IEEE SA
STANDARDS ASSOCIATION

STANDARDS

IEEE Standard for Local and Metropolitan Area Networks—

Media Access Control (MAC) Security

Amendment 4: MAC Privacy Protection

IEEE Computer Society

Developed by the
LAN/MAN Standards Committee

IEEE Std 802.1AEdk™-2023
(Amendment to IEEE Std 802.1AE™-2018 as amended by
IEEE Std 802.1AE-2018/Cor 1-2020)

IEEE

IEEE Std 802.1AEdk-2023
(Amendment to IEEE Std 802.1AE™-2018 as amended by
IEEE Std 802.1AE-2018/Cor 1-2020)

IEEE Standard for Local and Metropolitan Area Networks—

Media Access Control (MAC) Security

Amendment 4: MAC Privacy Protection

Developed by

LAN/MAN Standards Committee
of the
IEEE Computer Society

Abstract: This amendment specifies a MAC Privacy protection encapsulating protocol and its use in conjunction with the MAC Security protocol (MACsec) to hide the source and destination MAC addresses of user data frames, and to reduce any correlation between observable frame sizes and transmission timing. This helps to protect user and application identities and to hide the purpose and content of communications. Management of MACsec and privacy protection is supported by YANG models and SNMP MIBs. Privacy considerations for bridged networks are reviewed.

Keywords: amendment, authorized port, bridged networks, confidentiality, corrigendum, data origin authenticity, EDEs, IEEE 802.1AE, integrity, LANs, local area networks, MAC Bridges, MAC security, MAC Service, MANs, metropolitan area networks, port based network access control, privacy, secure association, security, transparent bridging

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 978-1-5044-9950-7 STD26329
PDF: ISBN 978-1-5044-9951-4 STDPD26329

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system. An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website. Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this standard was completed, the IEEE 802.1 working group had the following membership:

Glenn Parsons, *Chair*
Jessy Royer, *Vice Chair*
Mick Seaman, *Security Task Group Chair*
Don Fedyk, *Editor*

Katsuyuki Akizuki
Konstantinos Alexandris
Venkat Arunarthi
Ralf Assmann
Huajie Bao
Rudy Belliardi
Jeremias Blendin
Christian Boiger
Paul Bottorff
Radhakrishna Canchi
Feng Chen
Abhijit Choudhury
Paul Congdon
Rodney Cummings
Josef Dorr
Hesham Elbakoury
Anna Engelmann
Thomas Enzinger
Janos Farkas
Norman Finn
Geoffrey Garner
Amrit Gopal
Craig Gunther
Marina Gutierrez
Stephen Haddock
Mark Hantel
Marc Holness
Daniel Hopf
Woojung Huh
Satoko Itaya
Yoshihiro Ito
Michael Karl
Stephan Kehrer
Marcel Kiessling
Gavin Lai
Yizhou Li
Joao Lopes
Lily Lv
Christophe Mangin
Scott Mansfield
Olaf Mater
David McCall
Larry McMillan
Martin Mittelberger
Hiroki Nakano
Takumi Nomura
Donald R. Pannell
Dieter Proell
Karen Randall
Maximilian Riegel
Silvana Rodrigues
Atsushi Sato
Frank Schewe
Maik Seewald
Ramesh Sivakolundu
Johannes Specht
Marius Stanica
Guenter Steindl
Nemanja Stamenic
Karim Traore
Max Turner
Balazs Varga
Ganesh Venkatesan
Tongtong Wang
Karl Weber
Leon Wessels
Ludwig Winkel
Jordon Woods
Takahiro Yamaura
Uwe Zeier
Nader Zein
William Zhao
HelgeZinner

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Thomas Alexander
Amelia Andersdotter
Carol Ansley
Butch Anton
Harry Bims
Reid Bowen
Vern Brethour
William Byrd
Paul Cardinal
Jose Castro
Pin Chang
Aditya Chaudhuri
Rodney Cummings
Janos Farkas
Donald Fedyk
Avraham Freedman
Devon Gayle
Stephen Haddock
Marek Hajduczenia
Mark Hamilton
Marco Hernandez
Werner Hoelzl
Yasuhiro Hyakutake
Raj Jain
Pranav Jha
Michael Johas Teener
Peter Jones
Lokesh Kabra
Piotr Karocki
Stuart Kerry
Yongbum Kim
David Kornbau
Mark Laubach
Hyeong Ho Lee
James Lepp
Scott Mansfield
Stephen McCann
N. Kishor Narang
Satoshi Obara
Karen Odonoghue
Ulf Parkholm
Glenn Parsons
Bansi Patel
Arumugam Paventhan
Petar Pepeljugoski
Dieter Proell
Karen Randall
Maximilian Riegel
Benjamin Rolfe
Jessy Rouyer
Frank Schewe
Mick Seaman
Eugene Stoudenmire
Walter Struppler
Mitsutoshi Sugawara
Max Turner
John Vergis
Lisa Ward
James Weaver
Stephen Webb
Scott Willy
Andreas Wolf
Peter Wu
Yu Yuan
Oren Yuen

When the IEEE SA Standards Board approved this standard on 5 June 2023, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Sara R. Biyabani	Joseph S. Levy	Paul Nikolich
Doug Edwards	Howard Li	Annette D. Reilly
Ramy Ahmed Fathy	Johnny Daozhuang Lin	Robby Robson
Guido R. Hiertz	Gui Lin	Lei Wang
Yousef Kimiagar	Xiaohui Liu	F. Keith Waters
Joseph L. Koepfinger*	Kevin W. Lu	Karl Weber
Thomas Koshy	Daleep C. Mohla	Philip B. Winston
John D. Kulick	Andrew Myles	Don Wright

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.1AEdk-2023, IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Security—Amendment 4: MAC Privacy Protection

The first edition of IEEE Std 802.1AE was published in 2006. A first amendment, IEEE Std 802.1AEbn™-2011, added the option of using the GCM-AES-256 Cipher Suite. A second, IEEE Std 802.1AEbw™-2013 added the GCM-AES-XPN-128 and GCM-AES-XPN-256 Cipher Suites. These extended packet numbering Cipher Suites allow more than 2^{32} frames to be protected with a single Secure Association Key (SAK) and so ease the timeliness requirements on key agreement protocols for very high speed (100 Gb/s plus) operation. A third amendment, IEEE Std 802.1AEcg™-2017, specified Ethernet Data Encryption devices (EDEs) that provide transparent secure connectivity while supporting provider network service selection and provider backbone network selection as specified in IEEE Std 802.1Q.

The second edition, IEEE Std 802.1AE-2018, consolidated the text of IEEE Std 802.1AE-2006 with its amendments. A fourth amendment IEEE Std 802.1AEdk™-2023 based on that edition specified MAC Privacy protection, and added YANG modules for existing and new functionality.

Relationship between IEEE Std 802.1AE and other IEEE Std 802 standards

IEEE Std 802.1X specifies Port-based Network Access Control, and provides a means of authenticating and authorizing devices attached to a LAN, and includes the MACsec Key Agreement protocol (MKA) necessary to make use of the MAC Security protocol (MACsec) specified in IEEE Std 802.1AE.

MACsec is not intended as a substitute for the security mechanisms specified by IEEE Std 802.11™ Wireless LAN Medium Access Control. That standard also uses IEEE Std 802.1X, thus facilitating the use of a common authentication and authorization framework for LAN media to which this standard applies and for Wireless LANs.

Contents

Figures

Tables

IEEE Standard for
Local and Metropolitan Area Networks —

Media Access Control (MAC) Security

Amendment 4:
MAC Privacy Protection

[This amendment is based on IEEE Std 802.1AE™-2018.]

NOTE—The editing instructions contained in this amendment define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italics***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using strikethrough (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Deletions and insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this note will not be carried over into future editions because the changes will be incorporated into the base standard.

1. Overview

1.1 Introduction

Change 1.1 as follows:

IEEE 802® Local Area Networks (LANs) are often deployed in networks that support mission-critical applications. These include corporate networks of considerable extent, and public networks that support many customers with different economic interests. The protocols that configure, manage, and regulate access to these networks typically run over the networks themselves. Preventing disruption and data loss arising from transmission and reception by unauthorized parties is highly desirable, since it is not practical to secure the entire network against physical access by determined attackers.

The MAC Security protocol (MACsec), as defined by this standard, allows authorized systems that attach to and interconnect LANs in a network to maintain confidentiality of transmitted data and to take measures against frames transmitted or modified by unauthorized devices.

MACsec facilitates

a) Maintenance of correct network connectivity and services

b) Isolation of denial of service attacks

c) Localization of any source of network communication to the LAN of origin

d) The construction of public networks, offering service to unrelated or possibly mutually suspicious customers, using shared LAN infrastructures

e) Secure communication between organizations, using a LAN for transmission

f) Incremental and non-disruptive deployment, protecting the most vulnerable network components

To deliver these benefits, MACsec has to be used in conjunction with appropriate policies for higher-level protocol operation in networked systems, an authentication and authorization framework, and network management. IEEE Std 802.1X™ provides authentication and cryptographic key distribution.[1]

MACsec protects communication between trusted components of the network infrastructure, thus protecting the network operation. MACsec cannot protect against attacks facilitated by the trusted components themselves, and is complementary to, rather than a replacement for, end-to-end application-to-application security protocols. The latter can secure application data independent of network operation, but cannot necessarily defend the operation of network components, or prevent attacks using unauthorized communication from reaching the systems that operate the applications.

MAC Privacy protection protocol, as defined by this standard, can be used in conjunction with MACsec to reduce the ability of adversaries to correlate the MAC addresses, sizes, and transmission timing of user data frames with individual persons, network applications, details of those applications, and levels of application activity.

[1] Information on other references can be found in Clause 2.

1.2 Scope

Change 1.2 as follows:

The scope of this standard is to specify provision of connectionless user data confidentiality, frame data integrity, and data origin authenticity by media access independent protocols and entities that operate transparently to MAC Clients.

NOTE—The MAC Clients are as specified in IEEE Std 802, IEEE Std 802.1Q™, and IEEE Std 802.1X™.[2]

To this end it

a) Specifies the requirements to be satisfied by equipment claiming conformance to this standard.

b) Specifies the requirements for MAC Security ~~MACsec~~ in terms of provision of the MAC Service and the preservation of the semantics and parameters of service requests and indications.

c) Describes the threats, both intentional and accidental, to correct provision of the service.

d) Specifies security services that prevent, or restrict, the effect of attacks that exploit these threats.

e) Examines the potential impact of both the threats and the use of MACsec on the Quality of Service (QoS), specifying constraints on the design and operation of MAC Security entities and protocols.

f) Models support of the secure MAC Service in terms of the operation of media access control method independent MAC Security Entities (SecYs) within the MAC Sublayer.

g) Specifies the format of the MACsec Protocol Data Unit (MPDUs) used to provide secure service.

h) Identifies the functions to be performed by each SecY, and provides an architectural model of its internal operation in terms of Processes and Entities that provide those functions.

i) Specifies each SecY's use of an associated and collocated Port Access Entity (PAE, IEEE Std 802.1X) to discover and authenticate MACsec protocol peers, and its use of that PAE's Key Agreement Entity (KaY) to agree and update cryptographic keys.

j) Specifies performance requirements and recommends default values and applicable ranges for the operational parameters of a SecY.

k) Specifies how SecYs are incorporated within the architecture of end stations, bridges, and two-port Ethernet Data Encryption devices (EDEs).

l) Establishes the requirements for management of MAC Security, identifying the managed objects and defining the management operations for SecYs.

m) Specifies ~~the~~ a Management Information Base (MIB) module for SecY management ~~managing the operation of MAC Security in TCP/IP networks~~.

n) Specifies a YANG configuration and operational state model for SecY management.

o) Specifies requirements, criteria, and choices of Cipher Suites for use with this standard.

p) Describes threats to individual privacy that can result from an adversary's observation of individual frames, even if those frames are integrity protected and their data confidentiality protected.

q) Models support of a privacy protected secure MAC Service in terms of the operation of MAC Privacy protection Entities (PrYs) that encapsulate user data frames in MAC Privacy protection Protocol Data Units (MPPDUs) to hide the user source and destination MAC addresses and to reduce any correlation of the sizes and transmission timing of frames with user identities and communication purposes, applications, or content.

r) Specifies the addressing, encoding, and decoding of MPPDUs.

s) Identifies the functions to be performed by each PrY, and provides an architectural model of its internal operation in terms of Processes and Entities that provide those functions.

[2] Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

t) Specifies performance requirements and recommends default values and applicable ranges for the operational parameters of a PrY.

u) Specifies how PrYs can be incorporated within the architecture of end stations, bridges, two-port Ethernet Data Encryption devices (EDEs), and bridged networks.

v) Describes the requirements for management of MAC Privacy protection, identifying the managed objects and defining the manged objects for PrYs.

w) Specifies a Management Information Base (MIB) module for PrY management.

x) Specifies a YANG configuration and operational state model for PrY management.

2. Normative references

Change the list of normative references in Clause 2 as follows:

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.[3,4]

IEEE Std 802.1Q™, IEEE Standard for Local and Metropolitan Area Networks: Bridges and Bridged Networks.

IEEE Std 802.1X™, IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control.

~~IEEE Std 802.1Xbx™-2014, IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control—Amendment 1: MAC Security Key Agreement Protocol (MKA) Extensions.~~

IEEE Std 802.1AB™, IEEE Standard for Local and Metropolitan Area Networks: Station and Media Access Control Connectivity and Discovery.

IEEE Std 802.1AC™, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Service Definition.

IEEE Std 802.3™, IEEE Standard for Ethernet.

IETF RFC 1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II, McCloghrie, K., and Rose, M. T., March 1991.[5]

IETF RFC 2578, STD 58, Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2), McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and Waldbusser, S., April 1999.

IETF RFC 2579, STD 58, Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2), McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and Waldbusser, S., April 1999.

IETF RFC 2580, STD 58, Conformance Statements for SMIv2, McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and Waldbusser, S., April 1999.

IETF RFC 2863, The Interfaces Group MIB using SMIv2, McCloghrie, K., and Kastenholz, F., June 2000.

IETF RFC 3418, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), Preshun, R., editor, December 2002.

IETF RFC 7317, A YANG Data Model for System Management, Bierman, A., Bjorklund, M., August 2014.

IETF RFC 7950, The YANG 1.1 Data Modeling Laguage, Bjorklund, M., August 2016.

[3] IEEE publications are available from The Institute of Electrical and Electronics Engineers (https://www.standards.ieee.org).
[4] The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.
[5] IETF RFCs are available from the Internet Engineering Task Force (https://www.ietf.org/rfc.html).

IETF RFC 8343, A YANG Data Model for Interface Management, Bjorklund, M., March 2018.

ISO/IEC 14882, Information Technology—Programming languages—C++.[6]

NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.[7]

[6] ISO/IEC documents are available from the International Organization of Standardization (https://www.iso.org/) and from the International Electrotechnical Commission (http://www.iec.ch). These documents are also available from the American National Standards Institute (https://www.ansi.org/).

[7] NIST Special Publications are available from the National Institute of Standards and Technology (https://csrc.nist.gov/).

3. Definitions

Change the following definitions in Clause 3 as shown:

access priority: The priority that a client of the MAC Service or MAC Internal Sublayer Service (ISS) associates ~~associated~~ with a given transmit request ~~made by a MAC Security Entity (SecY) at its Common Port~~.

NOTE—In this standard a MAC Security Entity (SecY) is a **shim** that can be a client of a service access point supported (at its lower interface) by a specific media access method, e.g., IEEE Std 802.3, and can provide (at its upper interface) a service access point that is used by its own client. From the point of view of the SecY, the priority associated with each transmit request received from its client is the **user priority** for that request and the priority associated with the corresponding transmit request that it makes of the underlying service is the **access priority**.

Customer Network Port (CNP): A port on the network component of an Ethernet Data Encryption device (EDE-CS, EDE-CC, or EDE-SS) that provides internal connectivity to the edge component of that EDE.

Provider Network Port (PNP): The black-side port of an Ethernet Data Encryption device (EDE-CS, EDE-CC, or EDE-SS).

user priority: The priority associated with a transmit request accepted by an entity that provides the MAC Service or MAC Internal Sublayer Service (ISS) ~~received by the Controlled Port of a MAC Security Entity (SecY)~~.

Insert the following terms and definitions in Clause 3 in alphabetical order:

express frame: A frame that a protocol entity identifies as a candidate for early transmission using **preemption** capabilities.

NOTE—Not all protocol entities that forward a given frame need identify that frame as an express frame or a preemptable frame. In this standard that identification uses the priority of the frame.

preemptable frame: A frame that a protocol entity identifies as a candidate for suspension by **preemption** capabilities, so as to allow the earlier transmission of an **express frame**.

preemption: The temporary suspension of the transmission (or encoding for transmission) of a **preemptable frame** to allow the earlier transmission of an **express frame**.

NOTE—The preemption capabilities specified by IEEE Std 802.3 can be used in conjunction with the MAC Security protocol (MACsec) to expedite the transmission of an **express frame** that becomes available for transmission after transmission of a **preemptable frame** has begun. The MAC Privacy protection protocol also supports preemption, allowing the encoding of an express frame prior to the encoding of the remaining fragment(s) of a preemptable frame.

Privacy Channel: A sequence of frames with the same MAC source and destination addresses each conveying a single MAC Privacy Protocol Data Unit (MPPDU), with the sequence conveying a sequence of entire or fragmented user data frames and padding.

NOTE—In this standard all unqualified references to “fragments” and “fragmentation” are to MPPDU encoding.

Privacy Frame: A frame that conveys a single MAC Privacy Protocol Data Unit (MPPDU) that includes a single, unfragmented, user data frame followed by zero or more octets of padding.

Private Port: The access point used to provide the privacy protected secure MAC Service to a client of a MAC Privacy protection Entity (PrY).

shim: A protocol entity that uses the same service as it provides.

NOTE—Shims specified or referenced in this standard secure the ISS, enhance privacy, or provided multiplexing over separate instances of the ISS.

traffic: A sequence of frames forwarded in a network.

4. Abbreviations and acronyms

Insert the following abbreviations and acronyms in Clause 4 in alphabetical order:

ATS	Asynchronous Traffic Shaping
C-TAG	C-VLAN tag
C-VID	Customer VLAN Identifier
C-VLAN	Customer Virtual Local Area Network
CNP	Customer Network Port
MPP	MAC Privacy protection[8]
MPPDU	MAC Privacy protection Protocol Data Unit
MPPCI	MAC Privacy protection Protocol Component Identifier
PBN	Provider Bridged Network
PCI	Personal Correlatable Information
PII	Personally Identifiable Information
PNP	Provider Network Port
PrY	MAC Privacy protection Entity
PSFP	Per-Stream Filtering and Policing
PVID	port VLAN Identifier
S-VLAN	Service Virtual Local Area Network
TPMR	Two-Port MAC Relay
VID	VLAN Identifier
VLAN	Virtual Local Area Network
YANG	Yet Another Next Generation[9]

[8] The acronym MPP is used in figures.
[9] YANG is best viewed as a name, not an acronym.

5. Conformance

Change the introductory text of Clause 5 as follows:

A claim of conformance to this standard for the implementation of MAC Security is a claim that the behavior of an implementation of a MAC Security Entity (SecY) meets the requirements of this standard (5.3, 5.4) as they apply to the operation of the MACsec protocol, management of its operation, and provision of service to the protocol clients of the SecY, as revealed through externally observable behavior of the system of which the SecY forms a part.

A claim of conformance for the implementation of MAC Security may be a claim of full conformance, or a claim of conformance with Cipher Suite variance, as specified in 5.4.

Conformance to this standard does not ensure that the system of which a ~~the~~ MAC Security implementation forms a part is secure, or that the operation of other protocols used to support MAC Security, such as key management and network management do not provide a way for an attacker to breach that security.

Conformance to this standard does not require any restriction as to the nature of the system of which a SecY forms part other than as constrained by the SecY's required and optional capabilities (5.3, 5.4). Clause 11 describes the use of SecYs within a number of different types of systems. These include, but are not limited to, systems specified in IEEE Std 802.1Q and those that make use of IEEE Std 802.1X. Successful interoperable use of MACsec in those systems also requires conformance to those standards. In addition Clause 15 of this standard makes use of components specified in IEEE Std 802.1Q to define further systems, Ethernet Data Encryption devices (EDEs), whose purpose is to secure the MAC Service within networks comprising bridging systems specified by IEEE Std 802.1Q in a way that is transparent to the operation of those bridging systems. Additional claims of conformance can be made to this standard in respect of EDEs (5.5–5.7).

A claim of conformance to this standard for the implementation of MAC Privacy protection is a claim that the behavior of an implementation of a MAC Privacy protection entity (PrY) meets the requirements of this standard (5.10, 5.11) as they apply to the operation of the MAC Privacy protection protocol, management of its operation, and provision of service to the protocol clients of the PrY, as revealed through externally observable behavior of the system of which the PrY forms a part.

Conformance to this standard does not require any restriction as to the nature of the system of which a PrY forms part other than as constrained by the PrY's required and optional capabilities. Clause 21 describes the deployment of PrYs in a number of different types of system and network scenarios.

This amendment does not make changes to 5.1 Requirements terminology, but includes it to provide clarity in requirements terminology at all stages of review, and in the use of the published amendment prior to its inclusion in a revision of the base standard:

5.1 Requirements terminology

For consistency with existing IEEE and IEEE 802.1 standards, requirements placed upon conformant implementations of this standard are expressed using the following terminology:

a) ***shall*** is used for mandatory requirements.

b) ***may*** is used to describe implementation or administrative choices ("may" means "is permitted to", and hence, "may" and "may not" mean precisely the same thing).

c) ***should*** is used for recommended choices (the behaviors described by "should" and "should not" are both permissible but not equally desirable choices).

