }
}

/* ---------------------------------------------------
* Configuration objects used by 802.1ae YANG module
* --------------------------------------------------- *
*/

augment "/if:interfaces/if:interface" {
    when "if:type = 'ianaift:ethernetCsmacd'" {
        description
            "Augment interfaces with 802.1ae MACSec System specific configuration nodes."
    }
}

container secy {
    description
        "Augment interface with 802.1 SecY configuration nodes.";
    list secy {
        key "controlled-port-number";
        description
            "The management information for each SecY is indexed by controlled-portNumber within a SecY System. This containment relationship complements that specified in IEEE Std 802.1X, where the management information for each PAE is indexed by portNumber within a PAE System";
        reference
            "IEEE 802.1AE-2018 Clause 10.7. IEEE 802.1X-2010 Clause 12.9.2";
        leaf controlled-port-number {
            type dot1x-types:pae-if-index;
            //type dot1aetypes:sec-if-index-type;
            description
                "Controlled Port Number";
        }
    container verification {
        description
            "This is the Verification controls for validation and replay protect for a given secy.";
        reference
            "IEEE 802.1AE-2018 Clause 10.6";
        leaf max-receive-channels {
            type uint8;
            config false;
            description
                "Specifies Maximum Number of Receive Channels for a SecY";
            reference
                "IEEE 802.1AE-2018 Clause 10.7.7";
        }
        leaf max-receive-keys {
            type uint8;
            config false;
            description
                "Specifies Maximum Number of Receive Keys for a SecY";
            reference
                "IEEE 802.1AE-2018 Clause 10.7.7";
        }
        leaf validate-frames {
            type enumeration {
enum disabled {
  value 1;
  description
  "Frame Verification is disabled. Remove SecTAGs and ICVs (if present) from received frames.";
}
enum check {
  value 2;
  description
  "Frame Verification is enabled. Do not discard invalid frames.";
}
enum strict {
  value 3;
  description
  "Frame Verification is enabled and strictly enforced. Discard any invalid frames.";
}
enum null {
  value 4;
  description
  "No Frame Verification is performed, do not remove-secTags or ICVs";
}

default "strict";

description
"Controls the frame verification settings. If the management control validate-frames is not Strict, frames without a SecTAG are received, counted, and delivered to the Controlled Port; otherwise, they are counted and discarded. If validate-frames is Disabled, cryptographic validation is not applied to tagged frames, but frames whose original service user data can be recovered are delivered. Frames with a SecTAG that has the TCI E bit set but the C bit clear are discarded, as this reserved encoding is used to identify frames with a SecTAG that are not to be delivered to the Controlled Port. If validate-frames is Null, all received frames are delivered to the Controlled Port without modification, irrespective of the absence, presence, or validity of a SecTAG";
reference
"IEEE 802.1AE-2018 Clause 10.7.8";
}
leaf replay-protect{
  type boolean;
  default "true";
  description
  "If the Packet Number (PN) of the received frame is less than the lowest acceptable packet number for the SA, and replay-protect is enabled, the frame is discarded and the in-pkts-late counter incremented. The replayProtect and replayWindow controls allows replay protection to be disabled, to operate on a packet number window, or to enforce strict frame order. If replayProtect is set but the replayWindow is not zero, frames within the window can be received out of order; however, they are not replay protected.";
  reference
  "IEEE 802.1AE-2018 Clause 10.6.2, 10.4";
}
leaf replay-window{
  type uint32;
  default "0";
  description
  "Controls the replay-window size in packets that supports media access control methods and provider networks that can misorder frames with different priorities and/or addresses.";
  reference
leaf in-pkts-untagged {
  type yang:counter64;
  config false;
  description "The number of packets received without the MACsec tag (SecTAG) received while validate-frames was not strict.";
  reference "IEEE 802.1AE-2018 Clause 10.7.9";
}

leaf in-pkts-no-tag {
  type yang:counter64;
  config false;
  description "The number of packets received without the MACsec tag (SecTAG) discarded because validate-frames was set to strict.";
  reference "IEEE 802.1AE-2018 Clause 10.7.9";
}

leaf in-pkts-bad-tag {
  type yang:counter64;
  config false;
  description "The number of received packets discarded with an invalid MACsec tag (SecTAG), zero value PN, or invalid ICV.";
  reference "IEEE 802.1AE-2018 Clause 10.7.9";
}