The PICS proformas (~~see Annex A~~5.2) reflect~~s~~ the occurrences of the words *shall, may,* and *should* within the standard.

The standard avoids needless repetition and apparent duplication of its formal requirements by using ***is***, ***is not***, ***are***, and ***are not*** for definitions and the logical consequences of conformant behavior. Behavior that is permitted but is neither always required nor directly controlled by an implementor or administrator, or whose conformance requirement is detailed elsewhere, is described by ***can***. Behavior that never occurs in a conformant implementation or system of conformant implementations is described by ***cannot***.

Change 5.2 as follows:

5.2 Protocol Implementation Conformance Statements (PICS)

The supplier of a MAC Security Entity (SecY) implementation that is claimed to conform to this standard shall complete a copy of the PICS proforma provided in Annex A (normative) and shall provide the information necessary to identify both the supplier and the implementation.

The supplier of an EDE that is claimed to conform to this standard shall complete a copy of the PICS proforma provided in Annex D (normative) and shall provide the information necessary to identify both the supplier and the implementation. The supplier of an EDE implementation shall also complete or provide copies of the following PICS proforma(s) adhering to any restrictions required by conformance to this standard and marking any exceptions required by conformance to this standard:

a) For all types of EDE, the PICS proforma for each SecY implementation provided in Annex A of this standard.
b) For all types of EDE, the PICS proforma specified by IEEE Std 802.1X.
c) For an EDE-M: the IEEE Std 802.1Q PICS proforma as required for a VLAN-unaware MAC Bridge.
d) For an EDE-CS: the IEEE Std 802.1Q PICS proforma as required for a Provider Edge Bridge.
e) For an EDE-CC: the IEEE Std 802.1Q PICS proforma as required for each of the two C-VLAN components.
f) For an EDE-SS: the IEEE Std 802.1Q PICS proforma as required for each of the two S-VLAN components.

The supplier of a MAC Privacy protection Entity (PrY) implementation that is claimed to conform to this standard shall complete a copy of the PICS proforma provided in Annex H (normative) and shall provide the information necessary to identify both the supplier and the implementation.

5.5 EDE Conformance

Change the third paragraph and following text of 5.5 as follows:

An implementation of any type of EDE that is claimed to conform to this standard shall

a) Have two and only two externally accessible Bridge Ports, a red-side port and a black-side port.

   NOTE—A red-side port can also be referred to as the *customer* or *edge* port and the black-side port as a *provider* or *network* port. The use of either or both of the pair of terms, *customer/provider* and *edge/network* to refer to an EDE-M's ports is consistent with the relative roles played by ports in multicomponent bridges and EDEs.

b) Associate a Port Access Entity (PAE) that includes a MACsec Key Agreement Entity (KaY) capable of operating MKA with each SecY required by this standard for the particular type of EDE (15.2, 15.4, 15.5, 15.6, 15.7).

An implementation of any type of EDE that is claimed to conform to this standard may

c) Provide MAC Privacy protection for black-side port interface stack clients with a MAC Privacy protection entity (PrY) that uses the secure MAC Service provided by that port's SecY (21.5).

Change 5.8 as follows:

5.8 EDE-CC conformance

An implementation of an EDE-CC (15.6) that is claimed to conform to this standard shall

a) Comprise two C-VLAN bridge components, each as specified by IEEE Std 802.1Q ~~(5.5 of IEEE Std 802.1Q-2018)~~ —an edge component and a network component—internally connected as specified in 15.6.

b) Incorporate a SecY in each of the internal Provider Edge Port interface stacks (15.6).

c) Be capable of being configured to use the EDE-CC PAE group address as the destination MAC address of group addressed EAPOL PDUs for group addressed EAPOL PDUs (as specified in 15.6).

d) Filter and not forward all frames whose destination MAC address is either one of the addresses identified by IEEE Std 802.1Q as a C-VLAN component Reserved Address or the EDE-CC PAE group address.

An EDE-CC that is not capable of providing MAC Privacy protection shall

e) Transmit frames received from the red-side customer port and relayed to the black-side network port untagged if they were received untagged and C-tagged with the same C-VID if they were C-tagged on receipt (15.6).

An EDE-CC that is capable of providing MAC Privacy protection [5.5c)] shall

f) Be capable of being configured to transmit frames received from the red-side customer port and relayed to the black-side network port untagged if they were received untagged and C-tagged with the same C-VID if they were C-tagged on receipt (15.6).

An EDE-CC that is capable of providing MAC Privacy protection [5.5c)] may

g) Be capable, when relaying a frame received on the red-side customer port, of using the received C-VID to select the C-VID to be used when tagging the MACsec protected frame for transmission by the black-side network port (15.6).

NOTE—This optional capability was added by the IEEE Std 802.1AEdk MAC Privacy protection amendment to this standard to hide the use of two or more customer VLANs for traffic destined for the same peer PrY. Its use requires additional management and care in deployment. The frame transmitted through a peer red-side port, after protected transmission over a service provider network, is that originally received by the EDE.

Insert the following text (subclauses 5.10 and 5.11) after subclause 5.9:

5.10 MAC Privacy protection Entity requirements

An implementation of a MAC Privacy protection Entity (PrY) for which conformance to this standard is claimed shall

a) Support the Private and Controlled Ports as specified in Clause 20.

b) Support the MAC status and point-to-point parameters for the Private Port as specified in 20.4.

c) Transmit only MPPDUs whose encoding satisfies the constraints specified in 19.6.

d) Support requests to transmit user data frames through the Private Port as specified in 20.5 through 20.10.

e) Support per priority privacy selection (unprotected, as an individual Privacy Frame, or in a Privacy Channel) of user data frames transmitted by the client of the PrY's Private Port, as specified in 20.5.

f) Support unprotected user data frame transmission, with the ability to manage the access priority used to transmit that frame, as specified in 20.6.

g) Support individual Privacy Frame transmission, with the ability to manage the access priority used to transmit that frame, as specified in 20.7.

h) Support the configuration of two, one, or no transmit Privacy Channels as specified in 20.8.

i) Support the generation of Privacy Channel MPPDUs as specified in 20.9.

j) Be capable of being configured to use the default Privacy Channel MPPDU generation algorithm specified in 20.9.4.

k) Accept user data frames for Privacy Channel encapsulation as specified in 20.10.

l) Be capable of being configured to use the default Privacy Channel MPPDU encapsulation algorithm specified in 20.10.1.

m) Transmit all MPPDUs using the same destination MAC Address and the same source MAC Address as specified in 18.1.

n) Use the PAE Group Address that is being used by MKA as the destination MAC Address of transmitted MPPDUs and to recognize received MPPDUs when the PrY's Controlled Port is directly supported by a SecY (17.4, 18.1, 21.1.1).

o) Use the MAC Address used to generate MACsec SCI(s) as the source MAC Address of transmitted MPPDUs when the PrY's Controlled Port is directly supported by a SecY, as specified in 18.1.

p) Support administrative configuration of the destination MAC Address used to transmit MPPDUs if and when the PrY's Controlled Port is not directly supported by a SecY as an individual or group MAC address (18.1, 20.14).

q) Use a MAC Address associated with the PrY's interface stack as the source MAC Address of transmitted MPPDUs if and when the PrY is not directly supported by a SecY, as specified in 18.1.

r) Recognize received MPPDUs as specified in 20.11.

s) Validate and extract received MPPDU components as specified in 20.12 and 19.7.

t) Support simultaneous reassembly of Express and Preemptable frames from received Frame Fragments as specified in 20.13.

u) Deliver user data frames received in Encapsulated Frame components and reassembled from received Frame Fragments in the order specified in 20.13.

v) Be capable of being configured to use the default reassembly algorithm as specified in 20.13.1 to simultaneously reassemble Express Frame Fragments and Preemptable Frame Fragments transmitted by a single peer.

5.11 MAC Privacy protection Entity options

An implementation of a MAC Privacy protection Entity (PrY) for which conformance to this standard is claimed may

a) Be capable of being configured to use the default reassembly algorithm as specified in 20.13.1. for reception from a specified number (greater than one) of peer PrYs, each transmitting Express Frame Fragments and Preemptable Frame Fragments.

b) Use Private Port transmit request stream_handles to select Privacy Selection Table entries (20.5.2).

10. Principles of MAC Security Entity (SecY) operation

10.7 SecY management

Insert the following text after the second paragraph of 10.7:

Figure 10-5 includes data made available, through the LMI, to a SecY by other components (notably the KaY) within the same system, and data and operations that a SecY makes available to those components, in addition to operations and parameters accessible by one or more remote management protocols. Security considerations, and the utility of transient data, restrict management access to some parameters and operations. For particular managed object specifications based on this clause and figure, see Clause 13 (MAC Security Entity MIB) and Clause 23 (YANG Data Models).

Insert the following NOTE after NOTE 1 in 10.7, and renumber the NOTE that follows.

NOTE 2—Figure 10-5 was revised by IEEE Std 802.1AEdk-2022 to provide cross-references and to indicate the accessibility, read (r) or write (w) status of parameters that can be managed by remote management protocol, as specified for the SecY MIB module (Clause 13) or the YANG module (Clause 23). Whether access to particular management parameters is permitted is also subject to security considerations, as described for each of those modules.

Change the third paragraph of 10.7 as follows:

In Figure 10-5 the management information for each SecY is shown as indexed by controlledPortNumber 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 (12.9.2 of IEEE Std 802.1X-20~~1~~20) within a PAE System and includes the controlledPortNumber that identifies the Controlled Port of the associated SecY. The containment relationship also matches that specified in Clause 13 (MAC Security Entity MIB) and Clause 23 (YANG Data Models). In the MIB module a ~~, with a~~ SecY System corresponds~~ing~~ to a SecY MIB module instance, and each controlledPortNumber to the ifIndex (IETF RFC 2863) value used to identify a SecY within that module (13.3.2, 13.5). In the YANG module the management information for each PrY is indexed by an interfaceName.

Insert the following paragraph prior to the paragraph in 10.7 beginning "Conformance to this standard is strictly ..."

The representation of the time of the occurence of an event, such as the creation time for a receive SC (see 10.7.12, 10.7.14, 10.7.21, 10.7.23, 10.7.28), can depend on the management protocol used to convey that information. In SNMP sysUpTime, the time for which the SNMP agent has been running, is used. In YANG the wall-clock date-and-time is used.

Delete the last paragraph of 10.7, beginning "In some situations it can be desirable to substitute control using SNMP for the operation of key agreement protocols ..."

Replace Figure 10-5 with the following figure:

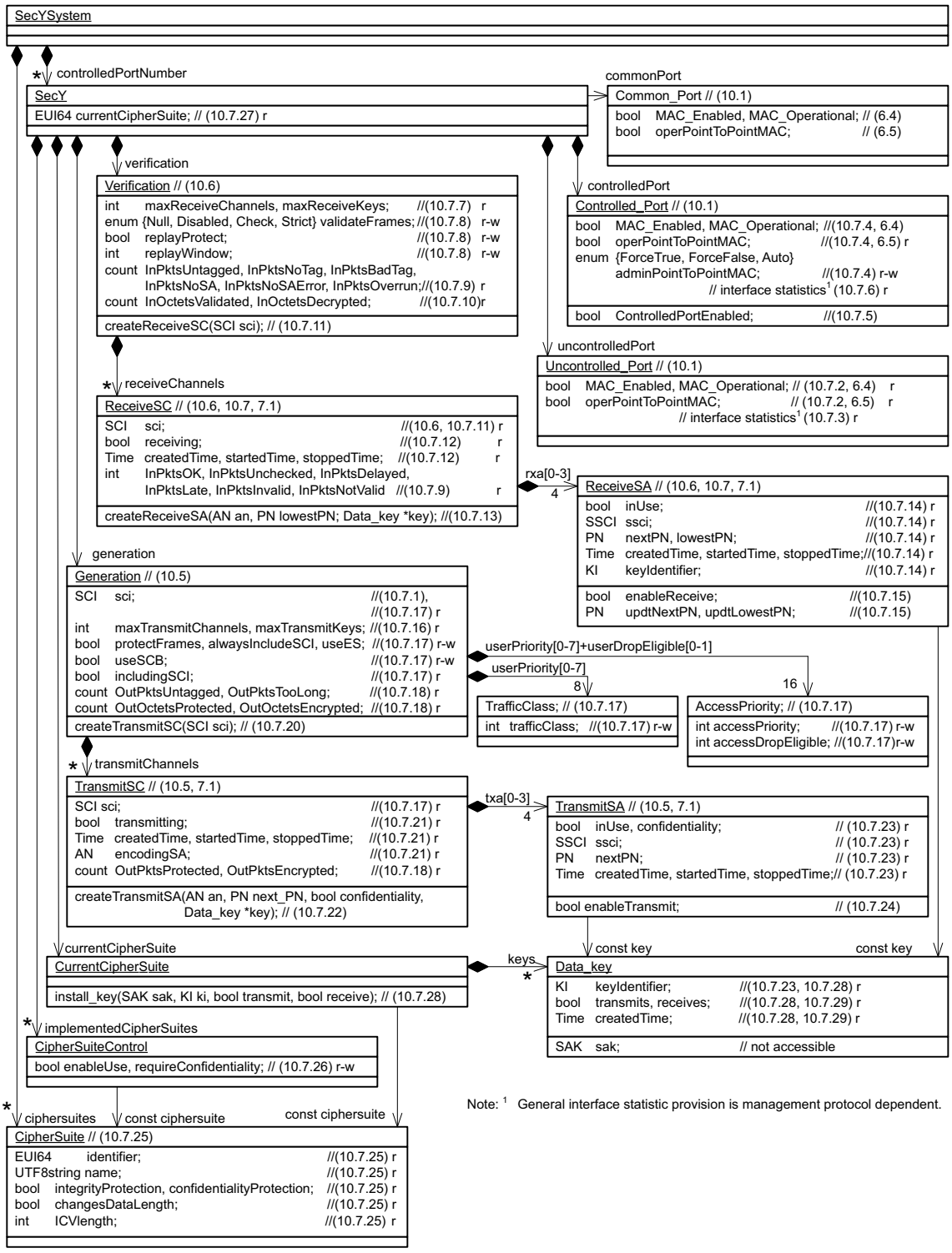


Figure 10-5—SecY managed objects

10.7.3 Uncontrolled Port statistics

Insert the following NOTE after the last paragraph of 10.7.3:

NOTE—The names of 32-bit counters specified by IETF RFC 2863 have been used to identify statistics, avoiding the need for a mapping from this standard's terminology to that specified for the Interfaces Group MIB. The use of 64-bit counters can be required to support that MIB, as specified in 3.1.6 of IETF RFC 2863, with the lower 32-bits reported when object is accessed using the names specified in a), b), e), and f) above. The corresponding 64-bit counter names are ifHCInOctets, ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCIOutOctets, ifHCOutUcastPkts, ifHCOutMulticastPkts, and ifHCOutBroadcastPkts. Corresponding 64-bit YANG data nodes are specified by IETF RFC 8343.

10.7.6 Controlled Port statistics

Insert the following NOTEs after the last paragraph of 10.7.6:

NOTE 1—The names of 32-bit counters specified by IETF RFC 2863 have been used to identify statistics, avoiding the need for a mapping from this standard's terminology to that specified for the Interfaces Group MIB. The use of 64-bit counters can be required to support that MIB, as specified in 3.1.6 of IETF RFC 2863, with the lower 32-bits reported when object is accessed using the names specified in a), b), e), and f) above. The corresponding 64-bit counter names are ifHCInOctets, ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCIOutOctets, ifHCOutUcastPkts, ifHCOutMulticastPkts, and ifHCOutBroadcastPkts. Corresponding 64-bit YANG data nodes are specified by IETF RFC 8343.

NOTE 2—Separate per SecY 64-bit counters for InPktsNoTAG, InPktsOverrun, InPktsBadTag, and InPktsNoSA, and separate per SC 64-bit counters for InPktsLate, and inPktsNotValid, are required to support the MIB and YANG modules specified in this standard (10.7.9, 13.6, Clause 23).

13. MAC Security Entity MIB

13.1 Introduction

Insert a NOTE after the text of 13.1 as follows:

NOTE—Annex G (informative) provides a detailed history and rationale for changes to SecY management and this MIB module. Normative changes to compliance and conformance statements are also described in 13.5.

13.3.2 Relationship to the Interfaces MIB

Replace Figure 13-1 with the following:

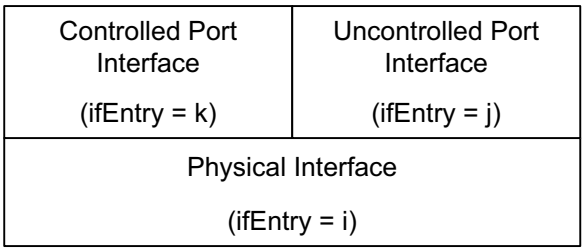


Figure 13-1—MACsec Interface Stack

13.6 MAC Security Entity (SecY) MIB definition[10, 11]

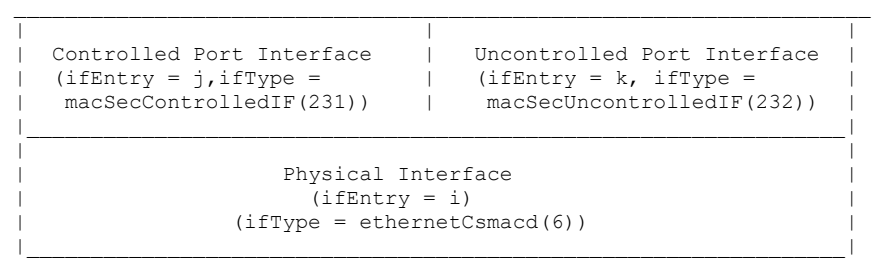
Replace the text of 13.6 with the following:

```
IEEE8021-SECY-MIB DEFINITIONS ::= BEGIN

-- ============================================================================
-- IEEEE802.1AE MAC Security Entity (SecY) MIB
-- ============================================================================

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    Unsigned32, Integer32, Counter32,Counter64
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowPointer, TimeStamp, TruthValue, RowStatus
        FROM SNMPv2-TC
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    InterfaceIndex, ifCounterDiscontinuityGroup
        FROM IF-MIB
    ;

ieee8021SecyMIB MODULE-IDENTITY
LAST-UPDATED "202206060000Z" -- June 6, 2022
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "  WG-URL: http://www.ieee802.org/1/
         WG-EMail: stds-802-1-l@ieee.org
           Contact: IEEE 802.1 Working Group Chair
            Postal: C/O IEEE 802.1 Working Group
                    IEEE Standards Association
                    445 Hoes Lane
                    Piscataway, NJ 08854
                    USA
            E-mail: stds-802-1-chairs@ieee.org"
    DESCRIPTION
    "The MAC security entity (SecY) MIB module. A SecY is a shim in an interface
     stack that uses the MAC Security (MACsec) protocol.

     Copyright (C) IEEE (2021).
     This version of this MIB module is part of IEEE Std 802.1AEdk-2022;
     see that standard for full legal notices.

     Unless otherwise indicated, the references in this MIB module are to
     IEEE Std 802.1AE-2018 as amended by IEEE Std 802.1AE-2018/Cor 1-2020 and
     IEEE Std 802.1AEdk-2022.

     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
```

[10] *Copyright release for MIBs*: Users of this standard may freely reproduce the MIB modules in this standard so that they can be used for their intended purpose.