leaf in-pkts-no-sa {
  type yang:counter64;
  config false;
  description "The number of received packets discarded with an unknown SCI or for an unused SA.";
  reference "IEEE 802.1AE-2018 Clause 10.7.9";
}

leaf in-pkts-no-sa-error {
  type yang:counter64;
  config false;
  description "The number of packets discarded because the received SCI is unknown or the SA is not in use.";
  reference "IEEE 802.1AE-2018 Clause 10.7.9";
}

leaf in-pkts-overrun {
  type yang:counter64;
  config false;
  description "The number of packets discarded because they exceeded cryptographic performance capabilities.";
  reference "IEEE 802.1AE-2018 Clause 10.7.9";
}

leaf in-octets-validated {
  type yang:counter64;
  config false;
  description "The number of plaintext octets recovered from packets that were integrity protected but not encrypted.";
  reference "IEEE 802.1AE-2018 Clauses 10.6, 10.6.3";
}
leaf in-octets-decrypted {
  type yang:counter64;
  config false;
  description
    "The number of plaintext octets recovered from packets
     that were integrity protected and encrypted.";
  reference
    "IEEE 802.1AE-2018 Clauses 10.6, 10.6.3";
}

list receive-sc {
  key "sci";
  config false;
  description
    "This is the Receive Security Channel Status for a given
     secure channel identifier.";
  reference
    "IEEE 802.1AE-2018 Clause 10.7.9";
  uses secy-secure-channel-grouping;
  leaf receiving {
    type boolean;
    config false;
    description
      "Receiving is True if in-use is True for any of the
       SAs for the SC, and False otherwise";
    reference
      "IEEE 802.1AE-2018 Clause 10.7.12";
  }
  leaf in-pkts-ok {
    type yang:counter64;
    config false;
    description
      "For this SC, the number of validated packets.";
    reference
      "IEEE 802.1AE-2018 Clause 10.6.5, 10.7.9";
  }
  leaf in-pkts-unchecked {
    type yang:counter64;
    config false;
    description
      "For this SC, the number of packets while validate-frames
       was disabled.";
    reference
      "IEEE 802.1AE-2018 Clause 10.6.5, 10.7.9";
  }
  leaf in-pkts-delayed {
    type yang:counter64;
    config false;
    description
      "For this SC, the number of received packets, with
       Packet Number (PN) lower than the lowest acceptable PN
       lowest-pn and replay-protect is false.";
    reference
      "IEEE 802.1AE-2018 Clause 10.6.5, 10.7.9";
  }
  leaf in-pkts-late {
    type yang:counter64;
    config false;
    description
      "For this SC, the number of discarded packets, because
       the Packet Number (PN) was lower than the lowest
       acceptable PN lowest-pn and replay-protect is true.";
    reference
      "IEEE 802.1AE-2018 Clause 10.7.9";
  }
  leaf in-pkts-invalid {
leaf in-pkts-not-valid {
  type yang:counter64;
  config false;
  description
    "For this SC, the number of packets that failed validation but
    could be received because validate-frames was 'check' and the
    data was not encrypted (so the original frame could be
    recovered)."
  reference
    "IEEE 802.1AE-2018 Clause 10.7.9";
}

list receive-sa {
  key "rxa";
  description
    " This is the Receive Security Association Status for this
    association";
  uses secy-secure-association-grouping;
  leaf rxa {
    type dot1aetypes:sec-an-type;
    description
      " The Association Number for this Receiving Security
      Association";
    reference
      "IEEE 802.1AE-2018 Clause 10.7.13";
  }

  leaf lowest-pn {
    type dot1aetypes:sec-pn-type;
    config false;
    description
      "The lowest acceptable packet number. A received frame
      with a lower PN is discarded if replay-protect
      is enabled.";
    reference
      "IEEE 802.1AE-2018 Clause 10.7.14";
  }

  leaf enable-receive {
    type boolean;
    description
      "When the SA is created, enable-receive and in-use are
      False and the SA cannot be used to receive frames. The
      SA shall be able to receive, and in-use shall be
      True, when enable-receive is set. The SA shall stop
      receiving, and in-use shall be False, when
      enable-receive is reset.";
    reference
      "IEEE 802.1AE-2018 Clause 10.7.15";
  }