[11] The MIB text in this clause includes clickable cross-references to MIB objects (highlighted). A plain text (UTF-8) version of this MIB is attached to the PDF version of this standard, and can be obtained by Web browser from the IEEE 802.1 Website at https://1.ieee802.org/mib-modules/.

```
     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. For each
     instance of a SecY two ifTable rows (one for each interface) run on
     top of an ifTable row representing the 'Common Port' interface,
     such as a row with ifType ='ethernetCsmacd(6)'.
```




```
        Example MACsec Interface Stack. i, j, k are ifIndexes each
        indicating a row in the ifTable.
    "
    REVISION "202206060000Z" -- June 6, 2022
    DESCRIPTION
    "Published as part of IEEE Std 802.1AEdk-2022.
     Cross references, contact information, and descriptions updated."

    REVISION    "201712071816Z"
    DESCRIPTION
    "Published as part of IEEE Std 802.1AE-2018.
     Updated CONTACT-INFO."

    REVISION    "201605102049Z"
    DESCRIPTION
    "Updated by the IEEE Std 802.1AEcg amendment. Object DESCRIPTIONs
     and references aligned with text of the standard (including prior
     amendments). IEEE 802.1AEcg Annex G details changes.
     The initial version of this ieee8021SecyMIB used the object
     name prefix 'secy' rather than 'ieee8021secy' (recommended by
     RFC 4181). The 'secy' prefix has been retained in this revision for
     for backwards compatibility and internal consistency."

    REVISION    "200601100000Z"
    DESCRIPTION "Initial version of this MIB in IEEE 802.1AE-2006"
    ::= { iso(1) std(0) iso8802(8802) ieee802dot1(1)
          ieee802dot1mibs(1) 3 }
-- ============================================================================
-- Textual Conventions

SecySCI ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
    "Textual convention for a Secure Channel Identifier (SCI).
    Each SC is identified by an SCI comprising a 48-bit MAC Address,
    allocated to the transmitting system and a 16-bit Port Identifier."
    REFERENCE   "7.1.2, Figure 7-7"
    SYNTAX OCTET STRING (SIZE (8))

SecyAN ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
    "Textual convention for an Association Number (AN).
    Each SC is comprised of a succession of SAs, each with a different SAK,
    identified by a Secure Association Identifier (SAI) comprising an SCI
    concatenated with a two-bit AN.  The SAI is unique for SAs used by SecYs
    participating in a given CA at any instant."
    REFERENCE   "7.1.3, Figure 7-7"
    SYNTAX Unsigned32 (0..3)

-- ============================================================================
-- subtrees in the SecY MIB
```

```
secyMIBNotifications    OBJECT IDENTIFIER ::= { ieee8021SecyMIB 0 }
secyMIBObjects          OBJECT IDENTIFIER ::= { ieee8021SecyMIB 1 }
    secyMgmtMIBObjects  OBJECT IDENTIFIER ::= {  secyMIBObjects 1 }
    secyStatsMIBObjects OBJECT IDENTIFIER ::= {  secyMIBObjects 2 }
secyMIBConformance      OBJECT IDENTIFIER ::= { ieee8021SecyMIB 2 }
    secyMIBCompliances  OBJECT IDENTIFIER ::= { secyMIBConformance 1 }
    secyMIBGroups       OBJECT IDENTIFIER ::= { secyMIBConformance 2 }
-- ============================================================================
--secyMgmtMIBObjects
--  secyIfTable
--  secyTSCTable   --
--  secyTSATable   --
--  secyIfTCTable  --
--  secyIfAPTable  --
--  secyRxSCTable
--  secyRxSATable
--  secyCipherSuiteTable
--  secyIfCipherTable
--The following are historic following approval of IEEE Std 802.1AEcg-2017,
--even if their STATUS remains 'current'. They do not include any objects
--that are part of a current conformance OBJECT-GROUP, and lack traffic
--class transmit SC and XPN support:
--  secyTxSCTable, secyTxSATable
-- ============================================================================
--  secyIfTable
secyIfTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyIfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "A table with an entry for each MAC Security protocol (MACsec) capable
    interface in the system, i.e. for each SecY. Configured value of writable
    objects in each table entry MUST be persistent and remain unchanged across
    re-initialization of the system's management entity."
    REFERENCE "10.7, Table 13-1"
    ::= { secyMgmtMIBObjects 1 }

--secyIfEntry
secyIfEntry    OBJECT-TYPE
    SYNTAX      SecyIfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "A table entry with service information for a particular SecY."
    INDEX   { secyIfInterfaceIndex }
    ::= { secyIfTable 1 }

--SecyIfEntry
SecyIfEntry ::= SEQUENCE {
    secyIfInterfaceIndex           InterfaceIndex,
    secyIfMaxPeerSCs               Unsigned32,
    secyIfRxMaxKeys                Unsigned32,
    secyIfTxMaxKeys                Unsigned32,
    secyIfProtectFramesEnable      TruthValue,
    secyIfValidateFrames           INTEGER,
    secyIfReplayProtectEnable      TruthValue,
    secyIfReplayProtectWindow      Unsigned32,
    secyIfCurrentCipherSuite       Unsigned32,
    secyIfAdminPt2PtMAC            INTEGER,
    secyIfOperPt2PtMAC             TruthValue,
    secyIfIncludeSCIEnable         TruthValue,
    secyIfUseESEnable              TruthValue,
    secyIfUseSCBEnable             TruthValue,
    secyIfSCI                      SecySCI,
    secyIfIncludingSCI             TruthValue,
    secyIfMaxTSCs                  Unsigned32
}
```

```
--secyIfInterfaceIndex
secyIfInterfaceIndex    OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The Interface Index for this SecY's Controlled Port."
    REFERENCE   "10.1"
    ::= { secyIfEntry 1 }

--secyIfMaxPeerSCs
secyIfMaxPeerSCs    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "security connections"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The maximum number of peer SCs for this SecY."
    REFERENCE   "10.7.7"
    ::= { secyIfEntry 2 }

--secyIfRxMaxKeys
secyIfRxMaxKeys    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "keys"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The maximum number of keys in simultaneous use for
                 reception for this SecY."
    REFERENCE   "10.7.7"
    ::= { secyIfEntry 3 }

--secyIfTxMaxKeys
secyIfTxMaxKeys    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "keys"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The maximum number of keys in simultaneous use for
                 transmission for this SecY."
    REFERENCE   "10.7.16"
    ::= { secyIfEntry 4 }

--secyIfProtectFramesEnable
secyIfProtectFramesEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Enables or disables protection of transmitted frames."
    REFERENCE   "10.7.17, Figure 10-3"
    DEFVAL { true }
    ::= { secyIfEntry 5 }

--secyIfValidateFrames
secyIfValidateFrames    OBJECT-TYPE
    SYNTAX      INTEGER {
                    disabled(1),
                    check(2),
                    strict(3),
                    null(4)    -- 802.1AEcg
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "Controls validation of received frames.
     disabled(1) : disable validation, remove SecTAGs and ICVs (if present.
     check(2)    : enable validation, do not discard invalid frames.
     strict(3)   : enable validation and  discard invalid frames.
     null(4)     : no processing, do not remove SecTAGs or ICVs."
    REFERENCE   "10.7.8, Figure 10-4"
    DEFVAL { strict }
    ::= { secyIfEntry 6 }
```

```
--secyIfReplayProtectEnable
secyIfReplayProtectEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Enables or disables replay protection."
    REFERENCE   "10.7.8, Figure 10-4"
    DEFVAL { true }
    ::= { secyIfEntry 7 }

--secyIfReplayProtectWindow
secyIfReplayProtectWindow    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "Packets"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The replay protection window size."
    REFERENCE   "10.7.8, Figure 10-4"
    DEFVAL { 0 }
    ::= { secyIfEntry 8 }

--secyIfCurrentCipherSuite
secyIfCurrentCipherSuite    OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The secyCipherSuiteTableindex for this SecY's in use Cipher
                 Suite. Should be read-only if secyIfCipherTable implemented."
    REFERENCE   "10.7.25"
    ::= { secyIfEntry 9 }

--secyIfAdminPt2PtMAC
secyIfAdminPt2PtMAC    OBJECT-TYPE
    SYNTAX      INTEGER {
                    forceTrue(1),
                    forceFalse(2),
                    auto(3)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
    "Controls the value of operPointToPointMAC (secyOperPt2PtMAC)
     reported to the user(s) of this SecY's Controlled Port:

     forceTrue(1)  : operPointToPointMAC is True, regardless of the
                     configuration and status of the SecY.
     forceFalse(2) : operPointToPointMAC is False, regardless of the
                     configuration and status of the SecY.
     auto(3)       : OperPointMAC is True if secyIfvalidateFrames is
                     strict and reception is from at most one peer SecY,
                     or if secyIfvalidateFrames is not strict and
                     operPointToPointMAC is True for the Common Port,
                     and is False otherwise."
    REFERENCE   "6.5, 10.7.4"
    DEFVAL { auto }
    ::= { secyIfEntry 10 }

--secyIfOperPt2PtMAC
secyIfOperPt2PtMAC    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "Reflects the current service connectivity to be assumed by the
     user(s) of the SecY's Controlled Port:

     true(1)  : connectivity is to at most one other system.
     false(2) : connectivity is to one or more other systems."
    REFERENCE   "6.5, 10.7.4"
    ::= { secyIfEntry 11 }
```

```
--secyIfIncludeSCIEnable
secyIfIncludeSCIEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Mandates inclusion of an explicit SCI in the SecTAG
                 when transmitting protected frames."
    REFERENCE   "10.5.3 alwaysIncludeSCI, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 12 }

--secyIfUseESEnable
secyIfUseESEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Enables use of the ES bit in the SecTAG when
                 transmitting protected frames."
    REFERENCE   "10.5.3 useES, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 13 }

--secyIfUseSCBEnable
secyIfUseSCBEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Enables use of the SCB bit in the SecTAG when
                 transmitting protected frames."
    REFERENCE   "10.5.3 useSCB, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 14 }

--secyIfSCI
secyIfSCI       OBJECT-TYPE
    SYNTAX      SecySCI
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The SCI for the SecY's default traffic class."
    REFERENCE   "7.1.2, 10.7.1"
    ::= { secyIfEntry 15 }

--secyIfIncludingSCI
secyIfIncludingSCI    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "True if an explicit SCI is included in the SecTAG when
                 transmitting protected frames."
    REFERENCE   "10.5.3 includingSCI, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 16 }

--secyIfMaxTSCs
secyIfMaxTSCs    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "security connections"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The maximum number of transmit SCs for this SecY."
    REFERENCE   "10.7.16"
    ::= { secyIfEntry 17 }
```

```
-- ============================================================================
--secyTSCTable
secyTSCTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A transmit SC management table for systems with SecY's capable
                of supporting  traffic class SCs."
    REFERENCE   "7.1.2, 10.7.17, 10.7.20"
    ::= { secyMgmtMIBObjects 10 }

--secyTSCEntry
secyTSCEntry    OBJECT-TYPE
    SYNTAX      SecyTSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry with transmit SC information for one of the system's
                 SecYs and one of its traffic classes."
    INDEX { secyIfInterfaceIndex, secyTSCI }
    ::= { secyTSCTable 1 }

--SecyTSCEntry
SecyTSCEntry ::= SEQUENCE {
    secyTSCI              SecySCI,
    secyTSCState          INTEGER,
    secyTSCEncodingSA     RowPointer,
    secyTSCCreatedTime    TimeStamp,
    secyTSCStartedTime    TimeStamp,
    secyTSCStoppedTime    TimeStamp
}
--secyTSCI
secyTSCI        OBJECT-TYPE
    SYNTAX      SecySCI
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The SCI for the transmit SC for this SecY and traffic class."
    REFERENCE   "7.1.2, 10.7.17, 10.7.20"
    ::= { secyTSCEntry 1 }

--secyTSCState
secyTSCState    OBJECT-TYPE
    SYNTAX      INTEGER { inUse(1), notInUse(2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The state of the transmit SC for this SecY and traffic class:
                 inUse(1)    : one or more SAs are in use.
                 notInUse(2) : no SAs are in use for this SC."
    REFERENCE   "10.7.20"
    ::= { secyTSCEntry 2 }

--secyTSCEncodingSA
secyTSCEncodingSA    OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The SA currently used to encode the SecTAG. The row pointer points to an
     entry in the secyTSATable. If no such information is available, the value
     shall be the OBJECT IDENTIFIER { 0 0 }."
    REFERENCE   "10.5.1, 10.7.21"
    ::= { secyTSCEntry 3 }

--secyTSCCreatedTime
secyTSCCreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC was created."
    REFERENCE   "10.7.21"
    ::= { secyTSCEntry 4 }
```

```
--secyTSCStartedTime
secyTSCStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SC last started transmitting."
    REFERENCE   "10.7.21"
    ::= { secyTSCEntry 5 }

--secyTSCStoppedTime
secyTSCStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SC last stopped transmitting."
    REFERENCE   "10.7.21"
    ::= { secyTSCEntry 6 }

-- ============================================================================
--secyTSATable
secyTSATable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A transmit SA management table for systems with SecY's capable
                 of supporting  traffic class SC's."
    REFERENCE   "10.7.22, Table 13-2"
    ::= { secyMgmtMIBObjects 11 }

--secyTSAEntry
secyTSAEntry    OBJECT-TYPE
    SYNTAX      SecyTSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry for a transmit SA."
    INDEX   { secyIfInterfaceIndex, secyTSCI, secyTSA }
    ::= { secyTSATable 1 }

--SecyTSAEntry
SecyTSAEntry ::= SEQUENCE {
    secyTSA                   SecyAN,
    secyTSAState              INTEGER,
    secyTSANextXPN            Counter64,
    secyTSAConfidentiality    TruthValue,
    secyTSAKeyIdentifier      SnmpAdminString,
    secyTSASSCI               Integer32,
    secyTSACreatedTime        TimeStamp,
    secyTSAStartedTime        TimeStamp,
    secyTSAStoppedTime        TimeStamp
}
--secyTSA
secyTSA     OBJECT-TYPE
    SYNTAX      SecyAN
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The association number (AN) for this transmit SA."
    REFERENCE   "10.7.22"
    ::= { secyTSAEntry 1 }

--secyTSAState
secyTSAState   OBJECT-TYPE
    SYNTAX      INTEGER { inUse(1), notInUse(2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The transmit SA current status: inUse(1) , notInUse(2)."
    REFERENCE   "10.7.23"
    ::= { secyTSAEntry 2 }
```

```
--secyTSANextXPN
secyTSANextXPN  OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The next packet number (PN) for this SA."
    REFERENCE   "10.5, 10.7.23"
    ::= { secyTSAEntry 3 }

--secyTSAConfidentiality
secyTSAConfidentiality    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "True if the SA provides confidentiality."
    REFERENCE   "10.7.23"
    ::= { secyTSAEntry 4 }

--secyTSAKeyIdentifier
secyTSAKeyIdentifier    OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (1..32))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The Key Identifier (KI) for the SAK for this SA."
    REFERENCE   "IEEE 802.1X, 10.7.23"
    ::= { secyTSAEntry 5 }

--secyTSASSCI
secyTSASSCI     OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The SSCI for this SA, 0 if not using an XPN Cipher Suite."
    REFERENCE   "IEEE 802.1X, 10.7.23"
    ::= { secyTSAEntry 6 }

--secyTSACreatedTime
secyTSACreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA was created."
    REFERENCE   "10.7.23"
    ::= { secyTSAEntry 7 }

--secyTSAStartedTime
secyTSAStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SA last started transmitting."
    REFERENCE   "10.7.23"
    ::= { secyTSAEntry 8 }

--secyTSAStoppedTime
secyTSAStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SA last stopped transmitting."
    REFERENCE   "10.7.23"
    ::= { secyTSAEntry 9 }
```

```
-- ============================================================================
--secyRxSCTable
secyRxSCTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyRxSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of receive SCs for the system."
    REFERENCE   "10.7.11, Table 13-2"
    ::= { secyMgmtMIBObjects 4 }

--secyRxSCEntry
secyRxSCEntry   OBJECT-TYPE
    SYNTAX      SecyRxSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table entry for a receive SC."
    INDEX    { secyIfInterfaceIndex, secyRxSCI }
    ::= { secyRxSCTable 1 }

--SecyRxSCEntry
SecyRxSCEntry ::= SEQUENCE {
    secyRxSCI              SecySCI,
    secyRxSCState          INTEGER,
    secyRxSCCurrentSA      RowPointer, -- deprecated
    secyRxSCCreatedTime    TimeStamp,
    secyRxSCStartedTime    TimeStamp,
    secyRxSCStoppedTime    TimeStamp
}

--secyRxSCI
secyRxSCI       OBJECT-TYPE
    SYNTAX      SecySCI
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The SCI for the receive SC."
    REFERENCE   "10.7.11"
    ::= { secyRxSCEntry 1 }

--secyRxSCState
secyRxSCState   OBJECT-TYPE
    SYNTAX      INTEGER {inUse(1), notInUse(2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The receive SCs current state:
                 inUse(1) : one or more SAs for this SC are in use.
                 notInUse(2) : no SAs for this SC is in use."
    REFERENCE   "10.7.12 receiving, 10.7.14 inUse, 10.7.15"
    ::= { secyRxSCEntry 2 }

--secyRxSCCurrentSA
secyRxSCCurrentSA    OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "The current receiving association number for the SC in use.
     The row pointer points to an entry in the secyRxSATable.  If no
     such information can be identified, the value of this object shall
     be the OBJECT IDENTIFIER { 0 0 }."
    REFERENCE   "10.7.15, 10.7.13"
    ::= { secyRxSCEntry 3 }

--secyRxSCCreatedTime
secyRxSCCreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this receiving SC was created."
    REFERENCE   "10.7.12"
    ::= { secyRxSCEntry 4 }
```

```
--secyRxSCStartedTime
secyRxSCStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SC last started receiving."
    REFERENCE   "10.7.12"
    ::= { secyRxSCEntry 5 }

--secyRxSCStoppedTime
secyRxSCStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SC last stopped receiving."
    REFERENCE   "10.7.12"
    ::= { secyRxSCEntry 6 }

-- ============================================================================
--secyRxSATable
secyRxSATable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyRxSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table with entries for the system's receive SAs."
    REFERENCE   "10.7.13"
    ::= { secyMgmtMIBObjects 5 }

--secyRxSAEntry
secyRxSAEntry   OBJECT-TYPE
    SYNTAX      SecyRxSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry for one of the SAs used by one of the system's
                 SecY's to receive protected frames."
    INDEX    { secyIfInterfaceIndex, secyRxSCI, secyRxSA }
    ::= { secyRxSATable 1 }

--SecyRxSAEntry
SecyRxSAEntry ::= SEQUENCE {
    secyRxSA               SecyAN,
    secyRxSAState          INTEGER,
    secyRxSANextPN         Unsigned32,      -- deprecated
    secyRxSASAKUnchanged   TruthValue,      -- deprecated
    secyRxSACreatedTime    TimeStamp,
    secyRxSAStartedTime    TimeStamp,
    secyRxSAStoppedTime    TimeStamp,
    secyRxSANextXPN        Counter64,
    secyRxSALowestXPN      Counter64,
    secyRxSAKeyIdentifier  SnmpAdminString,
    secyRxSASSCI           Integer32
}

--secyRxSA
secyRxSA        OBJECT-TYPE
    SYNTAX      SecyAN
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "This receive SA's association number (AN)."
    REFERENCE   "10.7.13"
    ::= { secyRxSAEntry 1 }

--secyRxSAState
secyRxSAState   OBJECT-TYPE
    SYNTAX      INTEGER { inUse(1), notInUse(2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This receive SA's current state."
    REFERENCE   "10.7.14"
    ::= { secyRxSAEntry 2 }
```

```
--secyRxSANextPN
secyRxSANextPN  OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      deprecated
    DESCRIPTION "Use secyRxSANextXPN for both 32-bit PN and 64-bit XPN values.
                 If implemented, this object contains the lower 32 bits."
    REFERENCE   "10.6.5, 10.7.14, Figure 10-4"
    ::= { secyRxSAEntry 3 }

--secyRxSASAKUnchanged
secyRxSASAKUnchanged   OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "An SAK reference, unchanged for the receiving SA's life."
    REFERENCE   "10.7.13"
    ::= { secyRxSAEntry 4 }

--secyRxSACreatedTime
secyRxSACreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this receiving SA was created."
    REFERENCE   "10.7.14"
    ::= { secyRxSAEntry 5 }

--secyRxSAStartedTime
secyRxSAStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SA last started receiving."
    REFERENCE   "10.7.14"
    ::= { secyRxSAEntry 6 }

--secyRxSAStoppedTime
secyRxSAStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SA last stopped receiving frames."
    REFERENCE   "10.7.14"
    ::= { secyRxSAEntry 7 }

--secyRxSANextXPN
secyRxSANextXPN        OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "One more than the highest PN conveyed in the SecTAG of
                 successfully validates frames received on this SA."
    REFERENCE   "10.6.5, 10.7.14, Figure 10-4"
    ::= { secyRxSAEntry 8 }

--secyRxSALowestXPN
secyRxSALowestXPN    OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The lowest acceptable packet number. A received frame
                 with a lower PN is  discarded if
                 secyIfReplayProtectEnable is enabled."
    REFERENCE   "10.6.2, 10.6.4, 10.6.5, 10.7.14,
        Figure 10-4"
    ::= { secyRxSAEntry 9 }
```

```
--secyRxSAKeyIdentifier
secyRxSAKeyIdentifier    OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (1..32))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The Key Identifier (KI) for the SAK for this SA."
    REFERENCE   "IEEE 802.1X, 10.7.14"
    ::= { secyRxSAEntry 10 }

--secyRxSASSCI
secyRxSASSCI    OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The SSCI for this SA, 0 if an XPN Cipher Suite is not in use."
    REFERENCE   "IEEE 802.1X, 10.7.14"
    ::= { secyRxSAEntry 11 }

-- ============================================================================
--secyCipherSuiteTable
secyCipherSuiteTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyCipherSuiteEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "A table of the system's Cipher Suite capabilities, which can differ
     by Cipher Suite implementation, so there can be more than one entry
     with the same secyCipherSuiteId. The secyIfCipherTable lists
     available entries by SecY, avoiding the need for remote network
     management to  write objects or create rows in this table. Any
     configured values shall be stored in persistent memory and remain
     unchanged across a re-initialization of the management system."
    REFERENCE   "10.7.25"
    ::= { secyMgmtMIBObjects 6 }

--secyCipherSuiteEntry
secyCipherSuiteEntry    OBJECT-TYPE
    SYNTAX      SecyCipherSuiteEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry for a Cipher Suite implementation."
    INDEX { secyCipherSuiteIndex }
    ::= { secyCipherSuiteTable 1 }

--SecyCipherSuiteEntry
SecyCipherSuiteEntry ::= SEQUENCE {
    secyCipherSuiteIndex              Unsigned32,
    secyCipherSuiteId                 OCTET STRING,
    secyCipherSuiteName               SnmpAdminString,
    secyCipherSuiteCapability         BITS,
    secyCipherSuiteProtection         BITS,
    secyCipherSuiteProtectionOffset   INTEGER,
    secyCipherSuiteDataLengthChange   TruthValue,
    secyCipherSuiteICVLength          Unsigned32,
    secyCipherSuiteRowStatus          RowStatus
}

--secyCipherSuiteIndex
secyCipherSuiteIndex    OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The CipherSuiteTable entry index."
    ::= { secyCipherSuiteEntry 1 }
```

```
--secyCipherSuiteId
secyCipherSuiteId    OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (8))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "A unique 64-bit (EUI-64) identifier for the Cipher
                 Suite."
    REFERENCE   "10.7.25, Table 14-1"
    ::= { secyCipherSuiteEntry 2 }

--secyCipherSuiteName
secyCipherSuiteName    OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (1..128))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "The Cipher Suite Name, 128 octets or fewer."
    REFERENCE   "10.7.25, Table 14-1"
    ::= { secyCipherSuiteEntry 3 }

--secyCipherSuiteCapability
secyCipherSuiteCapability    OBJECT-TYPE
    SYNTAX      BITS {
                    integrity(0),
                    confidentiality(1),
                    offsetConfidentiality(2)
                }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "Cipher Suite implementation capability information.

                 integrity(0)             : integrity protection.
                 confidentiality(1)       : confidentiality protection.
                 offsetConfidentiality(2) : offset confidentiality
                                            protection."
    REFERENCE   "10.7.24, 10.7.25"
    ::= { secyCipherSuiteEntry 4 }

--secyCipherSuiteProtection
secyCipherSuiteProtection    OBJECT-TYPE
    SYNTAX      BITS {
                    integrity(0),
                    confidentiality(1),
                    offsetConfidentiality(2)
                }
    MAX-ACCESS  read-create
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "The secyIfCipherSuite table supports per SecY configuration and should be
     used instead of this object.If the secyCipherSuiteCapability integrity
     bit is on, it can be turned on for this object. If the integrity and
     confidentiality bits of the secyCipherSuiteCapability are both on, the
     confidentiality bit of this object can be turned on provided that the
     integrity bit is also turned on, and the offsetConfidentiality bit can
     also be turned on if the secyCipherSuiteCapability has that bit on.

     integrity(0)       : enable (on) or disable integrity protection.
     confidentiality(1) : enable (on) or disable confidentiality protection.
     offsetConfidentiality(2) : enable (on) or disable offset confidentiality."
    REFERENCE   "10.7.25"
    DEFVAL { { integrity } }
    ::= { secyCipherSuiteEntry 5 }
```

```
--secyCipherSuiteProtectionOffset
secyCipherSuiteProtectionOffset    OBJECT-TYPE
    SYNTAX      Integer32 (0 | 30 | 50)
    UNITS       "bytes"
    MAX-ACCESS  read-create
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "The confidentiality protection offset options provided by the cipher
     suite. Can only be non-zero if the secyCipherSuiteProtection offset
     confidentiality bit is on, and then can only be 0 if the confidentiality
     bit is on."
    REFERENCE   "10.7.25, 10.7.26"
    DEFVAL { 0 }
    ::= { secyCipherSuiteEntry 6 }

--secyCipherSuiteDataLengthChange
secyCipherSuiteDataLengthChange    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "True if cipher suite changes the length of the data."
    REFERENCE   "10.7.25, Figure 9-1"
    ::= { secyCipherSuiteEntry 7 }

--secyCipherSuiteICVLength
secyCipherSuiteICVLength    OBJECT-TYPE
    SYNTAX      Unsigned32 (8..16)
    UNITS       "octets"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "The length of the integrity check value (ICV) field."
    REFERENCE   "10.7.25, Figure 9-1"
    ::= { secyCipherSuiteEntry 8 }

--secyCipherSuiteRowStatus
secyCipherSuiteRowStatus    OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
    "The secyIfCipherTable (if implemented) avoids the need for
     network manager creation of entries in the secyCipherSuiteTable,
     and RowStatus should always be valid(1), with any per SecY
     unavailability indicated by an absence of a corresponding
     secyIfCipherTable entry or one with secyCipherSuiteAvailable
     false (the latter can indicate temporary unavailability)."
    REFERENCE   "10.7.25"
    ::= { secyCipherSuiteEntry 9 }

-- ============================================================================
--secyIfCipherTable
secyIfCipherTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyIfCipherEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A table with an entry for the Cipher Suite capabilities
        implemented for each SecY in this system, providing per SecY
        control of Cipher Suite use.

        The configured value of writable objects in each table entry
        shall be stored in persistent memory and remain unchanged across
        a re-initialization of the system's management entity."
    REFERENCE   "10.7.26, Table 13-1"
    ::= { secyMgmtMIBObjects 7 }
```

```
--secyIfCipherEntry
secyIfCipherEntry    OBJECT-TYPE
    SYNTAX      SecyIfCipherEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table entry with Cipher Suite control for a SecY."
    INDEX   { secyIfInterfaceIndex, secyCipherSuiteIndex }
    ::= { secyIfCipherTable 1 }

--SecyIfCipherEntry
SecyIfCipherEntry ::= SEQUENCE {
    secyIfCipherImplemented         TruthValue,
    secyIfCipherEnableUse           TruthValue,
    secyIfCipherRqConfidentiality   TruthValue
}

--secyIfCipherImplemented
secyIfCipherImplemented    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "True if the Cipher Suite implementation can be used by
                 this SecY (if secIfCipherEnableUse is true)."
    REFERENCE   "10.7.26"
    DEFVAL { true }
    ::= { secyIfCipherEntry 1 }

--secyIfCipherEnableUse
secyIfCipherEnableUse    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Enables use of the Cipher Suite by this SecY."
    REFERENCE   "10.7.26"
    DEFVAL { true }
    ::= { secyIfCipherEntry 2 }

--secyIfCipherRqConfidentiality
secyIfCipherRqConfidentiality    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "True if confidentiality protection (without an offset)
                 is required if this Cipher Suite is used."
    REFERENCE   "10.7.26"
    DEFVAL { true }
    ::= { secyIfCipherEntry 3 }

-- ============================================================================
--secyIfTCTable
secyIfTCTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyIfTCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Traffic Class Table for each SecY in this system.
        The configured value of writable objects in each table entry
        MUST be stored in persistent memory and remain unchanged across
        a re-initialization of the system's management entity."
    REFERENCE   "10.5.1, 10.7.17, Table 13-1"
    ::= { secyMgmtMIBObjects 8 }

--secyIfTCEntry
secyIfTCEntry   OBJECT-TYPE
    SYNTAX      SecyIfTCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table entry providing Traffic Class selection for a given
                 SecY and user priority."
    INDEX   { secyIfInterfaceIndex, secyIfTCUserPriority }
    ::= { secyIfTCTable 1 }
```

```
--SecyIfTCEntry
SecyIfTCEntry ::= SEQUENCE {
    secyIfTCUserPriority           Integer32,
    secyIfTCTrafficClass           Integer32
}

--secyIfTCUserPriority
secyIfTCUserPriority    OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "One of the possible user priority values for a frame."
    REFERENCE   "10.7.17"
    ::= { secyIfTCEntry 1 }

--secyIfTCTrafficClass
secyIfTCTrafficClass    OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The Traffic Class for this SecY and user priority, as
        transmitted in the four most significant bits of the Port
        Identifier component of the SCI of protected frames."
    REFERENCE   "10.7.17"
    DEFVAL { 0 }
    ::= { secyIfTCEntry 2 }

-- ============================================================================
--secyIfAPTable
secyIfAPTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyIfAPEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Access Priority Table for each SecY in this system.
        The configured value of writable objects in each table entry
        MUST be stored in persistent memory and remain unchanged across
        a re-initialization of the system's management entity."
    REFERENCE   "10.5.1, 10.7.17, Table 13-1"
    ::= { secyMgmtMIBObjects 9 }

--secyIfAPEntry
secyIfAPEntry   OBJECT-TYPE
    SYNTAX      SecyIfAPEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table entry for a given SecY, selecting the access priority
                 and the drop_eligible parameter value used for a given user
                 priority and drop_eligible parameter value."
    INDEX   { secyIfInterfaceIndex, secyIfAPUserPCP }
    ::= { secyIfAPTable 1 }

--SecyIfAPEntry
SecyIfAPEntry ::= SEQUENCE {
    secyIfAPUserPCP             Integer32,
    secyIfAPAccessPCP           Integer32
}

--secyIfAPUserPCP
secyIfAPUserPCP OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The user priority (3 more significant bits) and drop_eligible
                 parameter (least significant bit) values."
    REFERENCE   "10.5, 10.7.17"
    ::= { secyIfAPEntry 1 }
```

```
--secyIfAPAccessPCP
secyIfAPAccessPCP    OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The access priority (3 more significant bits) and drop_eligible
                 parameter (least significant bit) values."
    REFERENCE   "10.5, 10.7.17"
    ::= { secyIfAPEntry 2 }
-- ============================================================================
--secyTxSCTable
secyTxSCTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current  -- ??
    DESCRIPTION "A transmit SC management table for systems not supporting
                 traffic class SC's, with an entry for each SecY."
    REFERENCE   "10.7.17, 10.7.20, Table 13-2"
    ::= { secyMgmtMIBObjects 2 }

--secyTxSCEntry
secyTxSCEntry    OBJECT-TYPE
    SYNTAX      SecyTxSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry with transmit SC information for a SecY."
    INDEX { secyIfInterfaceIndex }
    ::= { secyTxSCTable 1 }

--SecyTxSCEntry
SecyTxSCEntry ::= SEQUENCE {
    secyTxSCI              SecySCI,
    secyTxSCState          INTEGER,
    secyTxSCEncodingSA     RowPointer,
    secyTxSCEncipheringSA  RowPointer, -- deprecated
    secyTxSCCreatedTime    TimeStamp,
    secyTxSCStartedTime    TimeStamp,
    secyTxSCStoppedTime    TimeStamp
}
--secyTxSCI
secyTxSCI       OBJECT-TYPE
    SYNTAX      SecySCI
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The SCI for the SecY's transmit SC."
    REFERENCE   "7.1.2, 10.7.1"
    ::= { secyTxSCEntry 1 }

--secyTxSCState
secyTxSCState   OBJECT-TYPE
    SYNTAX      INTEGER { inUse(1), notInUse(2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The transmitting state of the SecY's transmit SC."
    REFERENCE   "10.7.21 transmitting, 10.7.23"
    ::= { secyTxSCEntry 2 }

--secyTxSCEncodingSA
secyTxSCEncodingSA    OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
    "The SA currently used to encode the SecTAG for frames awaiting transmission.
     The row pointer points to an entry in the secyTxSATable. If no such
     information is available, the value shall be the OBJECT IDENTIFIER { 0 0 }."
    REFERENCE   "10.5.1, 10.7.21"
    ::= { secyTxSCEntry 3 }
```

```
--secyTxSCEncipheringSA
secyTxSCEncipheringSA    OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
    "The SA currently used to encipher frames for transmission.
     The row pointer points to an entry in the secyTxSATable. If no such
     information is available, the value shall be the OBJECT IDENTIFIER { 0 0 }."
    REFERENCE   "10.5.4"
    ::= { secyTxSCEntry 4 }

--secyTxSCCreatedTime
secyTxSCCreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC was created."
    REFERENCE   "10.7.21"
    ::= { secyTxSCEntry 5 }

--secyTxSCStartedTime
secyTxSCStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SC last started transmitting."
    REFERENCE   "10.7.21"
    ::= { secyTxSCEntry 6 }

--secyTxSCStoppedTime
secyTxSCStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SC last stopped transmitting."
    REFERENCE   "10.7.21"
    ::= { secyTxSCEntry 7 }

-- ============================================================================
--secyTxSATable
secyTxSATable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A transmit SA management table for a system with no SecYs
                 capable of supporting traffic class transmit SCs."
    REFERENCE   "10.7.22, Table 13-2"
    ::= { secyMgmtMIBObjects 3 }

--secyTxSAEntry
secyTxSAEntry    OBJECT-TYPE
    SYNTAX      SecyTxSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry for a transmit SA."
    INDEX   { secyIfInterfaceIndex, secyTxSA }
    ::= { secyTxSATable 1 }

--SecyTxSAEntry
SecyTxSAEntry ::= SEQUENCE {
    secyTxSA                    SecyAN,
    secyTxSAState               INTEGER,
    secyTxSANextPN              Unsigned32,
    secyTxSAConfidentiality     TruthValue,
    secyTxSASAKUnchanged        TruthValue, -- deprecated
    secyTxSACreatedTime         TimeStamp,
    secyTxSAStartedTime         TimeStamp,
    secyTxSAStoppedTime         TimeStamp
}
```

```
--secyTxSA
secyTxSA        OBJECT-TYPE
    SYNTAX      SecyAN
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The association number (AN) for this transmit SA."
    REFERENCE   "10.7.22"
    ::= { secyTxSAEntry 1 }

--secyTxSAState
secyTxSAState   OBJECT-TYPE
    SYNTAX      INTEGER {inUse(1), notInUse(2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The transmit SAs current status: inUse(1), notInUse(2)."
    REFERENCE   "10.7.22"
    ::= { secyTxSAEntry 2 }

--secyTxSANextPN
secyTxSANextPN  OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The next packet number (PN) for this SA."
    REFERENCE   "10.5, 10.7.23"
    ::= { secyTxSAEntry 3 }

--secyTxSAConfidentiality
secyTxSAConfidentiality   OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "True if the SA provides confidentiality as well as
                 integrity for transmitted frames."
    REFERENCE   "10.7.23"
    ::= { secyTxSAEntry 4 }

--secyTxSASAKUnchanged
secyTxSASAKUnchanged      OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION "A reference to an SAK that is unchanged for the life
                 of the transmitting SA."
    REFERENCE   "10.7.22"
    ::= { secyTxSAEntry 5 }

--secyTxSACreatedTime
secyTxSACreatedTime       OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA was created."
    REFERENCE   "10.7.23"
    ::= { secyTxSAEntry 6 }

--secyTxSAStartedTime
secyTxSAStartedTime       OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SA last started transmitting."
    REFERENCE   "10.7.23"
    ::= { secyTxSAEntry 7 }
```

```
--secyTxSAStoppedTime
secyTxSAStoppedTime       OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this SA last stopped transmitting."
    REFERENCE   "10.7.23"
    ::= { secyTxSAEntry 8 }

-- ============================================================================
--secyStatsMIBObjects
-- secyTSCStatsTable
-- secyRxSCStatsTable
-- secyRxSAStatsTable
-- secyStatsTable
--The following are historic following approval of IEEE Std 802.1AEcg-2017,
--even if their STATUS remains 'current'. They do not include any objects
--that are part of a current conformance OBJECT-GROUP, and lack traffic
--class transmit SC and XPN support:
-- secyTxSCStatsTable, secyTxSAStatsTable
-- ============================================================================
--secyTSCStatsTable
secyTSCStatsTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of statistics for each SecY's transmit SCs."
    REFERENCE   "10.7.18, 10.7.19, Figure 10-3"
    ::= { secyStatsMIBObjects 12 }

--secyTSCStatsEntry
secyTSCStatsEntry    OBJECT-TYPE
    SYNTAX      SecyTSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A entry containing counts for a transmit SC, since SA counters
         are reset when the SA's AN  is reused these are a summation for
         all current and prior SAs belonging to the SC."
    AUGMENTS { secyTSCEntry }
    ::= { secyTSCStatsTable 1 }

--SecyTSCStatsEntry
SecyTSCStatsEntry ::= SEQUENCE {
    secyTSCStatsProtectedPkts      Counter64,
    secyTSCStatsEncryptedPkts      Counter64
}

--secyTSCStatsProtectedPkts
secyTSCStatsProtectedPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of integrity protected but not encrypted packets
         for this transmit SC."
    REFERENCE   "10.7.18, Figure 10-3"
    ::= { secyTSCStatsEntry 1 }
```

```
--secyTSCStatsEncryptedPkts
secyTSCStatsEncryptedPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of integrity protected and encrypted packets for
         this transmit SC."
    REFERENCE   "10.7.18, Figure 10-3"
    ::= { secyTSCStatsEntry 2 }

-- ============================================================================
--secyRxSAStatsTable
secyRxSAStatsTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyRxSAStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "A table that contains the statistics objects for each
         receiving SA in the MAC security entity."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsMIBObjects 3 }

--secyRxSAStatsEntry
secyRxSAStatsEntry    OBJECT-TYPE
    SYNTAX      SecyRxSAStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
       "An entry with statistics for a receive SA.  The AN that
        identifies an SA (for a given SC) and this corresponding entry
        can be reused. When creating the SA and before (re)using the
        entry, the SA counters are (re)set to 0. When the SA is stopped
        (secyRxSA notInuse) the counters stop incrementing.

        The secyRxSATable timestamps SA creation, start, and stop."
    AUGMENTS { secyRxSAEntry }
    ::= { secyRxSAStatsTable 1 }

--SecyRxSAStatsEntry
SecyRxSAStatsEntry ::= SEQUENCE {
    secyRxSAStatsUnusedSAPkts    Counter32, -- deprecated
    secyRxSAStatsNoUsingSAPkts   Counter32, -- deprecated
    secyRxSAStatsNotValidPkts    Counter32, -- deprecated
    secyRxSAStatsInvalidPkts     Counter32, -- deprecated
    secyRxSAStatsOKPkts          Counter32  -- deprecated
}

--secyRxSAStatsUnusedSAPkts
secyRxSAStatsUnusedSAPkts    OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "For this SA which is not currently in use, the number of
         received, unencrypted, packets with secyValidateFrames
         not in the strict mode."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 1 }
```

```
--secyRxSAStatsNoUsingSAPkts
secyRxSAStatsNoUsingSAPkts   OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "For this SA which is not currently in use, the number of
         received packets that have been discarded, and have
         either the packets encrypted or secyValidateFrames set to
         strict mode."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 4 }

--secyRxSAStatsNotValidPkts
secyRxSAStatsNotValidPkts     OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "For this SA, the number discarded packets with the
         condition that the packets are not valid and one of the
         following conditions are true: either secyValidateFrames in
         strict mode or the packets encrypted."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 13 }

--secyRxSAStatsInvalidPkts
secyRxSAStatsInvalidPkts     OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "For this SA, the number of packets with the condition
         that the packets are not valid and secyValidateFrames is in
         check mode."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 16 }

--secyRxSAStatsOKPkts
secyRxSAStatsOKPkts          OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "For this SA, the number of validated packets."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 25 }

-- ============================================================================
--secyRxSCStatsTable
secyRxSCStatsTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyRxSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of statistics for each receive SC for each of
                 the system's SecYs."
    REFERENCE   "10.7.9, 10.7.9, Figure 10-4"
    ::= { secyStatsMIBObjects 4 }
```

```
--secyRxSCStatsEntry
secyRxSCStatsEntry    OBJECT-TYPE
    SYNTAX      SecyRxSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "An entry containing counts for a receive SC. SA counters are
        reset when the SA's AN  is reused, so these SC counts are a
        summation for all current and prior SAs belonging to the SC."
    AUGMENTS { secyRxSCEntry }
    ::= { secyRxSCStatsTable 1 }

--SecyRxSCStatsEntry
SecyRxSCStatsEntry ::= SEQUENCE {
    secyRxSCStatsUnusedSAPkts         Counter64, -- deprecated
    secyRxSCStatsNoUsingSAPkts        Counter64, -- deprecated
    secyRxSCStatsLatePkts             Counter64,
    secyRxSCStatsNotValidPkts         Counter64,
    secyRxSCStatsInvalidPkts          Counter64,
    secyRxSCStatsDelayedPkts          Counter64,
    secyRxSCStatsUncheckedPkts        Counter64,
    secyRxSCStatsOKPkts               Counter64,
    secyRxSCStatsOctetsValidated      Counter64, -- deprecated
    secyRxSCStatsOctetsDecrypted      Counter64  -- deprecated
}

--secyRxSCStatsUnusedSAPkts
secyRxSCStatsUnusedSAPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "The sum of secyRxSAStatsUnusedSAPkts counts for all
                 current and prior SAs belonging to this SC."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 1 }

--secyRxSCStatsNoUsingSAPkts
secyRxSCStatsNoUsingSAPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "The sum of secyRxSAStatsNoUsingSAPkts counts for all
                 current and prior SAs belonging to this SC."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 2 }

--secyRxSCStatsLatePkts
secyRxSCStatsLatePkts        OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of packets discarded for this SC, with a received PN
                 lower than the lowest acceptable PN (secyRxSALowestXPN) while
                 secyIfReplayProtectEnable was true."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 3 }

--secyRxSCStatsNotValidPkts
secyRxSCStatsNotValidPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of packets discarded for this SC, because validation
                 failed and they were encrypted (unrecoverable) or
                 secyIfvalidateFrames was 'strict'."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 4 }
```

```
--secyRxSCStatsInvalidPkts
secyRxSCStatsInvalidPkts     OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of packets received for this SC, that failed validation
                 but were received unencrypted while secyIfvalidateFrames
                 was 'check'."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 5 }

--secyRxSCStatsDelayedPkts
secyRxSCStatsDelayedPkts     OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of packets received for this SC, with PN lower than
                 the lowest acceptable PN (secyRxSALowestXPN) while
                 secyIfReplayProtectEnable was false."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 6 }

--secyRxSCStatsUncheckedPkts
secyRxSCStatsUncheckedPkts   OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of packets received for this SC while
                 secyValidateFrames was 'disabled'."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 7 }

--secyRxSCStatsOKPkts
secyRxSCStatsOKPkts          OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of packets received for this SC that were
                 successfully validated and within the replay window."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 8 }

--secyRxSCStatsOctetsValidated
secyRxSCStatsOctetsValidated     OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "Count of plaintext octets recovered from packets that were
                 integrity protected but not encrypted."
    REFERENCE   "Deprecated, the secyIsStatsTable has per SecY counts
                 for cryptographic performance management."
    ::= { secyRxSCStatsEntry 9 }

--secyRxSCStatsOctetsDecrypted
secyRxSCStatsOctetsDecrypted    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "Count of plaintext octets recovered from packets
                 that were integrity protected and encrypted."
    REFERENCE   "Deprecated, the secyIsStatsTable has per SecY counts
                 for cryptographic performance management."
    ::= { secyRxSCStatsEntry 10 }
```

```
-- ============================================================================
--secyStatsTable
secyStatsTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of statistics for each of the system's SecYs."
    REFERENCE   "10.7.9, 10.7.18, Figure 10-3, 10.5"
    ::= { secyStatsMIBObjects 5 }

--secyStatsEntry
secyStatsEntry  OBJECT-TYPE
    SYNTAX      SecyStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing counts for a SecY."
    AUGMENTS { secyIfEntry }
    ::= { secyStatsTable 1 }

--SecyStatsEntry
SecyStatsEntry ::= SEQUENCE {
    secyStatsTxUntaggedPkts     Counter64,
    secyStatsTxTooLongPkts      Counter64,
    secyStatsRxUntaggedPkts     Counter64,
    secyStatsRxNoTagPkts        Counter64,
    secyStatsRxBadTagPkts       Counter64,
    secyStatsRxUnknownSCIPkts   Counter64, -- deprecated
    secyStatsRxNoSCIPkts        Counter64, -- deprecated
    secyStatsRxOverrunPkts      Counter64,
    secyStatsRxNoSAPkts         Counter64, -- 802.1AEcg
    secyStatsRxNoSAErrorPkts    Counter64, -- 802.1AEcg
    secyStatsTxOctetsProtected  Counter64, -- 802.1AEcg
    secyStatsTxOctetsEncrypted  Counter64, -- 802.1AEcg
    secyStatsRxOctetsValidated  Counter64, -- 802.1AEcg
    secyStatsRxOctetsDecrypted  Counter64  -- 802.1AEcg
}

--secyStatsTxUntaggedPkts
secyStatsTxUntaggedPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of packets transmitted without a SecTAG
                 because secyProtectFramesEnable is configured false."
    REFERENCE   "10.7.18, Figure 10-3"
    ::= { secyStatsEntry 1 }

--secyStatsTxTooLongPkts
secyStatsTxTooLongPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of transmit packets discarded because their
                 length is greater than the ifMtu of the Common Port."
    REFERENCE   "10.7.18, Figure 10-3"
    ::= { secyStatsEntry 2 }

--secyStatsRxUntaggedPkts
secyStatsRxUntaggedPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of packets without the MACsec tag (SecTAG)
                 received while secyValidateFrames was not 'strict'."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 3 }
```

```
--secyStatsRxNoTagPkts
secyStatsRxNoTagPkts       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of received packets  without a SecTAG
                 discarded because secyValidateFrames was 'strict'."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 4 }

--secyStatsRxBadTagPkts
secyStatsRxBadTagPkts      OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of received packets discarded with an
                 invalid SecTAG, zero value PN, or invalid ICV."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 5 }

--secyStatsRxUnknownSCIPkts
secyStatsRxUnknownSCIPkts  OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "The number of received packets with an unknown SCI."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 6 }

--secyStatsRxNoSCIPkts
secyStatsRxNoSCIPkts       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "The number of discarded packets with an unknown SCI."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 7 }

--secyStatsRxOverrunPkts
secyStatsRxOverrunPkts       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of packets discarded because they exceeded
                 cryptographic performance capabilities."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 8 }

--secyStatsRxNoSAPkts
secyStatsRxNoSAPkts       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of received packets with an unknown SCI
                 or for an unused SA."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 9 }
```

```
--secyStatsRxNoSAErrorPkts
secyStatsRxNoSAErrorPkts       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of packets discarded because the received
                 SCI is unknown or the SA is not in use."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 10 }

--secyStatsTxOctetsProtected
secyStatsTxOctetsProtected       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of plain text octets integrity protected
                 but not encrypted in transmitted frames."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 11 }

--secyStatsTxOctetsEncrypted
secyStatsTxOctetsEncrypted       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of plain text octets integrity protected
                 and encrypted in transmitted frames."
    REFERENCE   "10.7.9, Figure 10-4"
    ::= { secyStatsEntry 12 }

--secyStatsRxOctetsValidated
secyStatsRxOctetsValidated       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of plaintext octets recovered from packets
                 that were integrity protected but not encrypted."
    REFERENCE   "10.6.3, Figure 10-3"
    ::= { secyStatsEntry 13 }

--secyStatsRxOctetsDecrypted
secyStatsRxOctetsDecrypted       OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of plaintext octets recovered from packets
                 that were integrity protected and encrypted."
    REFERENCE   "10.6.3, Figure 10-3"
    ::= { secyStatsEntry 14 }
-- ================================================================================
--secyTxSCStatsTable
secyTxSCStatsTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A transmit SC statistics table for systems without traffic
                 class SC support, with an entry for each SecY."
    REFERENCE   "10.7.18, 10.7.19, Figure 10-3"
    ::= { secyStatsMIBObjects 2 }
```

```
--secyTxSCStatsEntry
secyTxSCStatsEntry    OBJECT-TYPE
    SYNTAX      SecyTxSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A transmit SC statistics table entry (for systems without
                 traffic class SC support) with cumulative counts for a given
                 SecY's current and prior SAs."
    AUGMENTS { secyTxSCEntry }
    ::= { secyTxSCStatsTable 1 }

--SecyTxSCStatsEntry
SecyTxSCStatsEntry ::= SEQUENCE {
    secyTxSCStatsProtectedPkts      Counter64,
    secyTxSCStatsEncryptedPkts      Counter64,
    secyTxSCStatsOctetsProtected    Counter64, -- deprecated
    secyTxSCStatsOctetsEncrypted    Counter64  -- deprecated
}
--secyTxSCStatsProtectedPkts
secyTxSCStatsProtectedPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of integrity protected but unencrypted packets for this
                 transmit SC."
    REFERENCE   "10.7.18, Figure 10-3"
    ::= { secyTxSCStatsEntry 1 }

--secyTxSCStatsEncryptedPkts
secyTxSCStatsEncryptedPkts    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Count of integrity protected and encrypted packets for this
                 transmit SC."
    REFERENCE   "10.7.18, Figure 10-3"
    ::= { secyTxSCStatsEntry 4 }

--secyTxSCStatsOctetsProtected
secyTxSCStatsOctetsProtected    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "Count of integrity protected plain text octets that are
integrity
                 protected but not encrypted for this transmit SC."
    REFERENCE   "10.7.19, Figure 10-3"
    ::= { secyTxSCStatsEntry 10 }

--secyTxSCStatsOctetsEncrypted
secyTxSCStatsOctetsEncrypted    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
        "The number of plain text octets that are integrity protected
         and encrypted on the transmit SC."
    REFERENCE   "10.7.19, Figure 10-3"
    ::= { secyTxSCStatsEntry 11 }
```

```
-- ============================================================================
--secyTxSAStatsTable
secyTxSAStatsTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSAStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION "A table of statistics for each transmit SA for each of
                 the system's SecYs."
    REFERENCE   "10.7.18, Figure 10-4"
    ::= { secyStatsMIBObjects 1 }

--secyTxSAStatsEntry
secyTxSAStatsEntry    OBJECT-TYPE
    SYNTAX      SecyTxSAStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
       "An entry with statistics for a transmit SA.  The AN that
        identifies an SA (for a given SC) and this corresponding entry
        can be reused. When creating the SA and before (re)using the
        entry, the SA counters are (re)set to 0. When the SA is stopped
        (secyTxSA notInuse) the counters stop incrementing.

        The secyTxSATable timestamps SA creation, start, and stop."
    AUGMENTS { secyTxSAEntry }
    ::= { secyTxSAStatsTable 1 }

--SecyTxSAStatsEntry
SecyTxSAStatsEntry ::= SEQUENCE {
    secyTxSAStatsProtectedPkts    Counter32,
    secyTxSAStatsEncryptedPkts    Counter32
}

--secyTxSAStatsProtectedPkts
secyTxSAStatsProtectedPkts    OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "Count of integrity protected but unencrypted packets for this
                 transmit SA. Zero if secyTxSAConfidentiality is True, and one
                 less than secyTxSANextPN otherwise."
    REFERENCE   "10.7.18, Figure 10-4"
    ::= { secyTxSAStatsEntry 1 }

--secyTxSAStatsEncryptedPkts
secyTxSAStatsEncryptedPkts    OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "Count of integrity protected and encrypted packets for this
                 transmit SA. Zero if secyTxSAConfidentiality is False, and
                 one less than secyTxSANextPN otherwise."
    REFERENCE   "10.7.18, Figure 10-4"
    ::= { secyTxSAStatsEntry 2 }
```

```
-- ============================================================================
--secyMIBCompliances
--secyMIBTcCompliance
secyMIBTcCompliance MODULE-COMPLIANCE
    STATUS current -- 802.1AEcg
    DESCRIPTION
    "The compliance statement for an IEEE8021-SECY-MIB supporting
     traffic class transmit SCs, added by IEEE 802.1AEcg."
    MODULE IF-MIB
        MANDATORY-GROUPS {
            ifCounterDiscontinuityGroup
            }
    MODULE  -- this module
        MANDATORY-GROUPS {
            secyIfGroup,
            secyIfCipherGroup,
            secyIfTCGroup,
            secyIfAPGroup,
            secyTSCGroup,
            secyTSAGroup,
            secyRSCGroup,
            secyRSAGroup,
            secyCipherInfoGroup,
            secyCipherStatsGroup,
            secyTSCStatsGroup,
            secyRSCStatsGroup,
            secyIfStatsGroup
        }
        OBJECT secyIfCurrentCipherSuite
            MIN-ACCESS    read-only
            DESCRIPTION "should be read-only, use the secyIfCipherTable
                         to control ciper suite use."
        OBJECT    secyCipherSuiteId
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteName
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, should be read-only."
        OBJECT    secyCipherSuiteCapability
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, should be read-only."
        OBJECT    secyCipherSuiteDataLengthChange
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, should be read-only."
        OBJECT    secyCipherSuiteICVLength
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, should be read-only."
    ::= { secyMIBCompliances 2 }
```

```
-- ============================================================================
--secyMIBCompliance
secyMIBCompliance MODULE-COMPLIANCE
    STATUS deprecated -- 802.1AEcg
    DESCRIPTION
    "The compliance statement for the IEEE8021-SECY-MIB as specified in
     IEEE Std 802.1AE-2006."
    MODULE  -- this module
        MANDATORY-GROUPS {
            secyIfCtrlGroup,
            secyTxSCGroup,
            secyTxSAGroup,
            secyRxSCGroup,
            secyRxSAGroup,
            secyCipherSuiteGroup,
            secyTxSAStatsGroup,
            secyTxSCStatsGroup,
            secyRxSAStatsGroup,
            secyRxSCStatsGroup,
            secyStatsGroup
        }
        OBJECT secyIfCurrentCipherSuite
            MIN-ACCESS    read-only
            DESCRIPTION "write access not required, may be read-only."
        OBJECT    secyCipherSuiteId
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteName
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteCapability
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteProtection
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteProtectionOffset
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteDataLengthChange
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteICVLength
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
        OBJECT    secyCipherSuiteRowStatus
            MIN-ACCESS    read-only
            DESCRIPTION "read-create not required, may be read-only."
    ::= { secyMIBCompliances 1 }
```

```
-- ============================================================================
--secyMIBGroups
--secyIfGroup
secyIfGroup    OBJECT-GROUP
    OBJECTS {
        secyIfMaxPeerSCs,
        secyIfRxMaxKeys,
        secyIfTxMaxKeys,
        secyIfProtectFramesEnable,
        secyIfValidateFrames,
        secyIfReplayProtectEnable,
        secyIfReplayProtectWindow,
        secyIfCurrentCipherSuite,
        secyIfAdminPt2PtMAC,
        secyIfOperPt2PtMAC,
        secyIfIncludeSCIEnable,
        secyIfUseESEnable,
        secyIfUseSCBEnable,
        secyIfSCI,                  -- 802.1AEcg
        secyIfIncludingSCI,         -- 802.1AEcg
        secyIfMaxTSCs               -- 802.1AEcg
    }
    STATUS      current --- Updated secyIfCtrlGroup
    DESCRIPTION "SecY service management (secyIfTable objects) for
                 systems supporting traffic class SCs."
    ::= { secyMIBGroups 12 }

--secyIfCtrlGroup
secyIfCtrlGroup    OBJECT-GROUP
    OBJECTS {
        secyIfMaxPeerSCs,
        secyIfRxMaxKeys,
        secyIfTxMaxKeys,
        secyIfProtectFramesEnable,
        secyIfValidateFrames,
        secyIfReplayProtectEnable,
        secyIfReplayProtectWindow,
        secyIfCurrentCipherSuite,
        secyIfAdminPt2PtMAC,
        secyIfOperPt2PtMAC,
        secyIfIncludeSCIEnable,
        secyIfUseESEnable,
        secyIfUseSCBEnable
    }
    STATUS      deprecated
    DESCRIPTION "SecY service management (secyIfTable) objects."
    ::= { secyMIBGroups 1 }

--secyIfTCGroup
secyIfTCGroup    OBJECT-GROUP
    OBJECTS {
        secyIfTCTrafficClass
    }
    STATUS      current --- 802.1AEcg
    DESCRIPTION "Traffic class control (secyIfTCTable)."
    ::= { secyMIBGroups 14 }

--secyIfAPGroup
secyIfAPGroup    OBJECT-GROUP
    OBJECTS {
        secyIfAPAccessPCP
    }
    STATUS      current
    DESCRIPTION "Access Priority Code Point control (secyIfAPTable)."
    ::= { secyMIBGroups 15 }
```

```
--secyTSCGroup
secyTSCGroup    OBJECT-GROUP
    OBJECTS {
        secyTSCState,
        secyTSCEncodingSA,
        secyTSCCreatedTime,
        secyTSCStartedTime,
        secyTSCStoppedTime
    }
    STATUS      current --- Updated secyTxSCGroup
    DESCRIPTION "Transmit SC management (secyTSCTable objects) for
                 systems supporting traffic class SCs."
    ::= { secyMIBGroups 16 }

--secyTxSCGroup
secyTxSCGroup    OBJECT-GROUP
    OBJECTS {
        secyTxSCI,
        secyTxSCState,
        secyTxSCEncodingSA,
        secyTxSCEncipheringSA,
        secyTxSCCreatedTime,
        secyTxSCStartedTime,
        secyTxSCStoppedTime
    }
    STATUS      deprecated
    DESCRIPTION "Transmit SC management objects (for systems without
                 traffic class SC capabilities)."
    ::= { secyMIBGroups 2 }

--secyTSAGroup
secyTSAGroup    OBJECT-GROUP
    OBJECTS {
        secyTSAState,
        secyTSANextXPN,
        secyTSAConfidentiality,
        secyTSAKeyIdentifier,
        secyTSASSCI,
        secyTSACreatedTime,
        secyTSAStartedTime,
        secyTSAStoppedTime
    }
    STATUS      current --- 802.1AEcg, updates secyTxSAGroup
    DESCRIPTION "Transmit SA management (secyTSATable objects) for
                 systems supporting traffic class SCs."
    ::= { secyMIBGroups 17 }

--secyTxSAGroup
secyTxSAGroup    OBJECT-GROUP
    OBJECTS {
        secyTxSAState,
        secyTxSANextPN,
        secyTxSAConfidentiality,
        secyTxSASAKUnchanged,
        secyTxSACreatedTime,
        secyTxSAStartedTime,
        secyTxSAStoppedTime
    }
    STATUS      deprecated
    DESCRIPTION "Transmit SA management objects (for systems without
                 traffic class SC capabilities)."
    ::= { secyMIBGroups 3 }
```

```
--secyRSCGroup
secyRSCGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSCState,
        secyRxSCCreatedTime,
        secyRxSCStartedTime,
        secyRxSCStoppedTime
    }
    STATUS      current --- 802.1AEcg, updates secyRxSCGroup
    DESCRIPTION "Receive SC management (secyRxSCTable objects)."
    ::= { secyMIBGroups 18 }

--secyRxSCGroup
secyRxSCGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSCState,
        secyRxSCCurrentSA,
        secyRxSCCreatedTime,
        secyRxSCStartedTime,
        secyRxSCStoppedTime
    }
    STATUS      deprecated
    DESCRIPTION "Receive SC management objects."
    ::= { secyMIBGroups 4 }

--secyRSAGroup
secyRSAGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSAState,
        secyRxSANextXPN,
        secyRxSALowestXPN,
        secyRxSAKeyIdentifier,
        secyRxSASSCI,
        secyRxSACreatedTime,
        secyRxSAStartedTime,
        secyRxSAStoppedTime
    }
    STATUS      current --- Updated secyRxSAGroup
    DESCRIPTION "Receive SA (secyRxSATable objects)."
    ::= { secyMIBGroups 19 }

--secyRxSAGroup
secyRxSAGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSAState,
        secyRxSANextPN,
        secyRxSASAKUnchanged,
        secyRxSACreatedTime,
        secyRxSAStartedTime,
        secyRxSAStoppedTime
    }
    STATUS      deprecated
    DESCRIPTION "Receive SA management objects."
    ::= { secyMIBGroups 5 }

-- Cipher information, use, and statistics MIB Groups

--secyCipherInfoGroup
secyCipherInfoGroup    OBJECT-GROUP
    OBJECTS {
        secyCipherSuiteId,
        secyCipherSuiteName,
        secyCipherSuiteCapability,
        secyCipherSuiteDataLengthChange,
        secyCipherSuiteICVLength
    }
    STATUS      current --- Updated secyCipherSuiteGroup
    DESCRIPTION "Cipher Suite implementation information
                 (secyCipherSuiteTable objects)."
    ::= { secyMIBGroups 21 }
```

```
--secyCipherSuiteGroup
secyCipherSuiteGroup    OBJECT-GROUP
    OBJECTS {
        secyCipherSuiteId,
        secyCipherSuiteName,
        secyCipherSuiteCapability,
        secyCipherSuiteProtection,
        secyCipherSuiteProtectionOffset,
        secyCipherSuiteDataLengthChange,
        secyCipherSuiteICVLength,
        secyCipherSuiteRowStatus
    }
    STATUS      deprecated
    DESCRIPTION "Cipher Suite information objects."
    ::= { secyMIBGroups 6 }

--secyIfCipherGroup
secyIfCipherGroup    OBJECT-GROUP
    OBJECTS {
        secyIfCipherImplemented,
        secyIfCipherEnableUse,
        secyIfCipherRqConfidentiality
    }
    STATUS      current --- 802.1AEcg
    DESCRIPTION "Cipher Suite use control (secyIfCipherTable  objects)."
    ::= { secyMIBGroups 13 }

--secyCipherStatsGroup
secyCipherStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyStatsTxOctetsProtected,
        secyStatsTxOctetsEncrypted,
        secyStatsRxOctetsValidated,
        secyStatsRxOctetsDecrypted
    }
    STATUS      current
    DESCRIPTION
        "Cipher Suite performance statistics (from secyStatsTable)."
    ::= { secyMIBGroups 24 }

--secyTxSAStatsGroup
secyTxSAStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyTxSAStatsProtectedPkts,
        secyTxSAStatsEncryptedPkts
    }
    STATUS      deprecated
    DESCRIPTION "Transmit SA statistics objects."
    ::= { secyMIBGroups 7 }

--secyRxSAStatsGroup
secyRxSAStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSAStatsUnusedSAPkts,
        secyRxSAStatsNoUsingSAPkts,
        secyRxSAStatsNotValidPkts,
        secyRxSAStatsInvalidPkts,
        secyRxSAStatsOKPkts
    }
    STATUS      deprecated
    DESCRIPTION "Receive SA statistics objects."
    ::= { secyMIBGroups 8 }
```

```
--secyTSCStatsGroup
secyTSCStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyTSCStatsProtectedPkts,
        secyTSCStatsEncryptedPkts
    }
    STATUS      current --- Updated secyTxSCStatsGroup
    DESCRIPTION "Transmit SC statistics (secyTSCStatsTable objects)."
    ::= { secyMIBGroups 22 }

--secyTxSCStatsGroup
secyTxSCStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyTxSCStatsProtectedPkts,
        secyTxSCStatsEncryptedPkts,
        secyTxSCStatsOctetsProtected,
        secyTxSCStatsOctetsEncrypted
    }
    STATUS      deprecated
    DESCRIPTION "Transmit SC statistics objects."
    ::= { secyMIBGroups 9 }

--secyRSCStatsGroup
secyRSCStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSCStatsLatePkts,
        secyRxSCStatsNotValidPkts,
        secyRxSCStatsInvalidPkts,
        secyRxSCStatsDelayedPkts,
        secyRxSCStatsUncheckedPkts,
        secyRxSCStatsOKPkts
    }
    STATUS      current --- Updated secyRxSCStatsGroup
    DESCRIPTION "Receive SC statistics (secyRxSCStatsTable objects)."
    ::= { secyMIBGroups 23 }

--secyRxSCStatsGroup
secyRxSCStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSCStatsUnusedSAPkts,
        secyRxSCStatsNoUsingSAPkts,
        secyRxSCStatsLatePkts,
        secyRxSCStatsNotValidPkts,
        secyRxSCStatsInvalidPkts,
        secyRxSCStatsDelayedPkts,
        secyRxSCStatsUncheckedPkts,
        secyRxSCStatsOKPkts,
        secyRxSCStatsOctetsValidated,
        secyRxSCStatsOctetsDecrypted
    }
    STATUS      deprecated
    DESCRIPTION
        "Receive SC statistics objects."
    ::= { secyMIBGroups 10 }

--secyIfStatsGroup
secyIfStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyStatsTxUntaggedPkts,
        secyStatsTxTooLongPkts,
        secyStatsRxUntaggedPkts,
        secyStatsRxNoTagPkts,
        secyStatsRxBadTagPkts,
        secyStatsRxNoSAPkts,
        secyStatsRxNoSAErrorPkts,
        secyStatsRxOverrunPkts
    }
    STATUS      current --- 802.1AEcg, updates secyRxSCStatsGroup
    DESCRIPTION
        "SecY statistics (secyStatsTable objects)."
    ::= { secyMIBGroups 20 }
```

```
--secyStatsGroup
secyStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyStatsTxUntaggedPkts,
        secyStatsTxTooLongPkts,
        secyStatsRxUntaggedPkts,
        secyStatsRxNoTagPkts,
        secyStatsRxBadTagPkts,
        secyStatsRxUnknownSCIPkts,
        secyStatsRxNoSCIPkts,
        secyStatsRxOverrunPkts
    }
    STATUS      deprecated
    DESCRIPTION
        "SecY statistics objects."
    ::= { secyMIBGroups 11 }

END
```