  //// The following is confusing to me.
  leaf updt-next-pn {
    type dot1aetypes:sec-pn-type;
    config false;
    description
      "The value of next-pn shall be set to the greater of its

existing value and the supplied of updt-next-pn. Initially, following creation, the values of next-pn will have been set to the values supplied by KaY.

reference
"IEEE 802.1AE-2018 Clause 10.7.15"
}
leaf updt-lowest-pn {
  type dot1aetypes:sec-pn-type;
  config false;
  description
  "The value of lowest-pn shall be set to the greater of its existing value and the supplied of updt-lowest-pn. Initially, following creation, the values of lowest-pn will have been set to the values supplied by KaY."

reference
"IEEE 802.1AE-2018 Clause 10.7.15"
}
leaf key-identifier {
  type dot1aetypes:sec-key-identifier-type;
  config false;
  description
  "The key-identifier is an octet string, whose format and interpretation depends on the key agreement protocol in use. It does not contain any information about the SAK other than that explicitly chosen by the key agreement protocol to publicly identify the key. If MKA is being used, it is the 128-bit Key Identifier (KI) specified by IEEE Std 802.1X encoded in an octet string as specified by that standard"

reference
"IEEE 802.1AE-2018 Clause 10.7.14, Clauses 14.7, 14.8"
}
}
}
}
container generation {
  description
  "This is the Generation controls for given secy."

reference
"IEEE 802.1AE-2018 Clause 10.???"
leaf sci-base {
  type string;
  config false;
  description
  "This is the base for a set of secure channels Security Channel Identifier."

reference
"IEEE 802.1AE-2018 Clause 10.7.17"
}
leaf max-transmit-channels {
  type uint16;
  description
  "Number of Transmit Channels"
}
leaf max-transmit-keys {
  type uint16;
  description
  "Number of Transmit Keys"

reference
"IEEE 802.1AE-2018 Clause 10.7.16"
}

All channels or per channel?
leaf protect-frames {
  type boolean;
  default true;
description
"The protect-frames control is provided to facilitate deployment."

reference
"IEEE 802.1AE-2018 Clause 10.7.17"
}

leaf always-include-sci {
  type boolean;
  default false;
  description
  "Mandates inclusion of an explicit SCI in the SecTAG when transmitting protected frames."
  reference
  "IEEE 802.1AE-2018 Clauses 10.5.3, 10.7.17"
}

leaf use-es {
  type boolean;
  default false;
  description
  "Enables use of the ES bit in the SecTAG when transmitting protected frames."
  reference
  "IEEE 802.1AE-2018 Clauses 10.5.3, 10.7.17"
}

leaf use-scb {
  type boolean;
  default false;
  description
  "Enables use of the SCB bit in the SecTAG when transmitting protected frames."
  reference
  "IEEE 802.1AE-2018 Clauses 10.5.3, 10.7.17"
}

leaf including-sci {
  type boolean;
  config false;
  description
  "True if an explicit SCI is included in the SecTAG when transmitting protected frames."
  reference
  "IEEE 802.1AE-2018 Clauses 10.5.3, 10.7.17"
}

leaf out-pkts-untagged {
  type yang:counter64;
  config false;
  description
  "The number of packets transmitted without a SecTAG because protect-frames is configured false."
  reference
  "IEEE 802.1AE-2018 Clause 10.7.18"
}

leaf out-pkts-too-long {
  type yang:counter64;
  config false;
  description
  "The number of transmit packets discarded because their length is greater than the ifMtu of the Common Port."
  reference
  "IEEE 802.1AE-2018 Clause 10.7.18"
}