15. Ethernet Data Encryption devices

15.6 Securing PBN connectivity with an EDE-CC

Change the sixth paragraph of 15.6 as follows:

The configuration of an EDE-CC that does not provide MAC Privacy protection is constrained to restrict egress for each Provider Edge Port to a single C-VID and to restrict the PVID for the internally connected Customer Network Port to the same value, with the consequence that the outer C-VID ~~will~~ always matches the inner C-VID. The PVID for the Customer Edge Port is constrained to be the same as that for the Provider Network Port and the Static VLAN Registration Entry (8.8.2 of IEEE Std 802.1Q-2018) for that and other VIDs are constrained so that frames for that VID are transmitted untagged on both ports, with the consequence that frames received untagged on either port are forwarded (if at all) untagged on the other. These restrictions simplify EDE management, supporting the desired separation of concerns (11.1) and maintaining the scope of address learning within each C-VLAN. If the desired secured connectivity between the EDE-CC and its potential (provider network attached) peers depends only on their characteristics and does not vary by C-VLAN, an EDE can create that secure connectivity on demand—initiating EAP or starting MKA instances to authenticate and authorize the VLAN connectivity as frames for each VLAN are received—reducing the need to communicate VLAN specific details between administrative organizations. Further restrictions on the use of EAPOL and MKA to support such dynamically created connectivity—including use of pre-shared or cached CAKs and announcements—are beyond the scope of this specification (see IEEE Std 802.1X for detailed capabilities).