leaf out-octets-protected {
  type yang:counter64;
  config false;
  description
  "The number of plain text octets integrity protected but not
encrypted in transmitted frames.
reference
"IEEE 802.1AE-2018 Clause 10.7.9";
}
leaf out-octets-encrypted {
type yang:counter64;
config false;
description
"The number of plain text octets integrity protected and
encrypted in transmitted frames."
reference
"IEEE 802.1AE-2018 Clause 10.7.9";
}
container user-priority-0 {
description
"Each entry in the Traffic Class Table is a traffic class,
represented by an integer from 0 (default) through 7 that also
comprises the numeric value of the four most significant bits of
the Port Identifier component of the SCI for the selected SC";
reference
"IEEE 802.1AE-2018 Clause 10.7.17"
leaf traffic-class {
type uint8 {
  range "0..7";
}
default 0;
}
}
container user-priority-1 {
description
"Each entry in the Traffic Class Table is a traffic class,
represented by an integer from 0 (default) through 7 that also
comprises the numeric value of the four most significant bits of
the Port Identifier component of the SCI for the selected SC The
user priority associated with the incoming frame. This is used as
an index into the table. Each entry in the Traffic Class Table
is a traffic class, represented by an integer from 0 (default)
through 7 that also comprises the numeric value of the four most
significant bits of the Port Identifier component of the SCI for
the selected SC. The table index and its output both comprise 4
bits, representing both the priority (most significant three
bits) and drop eligible (least significant bit) of the user
priority and access priority.";
reference
"IEEE 802.1AE-2018 Clause 10.7.17"
leaf traffic-class {
type uint8 {
  range "0..7";
}
default 1;
}
}
container user-priority-2 {
description
"Each entry in the Traffic Class Table is a traffic class,
represented by an integer from 0 (default) through 7 that also
comprises the numeric value of the four most significant bits of
the Port Identifier component of the SCI for the selected SC";
reference
"IEEE 802.1AE-2018 Clause 10.7.17"
leaf traffic-class {
type uint8 {
  range "0..7";
}
container user-priority-3 {
  description
  "Each entry in the Traffic Class Table is a traffic class,
  represented by an integer from 0 (default) through 7 that also
  comprises the numeric value of the four most significant bits of
  the Port Identifier component of the SCI for the selected SC";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf traffic-class {
    type uint8 {
      range "0..7";
    }
    default 3;
  }
}

container user-priority-4 {
  description
  "Each entry in the Traffic Class Table is a traffic class,
  represented by an integer from 0 (default) through 7 that also
  comprises the numeric value of the four most significant bits of
  the Port Identifier component of the SCI for the selected SC";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf traffic-class {
    type uint8 {
      range "0..7";
    }
    default 4;
  }
}

container user-priority-5 {
  description
  "Each entry in the Traffic Class Table is a traffic class,
  represented by an integer from 0 (default) through 7 that also
  comprises the numeric value of the four most significant bits of
  the Port Identifier component of the SCI for the selected SC";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf traffic-class {
    type uint8 {
      range "0..7";
    }
    default 5;
  }
}

container user-priority-6 {
  description
  "Each entry in the Traffic Class Table is a traffic class,
  represented by an integer from 0 (default) through 7 that also
  comprises the numeric value of the four most significant bits of
  the Port Identifier component of the SCI for the selected SC";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf traffic-class {
    type uint8 {
      range "0..7";
    }
  }
}
default 6;
}

container user-priority-7 {
  description "Each entry in the Traffic Class Table is a traffic class, represented by an integer from 0 (default) through 7 that also comprises the numeric value of the four most significant bits of the Port Identifier component of the SCI for the selected SC. Each entry in the Traffic Class Table is a traffic class, represented by an integer from 0 (default) through 7 that also comprises the numeric value of the four most significant bits of the Port Identifier component of the SCI for the selected SC. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority. The default value of each table entry is that of its index, thus leaving the priority and drop_eligible bits";
  reference "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf traffic-class {
    type uint8 {
      range "0..7";
    }
    default 7;
  }
}

container user-pcp-0 {
  description "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority."
  reference "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 0;
  }
}

container user-pcp-1 {
  description "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority."
  reference "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 1;
  }
container user-pcp-2 {
  description
  "The SecY may map the user priority of each frame's transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority.";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 2;
  }
}

container user-pcp-3 {
  description
  "The SecY may map the user priority of each frame's transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority.";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 3;
  }
}

container user-pcp-4 {
  description
  "The SecY may map the user priority of each frame's transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority.";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 4;
  }
}

container user-pcp-5 {
  description
  "The SecY may map the user priority of each frame's transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority.";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 5;
  }
}
(most significant three bits) and drop eligible (least significant bit) of the user priority and access priority.

reference
"IEEE 802.1AE-2018 Clause 10.7.17"

leaf access-priority {
  type uint8 {
    range "0..15";
  }
  default 5;
}
}

container user-pcp-6 {
  description
  "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop eligible (least significant bit) of the user priority and access priority."

  reference
  "IEEE 802.1AE-2018 Clause 10.7.17"

  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 6;
  }
}

container user-pcp-7 {
  description
  "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop eligible (least significant bit) of the user priority and access priority."

  reference
  "IEEE 802.1AE-2018 Clause 10.7.17"

  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 7;
  }
}

container user-pcp-8 {
  description
  "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop eligible (least significant bit) of the user priority and access priority."

  reference
  "IEEE 802.1AE-2018 Clause 10.7.17"

  leaf access-priority {
    type uint8 {
      range "0..15";
    }

container user-pcp-9 {
  description
  "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority.";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 9;
  }
}

container user-pcp-10 {
  description
  "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority.";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 10;
  }
}

container user-pcp-11 {
  description
  "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority.";
  reference
  "IEEE 802.1AE-2018 Clause 10.7.17";
  leaf access-priority {
    type uint8 {
      range "0..15";
    }
    default 11;
  }
}

container user-pcp-12 {
  description
  "The SecY may map the user priority of each frame’s transmit request at the Controlled Port to the access priority to be used
for the corresponding transmit request at the Common Port using
the Access Priority Table. The table index and its output
both comprise 4 bits, representing both the priority
(most significant three bits) and drop_eligible (least
significant bit) of the user priority and access
priority.";
reference
"IEEE 802.1AE-2018 Clause 10.7.17";
leaf access-priority {
    type uint8 {
        range "0..15";
    }
    default 12;
}
}
container user-pcp-13 {
    description
    "The SecY may map the user priority of each frame's transmit
    request at the Controlled Port to the access priority to be used
    for the corresponding transmit request at the Common Port using
    the Access Priority Table. The table index and its output
    both comprise 4 bits, representing both the priority
    (most significant three bits) and drop_eligible (least
    significant bit) of the user priority and access
    priority.";
    reference
    "IEEE 802.1AE-2018 Clause 10.7.17";
    leaf access-priority {
        type uint8 {
            range "0..15";
        }
        default 13;
    }
}
container user-pcp-14 {
    description
    "The SecY may map the user priority of each frame's transmit
    request at the Controlled Port to the access priority to be used
    for the corresponding transmit request at the Common Port using
    the Access Priority Table. The table index and its output
    both comprise 4 bits, representing both the priority
    (most significant three bits) and drop_eligible (least
    significant bit) of the user priority and access
    priority.";
    reference
    "IEEE 802.1AE-2018 Clause 10.7.17";
    leaf access-priority {
        type uint8 {
            range "0..15";
        }
        default 14;
    }
}
container user-pcp-15 {
    description
    "The SecY may map the user priority of each frame's transmit
    request at the Controlled Port to the access priority to be used
    for the corresponding transmit request at the Common Port using
    the Access Priority Table. The table index and its output
    both comprise 4 bits, representing both the priority
    (most significant three bits) and drop_eligible (least
    significant bit) of the user priority and access
    priority.";
    reference
    "IEEE 802.1AE-2018 Clause 10.7.17";
leaf access-priority {
    type uint8 {
        range "0..15";
    }
    default 15;
}

list transmit-sc {
    key "sci";
    config false;
    description "This is the transmit Security Channel, status for a given Security Channel Identifier.";
    reference "IEEE 802.1AE-2018 Clause 10.7.1";
    uses secy-secure-channel-grouping;
    leaf transmitting {
        type boolean;
        config false;
        description "True if in-use is True for any of the SAs for the SC, and False otherwise";
        reference "IEEE 802.1AE-2018 Clause 10.7.21";
    }
    leaf encoding-sa {
        type dot1aetypes:sec-an-type;
        config false;
        description "The current value of the encoding-sa variable for the selected transmit SC.";
        reference "IEEE 802.1AE-2018 Clause 10.7.24";
    }
    leaf out-pkts-protected { /* recommended in secyTcMIBCompliance */
        type yang:counter64;
        config false;
        description "The number of integrity protected but not encrypted packets for this transmit SC.";
        reference "IEEE 802.1AE Clause 10.7.18, Figure 10-3";
    }
    leaf out-pkts-encrypted { /* recommended in secyTcMIBCompliance */
        type yang:counter64;
        config false;
        description "The number of integrity protected and encrypted packets for this transmit SC.";
        reference "IEEE 802.1AE Clause 10.7.18, Figure 10-3";
    }
}