Insert the following text, following the existing text of 15.6:

An EDE-CC that provides MAC Privacy protection may be capable of using the C-VID of a user data frame received on the red-side Customer Edge Port (or the absence of a C-TAG on that frame) to select the C-VID (or the absence of a tag) when tagging the MACsec protected frame for transmission on the black-side Provider Network Port. This multiplexing functionality hides the otherwise visible distinction between user data frames assigned to different C-VLANs but destined for the same PrY. The required tagging is configured using the management controls specified by IEEE Std 802.1Q for each of the EDE's bridge components (edge and network) as follows:

a) On the network component:
   1) A distinct Customer Network Port (CNP, see Figure 15-8 and Figure 15-9) is configured for each desired outer, black-side visible, C-VID value with a PVID of that value.
   2) A Static VLAN Registration Entry (8.8.2 of IEEE Std 802.1Q) for that C-VID, with fixed registration that includes only the CNP and the Provider Network Port (PNP) in the member set and specifies untagged transmission on the CNP and tagged transmission on the PNP (with the possible exception of allowing untagged transmission for a single C-VID, allocated for that purpose, on the PNP).

b) On the edge component:
   1) A Provider Edge Port (PEP) for each of the above CNPs, connected to that CNP within the EDE.
   2) A Static VLAN Registration Entry for each red-side visible C-VID, with a fixed registration with a member set that includes only the Customer Edge Port (CEP) PVID and the PEP connected to the CNP whose PVID is the desired outer, black-side visible, C-VID, with tagged transmission on the CEP and PEP.
   3) A Static VLAN Registration Entry for a single C-VID value allocated for untagged transmission, with a member set that includes only the CEP and one PEP.

NOTE 2—The multiplexing functionality provided by this configuration [a) and b), above] can be configured in an EDE-CS by using the Customer Edge Port Configuration managed object (12.13.2 of IEEE Std 802.1Q-2018).

16. Using MIB modules to manage EDEs

16.4 EDE-CC and EDE-SS Management

Change the fourth paragraph of 16.4, splitting it into three paragraphs, as follows:

An EDE-CC that restricts egress for each Provider Edge Port (PEP) to a single red-side C-VLAN as identified by a single C-VID value and the PVID for the Customer Network Port to the same value or an EDE-SS, can be managed without explicitly managing its network component. The static or dynamic instantiation of each Provider Edge Port results in the instantiation of a matching Customer Network Port (CNP) and an internal connection between the PEP and the CNP. The PVID values, egress, ingress, and tagging parameters associated with each of the network component's ports are determined by values for the edge component and the EDE configuration restrictions (15.6, 15.7).

An EDE-CC that supports MAC Privacy protection (21.5, Figure 21-4) also restricts egress for each red-side C-VLAN/C-VID to a single PEP but may [5.8 g)] allow egress for multiple red-side C-VIDs over the same PEP, multiplexing those C-VLANs over the outer, black-side, C-VLAN that supports transmission to the PEP's peer PrY(s). Configuration of this capability requires configuration of both edge and network components of the EDE-CC (15.6, 21.5) using the MIB capabilities provided by IEEE Std 802.1Q.

The network component of an EDE-CC or EDE-SS is identified by a ComponentID of 2 to support the use of additional capabilities, such as CFM or queue service disciplines applied to the Provider Network Port as a whole, which can require or make use of network component management. The red-side port is the only Bridge Port for that component that is identified as a Provider Network Port.

Insert the following text (Clause 17) after Clause 16:

17. MAC Privacy protection

This clause provides an overview of MAC Privacy protection. It provides the context necessary to understand the MAC Privacy protection protocol (Clause 18), the encoding of MAC Privacy protection Protocol Data Units (MPPDUs, Clause 19), and the detailed operation of individual MAC Privacy protection Entities (PrYs, Clause 20), and describes the following:

a) The need for MAC Privacy protection (17.1).

b) How individual user data frames are protected (17.2).

c) Quality of Service impacts and their mitigation (17.3).

d) How MAC Privacy protection is configured (17.4).

Interoperability, deployment, and network configuration requirements are detailed in 18.2 (Data origin authenticity, frame data integrity and confidentiality), 18.5 (Coexistence and use), and Clause 21 (MAC Privacy protection in Systems). Should any conflict be apparent between the text of this clause (Clause 17) and that of Clause 18 through Clause 20, the latter take precedence.

17.1 Need for MAC Privacy protection

Privacy protection is associated with the rights of individual persons to control the disclosure of information associated with themselves and their activities (personally identifiable information, PII). PII can be carried in the user data fields of IEEE 802 MAC data frames and can be hidden from adversaries (persons or organizations attempting to gain unauthorized access to information) by the confidentiality protection provided by MACsec or higher layer protocols (e.g., IPsec). However adversaries can also correlate multiple items of information (personal correlatable information, PCI) in order to construct a ‘fingerprint’ of that person or to correlate their activities with previously computed fingerprints of known activities.

NOTE—IEEE Std 802E [B5] further describes privacy considerations and the use of the terms PII, PCI, adversary, correlation, fingerprint, personal device, and shared service device in the context of IEEE 802 networks.

17.1.1 Privacy and confidentiality

Any potential adversary can still observe the MAC source and destination addresses of a data frame that has been confidentiality protected by MACsec and therefore might, for example, be able to associate the source MAC address of a particular device (a personal device) with an individual and subsequently track that individual and his or her interactions with other individuals or network based services. The sizes of data frames and their transmission timing can also be correlated with particular network applications or the details of those applications (e.g., the size of financial transactions). These adversaries might gain important information simply from the amount of information being sent.

17.1.2 Privacy and organizations

The need to assure privacy is not limited to individuals. Organizations are under an obligation to protect and therefore not to disclose certain information and need to be aware of the extent to which adversaries can draw conclusions by combining multiple observations, each of which might convey little information when considered separately. Organizations that transmit data on behalf of individuals or other organizations are also under an obligation to keep all such data private.

17.1.3 Privacy and network operation

While privacy protection is desired, the operation of networks depends on access by the systems that make up that network (bridges, routers, and switches) to the destination and source MAC addresses of frames to be forwarded. Forwarding systems can require access to information in the frames' user data to associate some frames with data flows and bandwidth reservations. Annex I describes privacy considerations related to the use, design, and deployment of bridged networks in more detail. The requirement for intermediate system access to addresses and user data fields means that all user PCI cannot be simply cryptographically confidentiality protected from its original source to its original destination. Just as for MACsec without additional privacy protection, addresses and data are protected hop-by-hop, although a single hop can extend over a service provider connection with Ethernet Data Encryption devices (EDEs, Clause 15). Authenticated and authorized systems that protect user data frames with MACsec over potentially exposed network connections can also provide MAC Privacy protection as described in this clause. The MAC addresses of the privacy protection systems are exposed to external observation, but these devices can be shared service devices or at least (if they are personal devices) be at fixed locations so (when privacy protection is applied) there is little or no observable correlation between the flow of protected frames and PII.

17.2 Protecting user data frames

In the absence of privacy protection MACsec secures communication while minimizing its impact on the MAC Service's Quality of Service (QoS) parameters (6.10). Individual frames are cryptographically protected and transmitted with minimal delay including only the addition of only those octets required to support cryptographic integrity and confidentiality protection. Each frame's source and destination MAC Addresses remain unmodified.

This deliberately limited impact on the transmission and reception of frames can allow a potentially adversarial observer to correlate some or all of the following:

— Source and destination MAC addresses
— Patterns of frame sizes
— Transmission timing and transmission frequency

with:

— The identities of communicating users
— The reason they are communicating
— The content (in some cases) of confidentiality protected communication

When protecting privacy is paramount, QoS and simplicity of network configuration can be less important. MAC Privacy protection Entities (PrYs) can enhance privacy by the following:

a) Encapsulating user data frames, and their source and destination MAC addresses, within MAC Privacy protection Data Units (MPPDUs).

b) Padding MPPDUs to fixed sizes before their contents are confidentiality protected by MACsec.

c) Controlling the timing of MPPDU transmissions.

The MAC Source Address of each MPPDU identifies its encapsulating PrY, and the MAC Destination Address identifies its decapsulating PrY(s) with a unicast or multicast address. When MPPDUs are confidentiality protected by MACsec, the encapsulated user data frames' source and destination MAC addresses and frame sizes are hidden from an observer who lacks the protecting secret key. Figure 17-1 shows the addition of PrYs to the interface stacks of two bridges protecting frames exchanged with MACsec.

NOTE—The formal term MPPDU is used to avoid confusion with terminology defined elsewhere, and for consistency with similar terms.

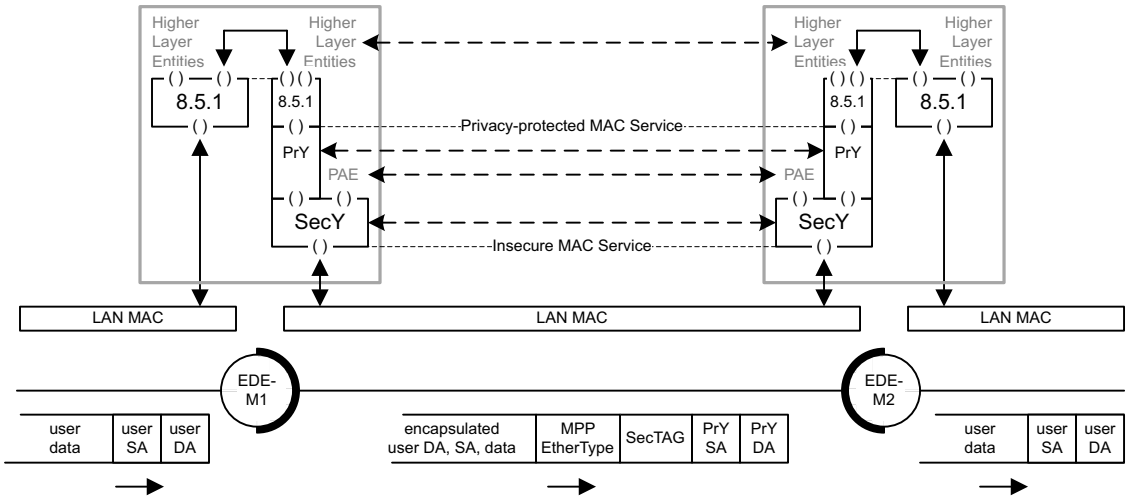
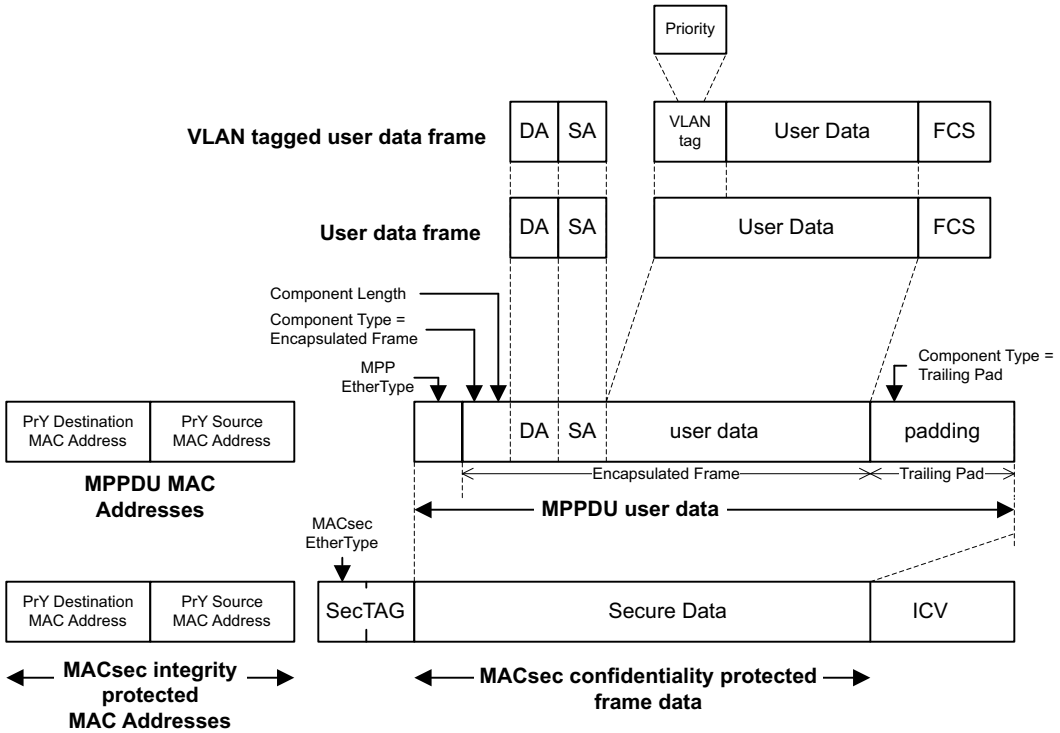


Figure 17-1—Privacy-protected communication between bridges

Figure 17-2 shows a user data frame (DA, SA, and user data) followed by padding, in an MPPDU that is then protected by MACsec.



Source and destination MAC addresses are shown as separate from user data. The supporting service encodes these separate ISS service request parameters into a frame and can add octets between them. Strictly this standard specifies service primitive parameters, not frames. However, it is often convenient to talk of these parameters as a frame.

The MPPDU shown encapsulates a single encapsulated user data frame. More than one user data frame can be encapsulated in a single MPPDU and user data frames can be fragmented and carried in successive MPPDUs (19.5).

Figure 17-2—A privacy protected user data frame

If the original user data frame included a VLAN tag, the user priority information in that tag is carried in the frame's MAC Service user data and will be unchanged when the receiving PrY decapsulates the user data frame. The received user data frame’s user priority does not depend on the access priority used to transmit the confidentiality protected MPPDU.

The length of the encapsulated data frame component is carried explicitly, so padding can be added to prevent observation of the original data frame size once the MPPDU has been protected by MACsec. This removes or reduces the contribution that observation of MPPDU frame sizes can make to inferences about the communication. An MPPDU can also be sent without any user data, including only padding, to guard against observation of user activity.

The media-dependent FCS of the user data frame is not carried within the MPPDU. The MACsec Integrity Check Value (ICV) provides superior protection against deliberate or inadvertent modification of data.

Hiding an original user data frame's MAC destination and source addresses is not necessarily a privacy requirement since the addresses can identify shared service devices, such as routers, and can be the same for all the original user data frames transiting a given link. However MPPDUs always encapsulate these addresses. Encapsulated data frames are independent of one another and, once received, of other user frames or information carried in the same MPPDU. The self-describing format of MPPDUs permits a range of PrY transmission strategies compatible with mandatory PrY reception behavior. However, fine tuning of MPPDU sizes and transmission timing can itself reveal information about the intended mix of user protocols, as can any temporary adjustments to those parameters to better accommodate those protocols.

MAC privacy protection is most effective when all the user data frames passing between two PrYs are protected. A network administrator can decide that MACsec integrity and confidentiality protection is sufficient for some user protocols, and that privacy protection is not to be applied to those protocols. Such protocols might already use frames of fixed sizes and MAC addresses that provide an external observer with little information (e.g., use of the Nearest Customer Bridge group address as the MAC Destination Address). A receiving PrY accepts such frames without modification.

17.3 Quality of Service impact and mitigation

MAC Privacy protection can impact quality of service by:

a) Increasing the number of octets required to encode individual user data frames within MPPDUs.

b) Adding padding to MPPDUs.

c) Requesting MACsec protected MPPDU transmission using a different priority (the access priority) from the original user priority.

   NOTE—The user priority accompanying each transmit request (see IEEE Std 802.1AC) is locally determined or derived from the value associated with a received and subsequently forwarded frame as encoded, e.g., in the received frame’s VLAN tag (see IEEE Std 802.1Q).

d) Hiding the value of the drop eligibility parameter associated with individual user data frames

e) Transmitting MPPDUs at intervals that are fixed, or at least uncorrelated with the user data frame sizes or transmission queue occupancy.

17.3.1 MPPDU encoding and padding

The encoding and padding octets added by privacy protection can be viewed either as delaying the completion of transmission or as delaying the completion of reception. On reception, a privacy protected user data frame cannot be passed to the receiving service user before the MACsec ICV protecting the entire MPPDU has been validated. The additional transmission delay is not necessarily of significant concern on

network connections that are most exposed to unauthorized observers and have a high bandwidth delay product, e.g., across a Provider Bridge Network (PBN), where the effect of the delay can be equivalent to a modest increase in the distance between MAC Privacy protection systems.

Additional octets also reduce the total bandwidth available, potentially delaying subsequent frames. Multiple user data frames can be encapsulated in a single MPPDU, using space that might otherwise be occupied by padding octets. Encapsulating multiple data frames in this way can improve bandwidth efficiency as the media-dependent overhead of sending separate frames, as well as the number of octets required for MACsec protection, is amortized over several user data frames.

17.3.2 MPPDU transmission priority

The access priority (i.e., the priority used to request transmission) of the MPPDU encapsulating a user data frame can be the same as that of the frame's user priority. However, configuring a PrY to transmit fixed sized MPPDUs of several different access priorities on a fixed schedule for several access priorities could degrade the effectiveness of user priority: transmission of a scheduled low priority MPPDU might postpone the transmission of a high priority user data frame. If the next user data frame to be transmitted is to be encapsulated in an MPPDU with a different access priority, the opportunity to encapsulate an additional user data frame in the MPPDU (instead of adding padding) would be lost; and the number of MPPDUs transmitted without conveying a user data frame (i.e., containing only padding) can increase.

NOTE—A PrY does not dictate the transmission order of user data frames or buffer those frames to improve MPPDU packing. In a bridge, the next frame to transmit through an interface is determined by the transmission selection process (see IEEE Std 802.1Q) when an opportunity to transmit arises.

17.3.3 Encoding multiple user data frames in a single MPPDU

If successive user data frames are to be encoded in MPPDUs with the same access priority they can also be encoded in the same MPPDU. All PrY implementations are capable of encoding multiple successive user data frames in the same MPPDU (subject to MPPDU size constraints), if those user data frames are available for transmission before transmission of the MPPDU begins. A PrY can also encapsulate user data frames that become available for transmission after MPPDU transmission has started, and can add padding at the beginning of the MPPDU's data field (if scheduled transmission has begun before a user data frame is available) or between encapsulated data frames.

The late addition of a user data frame to an in-progress MPPDU transmission reduces the transmit delay that would otherwise be experienced by that user data frame. A series of pads can be added between encapsulated frames to allow for the earliest addition of unscheduled frames. To limit the implementation-dependent workload imposed on a receiving PrY that has to scan through multiple pads, any two back to back pads should have a combined length of greater than 64 octets (see Clause 19).

User data frames can be fragmented, using space in fixed sized MPPDUs that would otherwise be padded, reducing the fluctuations in available bandwidth that might otherwise result from the interaction of separate user flows. Frames that can be fragmented are identified as preemptable frames or (more urgent) express frames. One or more express frames that become available for transmission after a preemptable Frame Fragment (19.5.4) in one MPPDU has been transmitted and before a following scheduled MPPDU transmission can preempt subsequent preemptable Frame Fragments. Express Frame Fragments can also be conveyed in MPPDUs transmitted with a higher access priority so their reception can be interleaved with the reception of previously transmitted preemptable Frame Fragments.

NOTE—In this standard all unqualified references to "fragments" and "fragmentation" are to MPPDU encoding.

The individual fragments that compose a given user data frame are transmitted with an incrementing sequence number, with separate sequence number spaces for express and preemptable Frame Fragments (19.5.4), and Initial and Final fragment marking of the first and last fragments. A conformant PrY is required to be able to reassemble a frame from a sequence of fragments received with successively incrementing sequence numbers, with Initial and Final marking. Sequence numbering is not required for unfragmented frames.

NOTE 1— A fragment marked as both Initial and Final is a complete valid frame.

NOTE 2—When a MAC Privacy protection system uses link aggregation (11.5, 21.4), PrYs (like SecYs, see 11.5) are positioned below the Link Aggregation Collection and Distribution functions so all the fragments of any given frame are transmitted through a single interface, and are received in order if the underlying service is order preserving.

NOTE 3—The MPPDU encoding described does permit reassembly of frame fragments received out of order.

17.3.4 Drop eligibility

A single MPPDU can convey multiple user data frames (17.4.2) with different values of the drop_eligible parameter (see IEEE Std 802.1AC). In that case the values of the drop_eligible parameter of those user data frames does not influence the choice of MPPDU access priority, and the drop_eligible parameter associated with the MPPDU transmission is always False.

If an encapsulated user data frame contains a VLAN tag, the priority and drop_eligible information in that tag is available to the receiving PrY's service user.

NOTE—Frames can be subject to traffic shaping, e.g., by the Forwarding Process of a bridge component (see IEEE Std 802.1Q). Traffic shaping can also be performed after user data frames are encapsulated in MPPDUs, e.g., by the network side component of an EDE-CS or EDE-CC.

17.3.5 MPPDU transmission scheduling

The transmission of MPPDUs can be delayed to allow their transmission at regular intervals (17.4), but is not otherwise delayed (e.g., in the expectation of further data frames becoming available for transmission in the same MPPDU). Explicit Pads (i.e., pads whose length is specified) can be encoded in an MPPDU so user data frames that become available for transmission after part of an MPPDU has been transmitted can be encoded in that MPPDU, reducing the delay experienced by user data frames associated with time-sensitive streams. If the PrY is part of an interface stack supporting the transmission of user data frames subject to the operation of a stream gate control list, MPPDU transmissions can be scheduled to allow the encoding of those user data frames towards the end of the MPPDU. Similarly, eligibility times for user data frame transmission can reflect the fact that those frames will be received after the MPPDU has been completely received, validated, and decoded.

NOTE—IEEE Std 802.1Q specifies stream gate control lists in support of Per-Stream Filtering and Policing (PSFP) and the calculation of eligibility times for Asynchronous Traffic Shaping (ATS).

17.4 Configuring MAC Privacy protection

MAC Privacy protection can be used wherever MAC Security can be deployed, with point-to-point, multipoint, or point-to-multipoint connectivity between peer PrYs. The use of privacy protection is an administrative decision, resulting in management configuration of each transmitting PrY. A PrY is always capable of transparently receiving user data frames that have not been privacy protected, so systems that incorporate PrYs can be deployed before privacy protected transmission is enabled.

A PrY is also always capable of receiving MPPDUs addressed to that PrY. A transmitting PrY can choose privacy protection and encapsulation strategies (e.g., protection or not, padding, MPPDU sizes, control over transmission timing) that depend on user data frame attributes without having to communicate those choices to the receiving PrY(s). This standard facilitates network management by specifying management controls

that each conformant PrY is required to support (i.e., can be configured to use). A configurable Privacy Selection Table (17.4.3) allows per user priority selection of one of the following ways of transmitting a user data frame:

a) Unmodified, without MAC Privacy protection.

b) In a Privacy Frame (17.4.1).
A Privacy Frame encapsulates a single user data frame, hiding that frame's MAC addresses, and can be padded to obscure its size.

c) In a Privacy Channel (17.4.2).
A Privacy Channel provides control over transmission timing, can carry multiple user data frames in a single MPPDU, and can fragment user data frames over successive MPPDUs to reduce the bandwidth inefficiencies and variations that can result from using fixed size MPPDUs.

NOTE 1—User priority commonly distinguishes frames associated with broad application classes (see IEEE Std 802.1Q) such as network critical traffic or flows requiring bandwidth allocation.

A receiving PrY processes all MPPDUs alike: the use of any given MPPDU as a Privacy Frame or to support a Privacy Channel is not encoded in the MPPDU. Each MPPDU comprises a MAC Privacy protection EtherType followed by one or more MPPDU components (19.5). Each of these components (conveying entire user data frames or fragments of those frames) is self describing, and can be separately extracted from the received stream of MPPDUs before each user data frame (reassembled from fragments if necessary) is passed to the PrY's user. A transmitting PrY may support additional MPPDU encapsulation and scheduling algorithms in addition to those that can be configured using the Privacy Selection Table, Privacy Frames, and Privacy Channels as specified in this standard.

NOTE 2—The specified PrY reception behavior permits a range of transmitting PrY encapsulation, MPPDU sizing, and scheduling algorithms beyond that specified for standardized management. If additional privacy protection enhancements require a transmitting PrY to know a receiving PrYs capabilities, the MACsec Key Agreement protocol (MKA) provides a suitable secure transport for standardized parameters.

Each user data frame can include priority information encoded in a VLAN tag, allowing its recovery by, for example, the Bridge Port Transmit and Receive process of a VLAN Bridge (IEEE Std 802.1Q). The priority used to transmit an MPPDU is not used on receipt and could have been modified as the MPPDU was forwarded to the receiving PrY.

The destination MAC Address of MPPDUs transmitted by a PrY located in the same interface stack as its associated SecY is the PAE Group Address configured for use by MKA in support of that SecY (Table 15-2, Table 15-3), and does not need to be separately configured. If MKA does not operate and discover that SecY's peers, MPPDU encapsulation of user data frames is disabled. MPPDU encapsulation of data frames is enabled (if administratively configured) if the SecY's peers are present, even if confidentiality protection is not being provided. The destination MAC Address used by a transmitting PrY that is in a separate system from its associated SecY (21.8) can be configured to be an individual address or a group address.

NOTE 3—The use of the PAE Group Address with the check that peer SecYs are using that address, defends (in the case of SecY and PrY collocation) against the inadvertent broadcast of MPPDUs throughout the connected network.

A PrY relies on its associated SecY's operation of MACsec for the data integrity, data confidentiality, and data origin authenticity of the MPPDUs transmitted by peer PrYs. No additional parameters are required for MAC Privacy protection protocol operation.

17.4.1 Privacy Frames

A Privacy Frame encapsulates a single user data frame.

A Privacy Frame's MPPDU components comprise an Encapsulated Frame (19.5.1) and, if required, a Trailing Pad (19.5.2). The PrY management controls specified by this standard provide for a user priority dependent constraint (privacyPadding, 20.7) on the size of the MPPDU to four octets [to allow for the MAC

Privacy protection EtherType and the MAC Privacy protection Protocol Component Identifier (MPPCI)] plus a multiple of 16, 32, or 64 octets. The specified size does not include the original user data frame's FCS (not carried in the MPDU), the MPPDU's own source and destination MAC addresses, and SecTAG and ICV added by MACsec when confidentiality protecting the frame, nor does it include additional tags that can be subsequently added by other components of a system prior to physical transmission such as a VLAN tag added by the network component of an EDE (Clause 15). The maximum size of the MPPDU (for a given user priority) is the maximum size of the user data frame specified for the PrY's user's Private Port interface (20.1) plus the number of octets required to encode and pad that maximum sized frame.

NOTE—The number of octets that the PrY uses to encapsulate a given user data frame in a Privacy Frame can be calculated from the minimum protected frame size and size increment specified in the Privacy Selection Table. Those parameters are available to other system components through the Pry's LMI (there is little reason to limit their visibility within the system of which the PrY is a part, since they could probably be deduced by an adversary observing the transmitted traffic) and can be used, e.g., by a bridge's MAC Relay Entity (8.6.8), as part of scheduling transmissions.

The use of Privacy Frames as opposed to Privacy Channels to convey individual user data frames reflects a decision on the part of the network administrator that any leakage of PCI resulting from changes in their sizes or transmission timing is unimportant (e.g., they might encapsulate network routing traffic, rather than packets sent by network applications). Where the underlying service provided to communicating PrYs does not usefully distinguish between frames transmitted with different priorities, Privacy Frames that encapsulate user data frames of different user priority can be sent with the same access priority, concealing the original priority from any observer. Where the original user priority is encoded in a VLAN tag in the user data frame, that priority can be recovered by the receiving PrY's user.

Privacy Frames, and their component Encapsulated Frames, are not queued for transmission by a PrY. Their transmission, via a transmit request at the PrY's supporting Controlled Port, is a direct result of the transmit request made by the PrY's user at its Private Port interface.

17.4.2 Privacy Channels

A Privacy Channel facilitates the regular transmission of fixed sized MPPDUs.

A Privacy Channel's MPPDU's components comprise zero or more Encapsulated Frames, zero or more Encapsulated Frame Fragments, zero or more Explicit Pads, and zero or one Trailing Pad (19.5). A conformant PrY can be configured to support a Preemptable Privacy Channel, and/or an Express Privacy Channel. If both Privacy Channels are enabled, user data frames identified by the Privacy Selection Table (17.4.3) as allocated to a given Privacy Channel are conveyed by that Privacy Channel. If a single Privacy Channel is enabled frames allocated to either Privacy Channel are conveyed by the enabled Privacy Channel, and if neither Privacy Channel is enabled they are transmitted as Privacy Frames using the other parameters (accessPriority, revealDE, and privacyPadding) configured for each user priority in the Privacy Selection Table.

Each Privacy Channel is characterized by the following parameters:

a) The user data frame size (userDataFrameSize) in octets.
   This parameter specifies the maximum sized user data frame that can be encapsulated in a Privacy Channel MPPDU as an Encapsulated Frame (19.5.1). It includes the MAC addresses, user data, and FCS of the user data frame. The fixed size MPPDU used (prior to the application of MACsec confidentiality protection, and the FCS added by the specific media access method used to transmit the protected frame) includes 12 additional octets (including the PrY MAC Addresses, MAC Privacy protection EtherType, Encapsulated Frame component identifier, but excluding the original FCS which is not encoded in the MPPDU).

b) The requested bit rate (requestedKbitRate) in kilobits per second.
   In the absence of individual user data frames that are not privacy protected or that are conveyed as Privacy Frames, MPPDUs for the Privacy Channel are transmitted at regular intervals determined by

the userDataFrameSize and requestedKbitRate when the default Channel MPPDU generation algorithm is used. See 20.9.4 for detailed calculations.

NOTE 1—Individual frames that can compete for transmission bandwidth can be sent by the PrY's user, but can also be sent by other protocol entities that insert frames for transmission lower in the interface stack, e.g., periodic and as required MACsec Key Agreement (MKA) transmissions through the SecY's Uncontrolled Port. Any resulting delay in the transmission of Privacy Channel MPPDUs does not reveal information about the content of those MPPDUs.

NOTE 2—The requestedKbitRate and userBurstOctets parameters are used to provide compatibility with the specifications for handling time-sensitive traffic in IEEE Std 802.1Q, including Asynchronous Traffic Shaping (ATS). Privacy Channel MPPDU transmission can also be scheduled to support the use of transmission gates and a gate control list (20.9.5). A bridge that supports per-stream filtering and policing (PSFP, IEEE Std 802.1Q) can use a number of criteria including priority to gate the transmission of frames of individual streams by bridge ports, and thus remove or reduce the impact of potential PrY scheduling conflicts on time-sensitive streams.

c) The permitted burst size (userBurstOctets).

   This parameter allows temporarily rapid transmission of Privacy Channel MPPDUs to recover bandwidth lost to competing frames (20.9.4).

d) The access priority used to transmit the Privacy Channel's MPPDUs.

   NOTE 3—If the PrY and SecY form part of the Provider Edge Port of an EDE-CS or EDE-CC (see Figure 15-6, Figure 15-7, and Figure 15-8) the SecY's transmitted access priority is encoded by the black-side bridge component in the outer VLAN tag added to the protected frames transmitted by the Provider Network Port.

The underlying service used by the PrY is expected to preserve the transmission order of frames of any given priority. If the interface stack (or the interface stacks of frame forwarding devices on the path between peer PrYs) supports IEEE Std 802.3 frame preemption or other class of service differentiated forwarding capabilities, the access priority of the Express Privacy Channel can be configured to take advantage of those capabilities.

17.4.3 Privacy Selection Table

A configurable Privacy Selection Table has an entry for each of the eight possible user priority values (0 through 7). Each entry indicates whether user data frames of that priority are transmitted unmodified, in a Privacy Frame, or in a Privacy Channel (17.4). If the frames are to be transmitted in a Privacy Channel they are classified as preemptable frames or express frames, in case they need to be fragmented (17.4).

Figure 17-3 illustrates the use of a PrY's Privacy Selection Table in conjunction with the Traffic Class and Access Priority Tables specified for a SecY (10.7.17). Each user data frame transmission request to the PrY (from, for example, the Bridge Port Transmit and Receive function as illustrated in Figure 17-3 and specified by IEEE Std 802.1Q) is shown at the top of the figure. Frames with a user priority of 0 or 1 are assigned to the Preemptable Privacy Channel, with access priority 0. The PrY is the user of the service provided by the SecY, so that communicated priority value is the SecY's user priority, and is used (by the SecY's Traffic Class Table) to assign the MPPDUs conveying to Secure Channel (SC) 0 requesting priority 0 (as specified by the SecY's Access Priority Table) from the underlying medium when the MACsec protected frames are transmitted.

Similarly, the PrY assigns frames with user priorities of 2 through 4 to the Express Privacy Channel. They are conveyed by that Privacy Channel (since it has been configured) and transmitted in MPPDUs with access priority 3. The SecY assigns these to SC 1.

NOTE—Different values of PrY access priority are only relevant if they select differentiated services from a supporting SecY (between two transmit SCs, for example) or from the media access method or other protocol entity that supports the SecY's Controlled Port (providing frame preemption, for example).

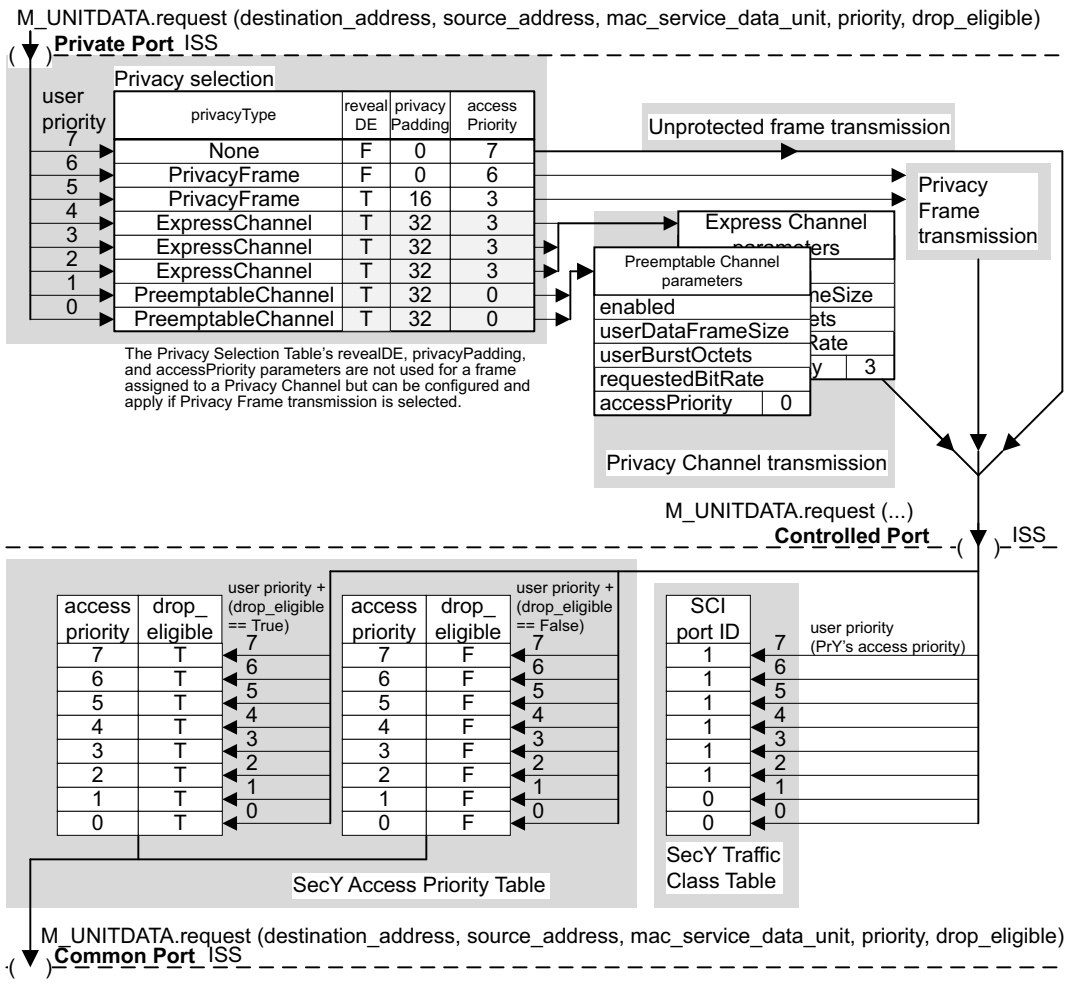


Figure 17-3—Privacy selection, priority and traffic class mapping

In Figure 17-3 user data frames with a user priority of 5, 6, or 7 are not assigned to a Privacy Channel. Their transmission is not constrained by a channel schedule and can occur at any time, even though (with the configuration shown) they are assigned to the same SC. Their transmission can delay a Privacy Channel's scheduled transmissions, but this timing conflict and delay does not expose Privacy Channel PCI, unless the original source(s) of those frames correlated their transmission with frames carried by the Privacy Channel.

Insert the following text (Clause 18) after Clause 17:

18. MAC Privacy protection protocol

The MAC Privacy protection protocol comprises rules for the following:

a) Encapsulation of user data frames in MAC Privacy protection Protocol Data Units (MPPDUs)
b) Addressing of MPPDUs
c) MPPDU decoding, and the recovery of user data frames from a stream of MPPDUs

A given user data frame can be encoded in one or more MPPDUs, each with the same source and the same individual or group destination MPPDU MAC Address. Once protected by MACsec, these MPPDUs enhance the privacy of user communication between that source and destination(s). The structure of MPPDUs is self-describing and allows a recipient of MPPDUs to recover encapsulated user data frames without the need to share additional parameters with their source.

Clause 17 provides an overview of MAC Privacy protection, and explains why user data frames need to be encapsulated in MPPDUs as well as being integrity and confidentiality protected by MACsec. MPPDU encoding is specified in Clause 19 and their transmission and reception by Privacy protection entities (PrYs) in Clause 20. Clause 21 describes how MAC Privacy protection is supported in a number of interoperable interface stack and system configurations.

This clause adds details of design and support requirements not already described in Clause 17 and provides additional context for other clauses in this standard, as follows:

— Addressing (18.1)
— Data origin authenticity, frame data integrity and confidentiality (18.2)
— Applicability (18.3)
— Bandwidth utilization, fragmentation, and transit delay (18.4)
— Coexistence and use (18.5)

18.1 Addressing

The intended recipient of an MPPDU is identified by the destination MAC Address (DA) of the transmitted MPPDU, and the transmitter by its source MAC Address (SA). The recipient identifies those MPPDUs that can convey a given fragmented user data frame by their DA, SA pair. The DA can be a group address and can identify one or more recipients. All MPPDUs transmitted by a given PrY use the same DA and SA.

A receiving PrY only decodes MPPDUs addressed to that PrY (i.e., MPPDUs with an individual or group MAC Destination Address associated with that PrY). MPPDUs destined for other addresses are passed to the PrY's user without modification. This forwarding permits system by system deployment of privacy protection capability in a network, and layered deployment as illustrated for MACsec in Figure 11-12 and Figure 11-13.

NOTE 1—Bridge components within the transmitting system can add SecTAGs or VLAN tags to the transmitted MPPDUs, e.g., as described in Figure 17-2 and Clause 21. Network components between the transmitter and the intended recipient(s) can also add tags, and use those tags to direct frames to a particular network region, potentially avoiding MAC address allocation conflicts, e.g., between different customer networks attached to a Provider Bridged Network (PBN). Peer network components are responsible for removing those added tags before the MPPDU is delivered to the intended recipient(s).

NOTE 2—A SecY can use SecTAGs with different SCIs to support MACsec with multiple access priorities while retaining strict delivery and replay protection within an access priority. The normative text above (18.1) deliberately excludes both the use of the SCI to distinguish streams of MPPDUs between the same DA, SA pair and any suggestion that an implementation needs to be capable of reassembling fragmented user data frames on a finer granularity than

provided for by the combination of DA, SA, and the Express/Preemptable classification (19.5.4). All Frame Fragments with the same value for Express/Preemptable transmitted between a given DA, SA pair are conveyed in MPPDUs transmitted with the same access priority.

When a PrY is directly supported by a SecY, the destination MAC Address (DA) used to transmit MPPDUs and to recognize received MPPDUs sent to that PrY shall be the PAE Group Address configured for use by MKA (see IEEE Std 802.1X) in support of that SecY.