list transmit-sa {
    key txa;
    config false;
    description "This is the transmit security association status for a given association number.";
    uses secy-secure-association-grouping;
    leaf txa {
        type dot1aetypes:sec-an-type;
        config false;
        description "The association number for the SA";
        reference
leaf confidentiality {
    type boolean;
    config false;
    description "True if the SA provides confidentiality as well as integrity for transmitted frames."
    reference "IEEE 802.1AE-2018 Clause 10.7.23"
}

leaf key-identifier {
    type dot1aetypes:sec-key-identifier-type;
    config false;
    description "The key-identifier is an octet string, whose format and interpretation depends on the key agreement protocol in use. It does not contain any information about the SAK other than that explicitly chosen by the key agreement protocol to publicly identify the key. If MKA is being used, it is the 128-bit Key Identifier (KI) specified by IEEE Std 802.1X encoded in an octet string as specified by that standard"
    reference "IEEE 802.1AE-2018 Clause 10.7.14, Clauses 14.7, 14.8"
}

carrier current-cipher-suite {
    description ""
    leaf cipher-suite-identifier {
        type dot1aetypes:sec-eui64-type;
        //must "boolean(/../../cipher-suites/cipher-suite=.)"
        description "The Cipher Suite currently used by this SecY."
        reference "IEEE 802.1AE-2018 Clause 10.7.27"
    }
    list data-key {
        key keys;
        description ""
        leaf keys {
            type uint32;
            description "Numeric key number used as index"
            reference "IEEE 802.1AE-2018 Clause 10.7.27"
        }
        leaf key-identifier {
            type dot1aetypes:sec-key-identifier-type;
            config false;
            description "Key Identifier (KI), comprising the Key Server's MI (providing the more significant bits) and a 32-bit Key Number (KN) assigned by that Key Server (sequentially, beginning with 1). Each KI is used to identify the corresponding SAK for the purposes of SAI assignment, and appears in the clear in MKPDU's, so network management equipment and personnel can observe and diagnose MKA operation (if necessary) without having access to any secret key."
            reference "IEEE 802.1AE-2018 Clause 10.7.28"
leaf transmits {
    type boolean;
    config false;
    description "Transmits true means key is used for transmitting direction ???");
    reference "IEEE 802.1AE-2018 Clause 10.5";
}

leaf receives {
    type boolean;
    config false;
    description "Receives true means key is used for receiving direction ???");
    reference "IEEE 802.1AE-2018 Clause 10.5";
}

container controlled-interface {
    description "Controlled interface control and status";
    uses provided-interface-grouping;

    leaf controlled-port-enabled {
        type boolean;
        config false;
        description "By setting ControlledPortEnabled False, the KaY can prohibit use of the Controlled Port until the secure connectivity required has been configured.";
        reference "IEEE 802.1AE-2018 Clause 10.7.6";
    }
}

container uncontrolled-interface {
    description "Uncontrolled interface control and status";
    uses provided-interface-grouping;
}

/* See IEEE 802.1AE-2018 Clause 10.4. The common port is an instance of a MAC Internal Sublayer Service (ISS), which is defined outside of macsec (defined in 802.1d). Assume that the operational state and statistics are already implemented in a yang model for 802.1d and that an pae-if-ref is sufficient. */

container common-port {
    description "This list the statistics for the Provided interface ports both the controlled port and the uncontrolled port.";

    leaf common-port {
        type dot1x-types:pae-if-index;
        config false;
        description "The common Port for this Secy.";
        reference "IEEE 802.1AE-2018 Clause 10.7.4";
    }
}

list cipher-suite-control {
    key implemented-cipher-suite;
    leaf implemented-cipher-suite {
type dotlaetypes:sec-eui64-type;
//must "boolean(/../i../cipher-suites/cipher-suite=.)"

description
"cipher suite identifier (EUI-64)"
reference
"IEEE 802.1AE-2018 Clause 10.7.26"
}
leaf enable-use {
  type boolean;
  default true;
  description
  "Enables use of the Cipher Suite by this SecY."
  reference
  "IEEE 802.1AE-2018 Clause 10.7.26"
}
leaf require-confidentiality {
  type boolean;
  default true;
  description
  "This value is true if the Cipher Suite can only be used to provide
  both confidentiality and integrity (and not integrity only, or
  confidentiality with an offset)Enables use of the Cipher Suite by
  this SecY."
  reference
  "IEEE 802.1AE-2018 Clause 10.7.26"
}

description ""
}
}

list cipher-suites {
  key "cipher-suite"
  leaf cipher-suite {
    type dotlaetypes:sec-eui64-type;
    description
    "A globally unique 64-bit (EUI-64) identifier for this
    cipher suite"
    reference
    "IEEE 802.1AE-2018 Clause 10.7.25"
  }
  leaf name {
    type string {
      length "1..254"
    }
    config false;
    description
    "Cipher Suite Name, a human readable and displayable UTF-8
    (IETF RFC 2279) string."
    reference
    "IEEE 802.1AE-2018 Clause 10.7.25"
  }
  leaf integrity-protection {
    type boolean;
    config false;
    description
    "True if integrity protection without confidentiality
    can be provided.";
    reference
    "IEEE 802.1AE-2018 Clause 10.7.25"
  }
  leaf confidentiality-protection {
    type boolean;
    config false;
    description
"True if confidentiality with integrity protection can be provided."
reference
"IEEE 802.1AE-2018 Clause 10.7.25"
}
leaf offset-confidentiality {
type boolean;
config false;
description
"True if a selectable offset for confidentiality can be provided"
reference
"IEEE 802.1AE-2018 Clause 10.7.25"
}
leaf changes-data-length {
type boolean;
config false;
description
"Indicates that the cipher suite changes the data length."
reference
"IEEE 802.1AE-2018 Clause 10.7.25"
}
leaf icv-length {
type uint16;
config false;
description
"The number of octets in the ICV"
reference
"IEEE 802.1AE-2018 Clause 10.7.25"
}
description
"A list of configuration parameters and operational state associated with a cipher suite."

* Interfaces

augment "/if:interfaces/if:interface" {
when "if:type = 'ianaift:ethernetCsmacd'" {
description
"Applies to the Controlled Port of SecY Ethernet related Interface."
}
description
"Augment interface model with Secy configuration and operational nodes."
reference
""
container controlled-port {
description
"Controlled interface control and status"
leaf controlled-port-number {
    type dot1x-types:pae-if-index;
description
"Used to reference configured controlled port."
}
uses provided-interface-grouping;

leaf controlled-port-enabled {
type boolean;
config false;
description
"
"By setting ControlledPortEnabled False, the KaY can prohibit use of the Controlled Port until the secure connectivity required has been configured."

reference
"IEEE 802.1AE-2018 Clause 10.7.6"

]}
	augment "/if:interfaces/if:interface" {
when "if:type = 'ianaift:ethernetCsmacd'" {

description
"Applies to the Controlled Port of SecY Ethernet related Interface.";
}
}

type container uncontrolled-port {

description
"Uncontrolled interface control and status";
leaf controlled-port-number {

type dot1x-types:pae-if-index;

description
"Used to reference configured controlled port.";
}
uses provided-interface-grouping;
}

/* See IEEE 802.1AE-2018 Clause 10.4. The common port is an instance of a MAC Internal Sublayer Service (ISS), which is defined outside * of macsec (defined in 802.1d). Assume that the operational state and * statistics are already implemented in a yang model for 802.1d and that * an pae-if-ref is sufficient. */

augment "/if:interfaces/if:interface" {
when "if:type = 'ianaift:ethernetCsmacd'" {

description
"Applies to the Controlled Port of SecY Ethernet related Interface.";
}
}

container common-port {

description
"This list the statistics for the Provided interface ports both the controlled port and the uncontrolled port.";
leaf controlled-port-number {

type dot1x-types:pae-if-index;

description
"Used to reference configured controlled port.";
}
leaf common-port {

type dot1x-types:pae-if-index;
//type if:interface-ref;
config false;

description
"The common Port for this Secy.";
reference
"IEEE 802.1AE-2018 Clause 10.7.4"
}
leaf mac-enabled {

type boolean;
config false;

description
"The mac-enabled parameter is TRUE if use of the service is permitted and is otherwise FALSE. The value of this parameter is determined by administrative controls specific to the entity providing the service.";
reference
leaf mac-operational {
  type boolean;
  config false;
  description
    "The mac-operational parameter is TRUE if, and only if, service requests can be made and service indications can occur.";
  reference
    "IEEE 802.1AE-2018 Clause 6.4";
}
"IEEE 802.1AE-2018 Clause 6.4";