When a PrY is directly supported by a SecY, the source address of each MPPDU that it transmits shall be the MAC Address used to transmit MKPDUs and to generate the SCI(s) used by that SecY. If the secure Connectivity Association (CA) established for MACsec use is a group CA, MKA checks that each SCI is unique within the scope of the CA, and so establishes the required uniqueness for use of the MPPDU SA to identify the reassembly context for fragments received from peer PrYs. Thus neither the MPPDU DA nor the MPPDU SA need be configured. If the receiving interface stack is a Bridge Port that does not include an active PrY, received MPPDUs are discarded and not relayed by the Bridge.

When a PrY is separated from the supporting SecY (see 21.8), the destination MAC Address of the MPPDUs that the PrY transmits needs to be configured. Care needs to be taken to avoid the use of an unrecognized individual or group address that could be flooded throughout a network. Successful configuration of privacy protected connectivity can be verified, if necessary, by sending a suitable CFM message (see IEEE Std 802.1Q) in Privacy Frames to elicit a response from each peer.

When a PrY is separated from the supporting SecY, the destination MAC Address of the MPPDUs that it recognizes for reception shall be either the individual MAC address that it uses as the source MAC Address (SA) of the MPPDUs that the PrY transmits or a configured group address (20.14).

When a PrY is separated from the supporting SecY, the source address of each MPPDU that it transmits shall be an individual address that is associated with the interface stack of which the PrY is a part and is persistent across reinitialization of system components.

See 21.1.1 for additional addressing considerations.

18.2 Data origin authenticity, frame data integrity and confidentiality

MAC Privacy protection relies on the use of MACsec and its supporting authentication, authorization, and key agreement protocols to ensure that:

a) MPPDUs are only sent to and received from authenticated and authorized peer PrYs.

b) The MPPDU destination and source MAC addresses and data are received as sent, without modification.

c) The MPPDU data is kept confidential, and only available to the intended recipient PrYs.

18.3 Applicability

Privacy protection can be provided by encapsulating user data frames in MPPDUs wherever MACsec is applicable and does not compromise MACsec’s ability to meet the requirements described in 8.1.

Additional considerations do apply when privacy protection is supported by shared media, as follows:

a) True shared media—where each frame transmission is available to a set of potential recipients without the possibility of intermediate network systems selectively forwarding any given frame to a subset of those recipients.

b) Virtual shared media—where intermediate systems can deliver group addressed frames to one or more sets of potential recipients while selectively delivering frames to individual destination MAC addresses.

NOTE 1—In the case of the Ethernet Data Encryption (EDEs) described in Clause 15, virtual shared media considerations do not apply as the VLAN tagged service selection restricts each EDE PEP to a single peer.

In both cases a system transmitting on the shared media could choose to encapsulate some user data frames in group addressed MPPDUs (for reception by all of its peers) and some (each destined for a specific peer) in individually addressed frames. A requirement to keep the nature and level of current activity private would then require a constant level of Privacy Channel transmission for each DA, SA pair.

In the case of true shared media as, e.g., originally envisaged for Ethernet using CSMA/CD, using Privacy Channels with user data frame fragmentation requires each recipient to be capable of supporting the simultaneous reassembly of at least one Express user data frame and one Preemptable user data frame from each of its peers. The use of individually addressed MPPDUs does not reduce that requirement, and is not recommended.

In the case of virtual shared media the use of individual MPPDU destination addresses can significantly reduce intermediate network resource usage, even when there is a privacy requirement to maintain a constant level of activity. The applicable considerations are similar to those for Provider Backbone Bridged Networks (PBBNs, see IEEE Std 802.1Q) with the additional restriction that successive fragments of any given user data frame need to be conveyed in MPPDUs with the same DA. User data frames with any given destination address should be consistently encapsulated in MPPDUs with the same DA. User data frames can be replicated by a bridge component and transmitted through multiple ports on that component, in separate MPPDUs, if they need to be flooded to multiple destinations.

NOTE 2—Bridged networks can flood frames to individual end stations whose addresses have not been learned, but a newly active station usually transmits one or more group addressed frames to seek peers or acquire configuration information and waits for a response, allowing other bridges to learn its location, before it consumes much bandwidth.

NOTE 3—A Privacy protection entity (PrY, Clause 20) uses the same DA to transmit all MPPDUs.

18.4 Bandwidth utilization, fragmentation, and transit delay

The MAC Privacy protocol allows frames to be padded to arbitrary lengths, and thus allows transmission of fixed sized MPPDUs that conceal the size of user data frames. Use of fixed sized MPPDUs carries with it the possibility of inefficient bandwidth utilization, as the next user data frame to be transmitted might be at or close to the maximum size permitted and not quite fit. MPPDU encoding allows fragmentation, to allow most if not all of the MPPDU to carry user data. The resulting improvement in bandwidth utilization can also reduce the transit delay experienced by individual user data frames.

The MAC Privacy protocol does not constrain the timing of MPPDU transmission, MPPDUs can be transmitted at fixed intervals to conceal the timing characteristics of network applications and the level of network activity. In the interval between MPPDU transmissions one or more high priority user data frames (identified as Express frames) can become available for transmission. Those Express frames, possibly fragmented, can be transmitted before any remaining fragment of lower priority frames (identified as Preemptable frames).

The MPPDU encoding rules (Clause 19) constrain the use of fragmentation and padding to facilitate interoperability, allowing externally observable implementation characteristics to be specified and tested, as follows:

a) The minimum size of each fragment is restricted, as is the minimum size of pads other than the last.
b) The fragments of Express and Preemptable user data frames are independently sequence numbered, and the initial and final fragments of each user data frame identified.

c) A conformant receiver can rely on in-order delivery of MPPDUs, need only be capable of both reassembling one Express and reassembling one Preemptable user data frame from each of its peers, and delivering each user data frame in the order that its reception is complete.

NOTE 1—A conformant implementation using the mandatory to implement default MPPDU encapsulation algorithm (20.10.1, Figure 20-3) encodes at most two fragments in any given MPPDU as follows: a final fragment (Express or Preemptable), followed by zero or more complete user data frames, followed by an initial fragment (Express or Preemptable) followed by zero or more octets of pad. A conformant implementation that fragments Preemptable frames to allow for the latest possible addition of Express frames can transmit MPPDUs that encode minimum sized Preemptable frame fragments interspersed with minimum sized Express frames, followed by zero or more octets of pad.

NOTE 2—An MPPDU that conveys multiple small user data frames can use less bandwidth than the independent transmission of those frames.

NOTE 3—The MAC Privacy protocol does not enforce in-order delivery of user data frames, but a supporting SecY can be configured to provide strict replay protection and in-order delivery of MPPDUs and other frames, resulting in transmission order reception of all user data frames of a given traffic class.

Express and Preemptable fragments are independently numbered, each using a 24-bit sequence number, that is incremented by the transmitter for each fragment sent. A conformant transmitter does not restart or change the order of the sequence numbers used unless at least 2 seconds has elapsed since the MPPDU with the prior sequence number has been transmitted. A conformant receiver retains fragments pending reassembly of a user data frame for at most 0.1 second. If the PrY's Controlled Port's MAC_Operational parameter becomes false, any retained fragments are discarded (20.13, 20.13.1).

NOTE 4—The 24-bit sequence number space does not wrap within the specified fragment reassembly time for a 1 Tb/s transmitter using the default MPPDU encapsulation algorithm and an MPPDU capable of conveying a maximum sized user data frame, even if the traffic load constantly forces the transmission of two fragments in each MPPDU. User data frames that are encapsulated without fragmentation are not sequence numbered.

18.5 Coexistence and use

Privacy protected user data frames can be transmitted between the same pair of stations as, and interspersed with, user data frames that are not privacy protected. The choice of whether any particular frame is privacy protected or not is made by the transmitter and can be changed without reference to the receiver, and without reordering user data frames.

A system that transmits MPPDUs can encode user data frames in those MPPDUs in any way permitted by the encoding rules (Clause 19), encoding one or multiple user data frames in each MPPDU and transmitting each MPPDU with any value of access priority, subject only to the rules for fragmenting user data frames which limit the complexity of the receiving reassembly process. The transmission of a sequence of MPPDUs that convey fragments of any given user data frame (classified as fragments of a preemptable frame or an express frame) can be interspersed with the transmission of user data frames that are not privacy protected or of MPPDUs that do not convey fragments of the same classification, up to the fragment reassembly time limit.

NOTE 1—This standard supports a common approach to transmission behavior, and management of that behavior, with a PrY specification that support transmission of Privacy Frames (MPPDUs that convey a single user data frame) and up to two Privacy Channels (sequences of MPPDUs that convey one or more user data frames with the possibility of fragmentation). However the encoding of MPPDUs does not distinguish Privacy Frames and Privacy Channel MPPDUs, and PrY specification supports the reception and recovery of user data frames from any MPPDU or sequence of MPPDUs encoded as specified in Clause 19.

Insert the following text (Clause 19) after Clause 18:

19. Encoding of MAC Privacy protection Protocol Data Units

This clause specifies the structure and encoding of the MAC Privacy protection Protocol Data Units (MPPDUs) exchanged between MAC Privacy protection Entities (PrYs). It specifies and describes the following:

a) Rules for the structure, representation, and encoding of protocol fields (19.1)

b) The general format of MPPDUs (19.2)

c) The allocation of the MAC Privacy protection EtherType to identify MPPDUs (19.3)

d) The strategy for interoperability with possible future protocol versions (19.4).

e) The encoding of MPPDU components (19.5)

f) Rules for the generation of MPPDUs using the specified components (19.6)

g) Validation and extraction of MPPDU components on reception (19.7)

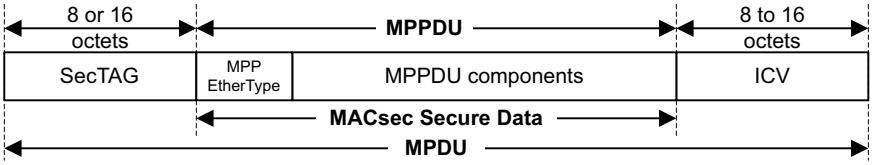
NOTE—The MPPDU validation checks specified do not overlap with the specification of PrY operation (Clause 20).

19.1 Structure, representation, and encoding

All MPPDUs contain an integer number of octets. Octets and bits in the text and figures in this specification are represented and numbered, and values are encoded, using the conventions specified in 9.1.

19.2 MPPDU Format

Figure 19-1 shows a MPPDU that has been confidentiality protected by MACsec. The MACsec Secure Data includes the MAC Privacy protection EtherType (19.3) and one or more MPPDU components (19.5). Each component encodes a user data frame (or a fragment of a user data frame) or padding.



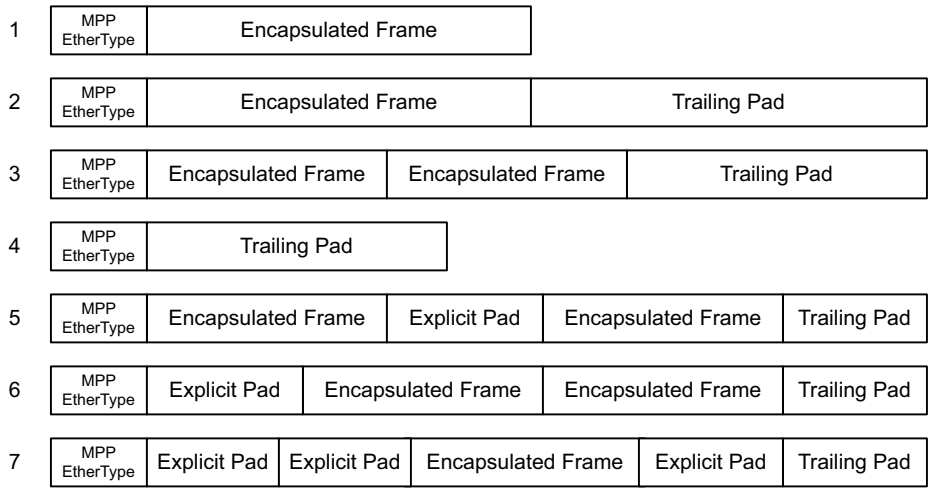
The source and destination MAC addresses of the communicating PrYs are not shown, as these are separate parameters of the service requests and indications that support transmission and reception of the MACsec protected MPPDU (see Figure 8-1).

Figure 19-1—MACsec protected MPPDU

Figure 19-2 illustrates some possible MPPDU component combinations.

The first two examples show a single Encapsulated Frame, concealing (once the MPPDU has been MACsec confidentiality protected) the encapsulated user data frame's source and destination MAC addresses. The addition of a Trailing Pad in the second example obscures the length of that frame.

The third example shows multiple Encapsulated Frames in a single MPPDU, again with a Trailing Pad. An observer of the confidentiality protected frame is unaware of how many user data frames have been encoded or how much padding has been used. The only MPPDU component in the protected frame might be a Trailing Pad, as shown in the fourth example, sent to obscure changes in network activity when no user data frame is available for transmission. The remaining examples show the use of Explicit Pads, each with a specified length, used to allow the possible later addition of one or more Encapsulated Frames (19.5.1).



In each of these examples one or more of the Encapsulated Frame components could have been a Frame Fragment (19.5.4)

Figure 19-2—MPPDU Examples

NOTE 1—Not all PrYs and PrY configurations need to be capable of transmitting MPPDUs with all the component possibilities shown. A conformant PrY (5.10, 5.11, Clause 20) need only be capable of transmitting Privacy Frames and sequences of MPPDUs that support Privacy Channels. A Privacy Frame (17.4.1) is an MPPDU that includes a single Encapsulated Frame, conveying a single unfragmented user data frame, with a possible Trailing Pad. A Privacy Channel (17.4.2, 20.8, 20.10) can be supported without the use of Explicit Pads. However all conformant PrYs are required to be capable of receiving all correctly encoded MPPDUs (17.4).

NOTE 2—The use of any given MPPDU as a Privacy Frame, or to support a Privacy Channel, or as part of any other encapsulation strategy is not encoded in the MPPDU (17.4).

19.3 MAC Privacy protection EtherType

The MAC Privacy protection EtherType (Table 19-1) comprises octet 1 and octet 2 of each MPPDU. This EtherType is included within the MPPDU to allow:

a) Coexistence of MAC Privacy protection capable systems in the same environment as other systems.
b) Incremental deployment of MAC Privacy protection capable systems.
c) Transmission of privacy protected and unprotected user data frames on the same media and between the same systems, using the same system addresses.

Table 19-1—MAC Privacy protection EtherType allocation

Name	Value
MAC Privacy protection EtherType	E2-3B

The encoding of the MAC Privacy protection EtherType in the MPDU is illustrated in Figure 19-3.

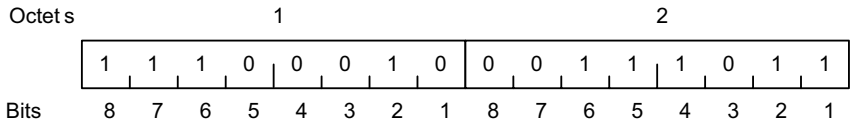


Figure 19-3—MAC Privacy protection EtherType encoding

19.4 Protocol Version strategy

This standard specifies basic MAC Privacy protection protocol capabilities and the MPPDU components that support those capabilities. MPPDUs do not include a protocol version identifier. Future revisions of this standard can specify additional MPPDU component types while supporting interoperability with implementations conformant to this standard by correctly recognizing and processing the MPPDU components specified in this clause (Clause 19).

The general format of MPPDU components is extensible (19.5), allowing future revisions of this standard to specify additional MPPDU components if necessary. This general extensible format includes a component length, intelligible to implementations conformant to the initial specification of the protocol and all future versions. All conformant implementations are thus capable of discarding unrecognized component types, while continuing to process other components that can be encoded later in an MPPDU.

This protocol version strategy is not limited to supporting the incremental addition of capabilities and component types in future revisions of this standard. It permits the addition of independent optional sets of capabilities, if that proves desirable. However, the set of basic capabilities specified in this standard is conveniently identified as version 0 for management purposes.

NOTE—Any future need to communicate protocol capabilities between PrYs can be met by adding new MPPDU component types, by using an encoding that is discarded by this initial protocol version, or by adding attributes to MKA (see IEEE Std 802.1X).

19.5 MPPDU component encoding

An MPPDU component is a Trailing Pad (19.5.2) if the first octet of the component is zero and there are no more octets in the MPPDU, or the first two octets are both zero. Otherwise the component has the general format shown in Figure 19-4, with a component identifier (MPPCI) followed by zero or more octets.

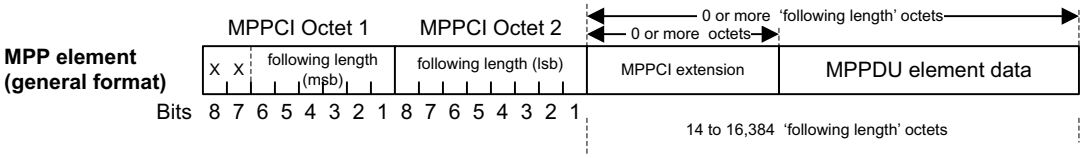
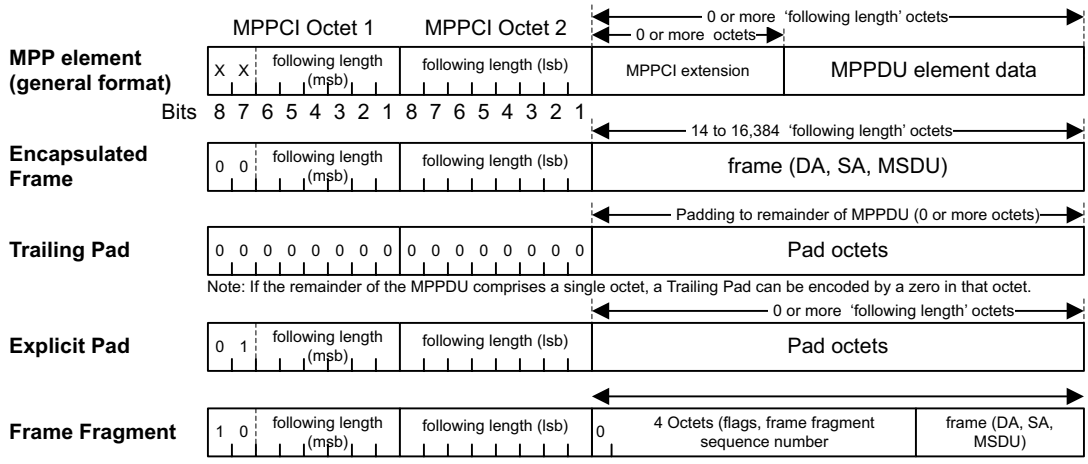


Figure 19-4—MPPDU component format

In the general format, bit 6 (most significant) through bit 1 of the first octet of the MPPDU and bits 8 through 1 (least significant) of the second octet always encode the length of the component following the first two octets. This length field allows a receiver to skip any unrecognized MPPDU component and process any following components present in the MPPDU. The components and component identifiers specified by this standard are shown in Figure 19-5 and specified in detail in 19.5.1–19.5.4.

Encapsulated Frames, Explicit Pads, and Frame Fragments can each be present zero or more times in an MPPDU and their relative position(s) in the MPPDU are not constrained. If a Trailing Pad is present, it is the last component in the MPPDU. Constraints on the use of these components in the generation of MPPDUs are specified in 19.6. Validation of received MPPDUs and of the components they contain, and the extraction of those components is specified in 19.7.



An Encapsulated Frame has a following length of 14 octets or greater (19.5.1). While this encoding is capable of encapsulating user data frames with lengths from 14 to 16,383 octets in length, that does not imply that any specific media access control method can support frames with that entire range of lengths.

Each Frame Fragment conveys a fragment of a frame that is at least 64 octets in length (19.5.4).

The use of the bit pattern 11 in bits 8 and 7 of Octet 1 is reserved for future specification, as is the use of 10 in those bits with 1 in bit 8 of the third octet of the component.

Figure 19-5—MPPDU component encoding

19.5.1 Encapsulated Frame

An MPPDU component is identified as an Encapsulated Frame if:

a) Bit 8 and bit 7 of the first octet of the component are both zero, and

b) Bits 6 (most significant) through 1 of the first octet and bits 8 through 1 (least significant) of the second octet (least significant) encode a *following length* value of 14 or greater.

The encapsulated frame follows, beginning in the third octet of the MPPDU component. The first 6 octets of the frame encode a 48-bit MAC Destination Address in the standard encoding, i.e., in the canonical format (also Figure 10, 8.2, and Annex C of IEEE Std 802-2014), and the next 6 octets encode a 48-bit MAC source address in the standard encoding. The two following octets are the initial octets of the MSDU (the Length/Type field of IEEE Std 802.3 frames), and any subsequent octets are the remaining octets of the MSDU. The original user data frame's FCS is not encoded in the component.

The use of bit 8 and bit 7 of the first octet of an MPPDU component, together with a following *length value* of 1 through 13 is reserved for future specification. The reception of such a component is specified in 19.7.

19.5.2 Trailing Pad

An MPPDU component is a Trailing Pad if:

a) The first and second octets of the component are both zero, or

b) The first octet is zero and there are no more octets in the MPPDU.

Any octets in the MPPDU following the second octet are part of that Trailing Pad. A Trailing Pad indicates that no further components are present in the MPPDU (19.6, 19.7).

19.5.3 Explicit Pad

An MPPDU component is an Explicit Pad if:

a) Bit 8 of the first octet is zero and bit 7 is one.

Bits 6 through 1 of octet 1 (more significant) and bits 8 through 1 of octet 2 (less significant) encode the *following length*, i.e., the number of octets following in the Explicit Pad after octet 2.

The octets in the MPPDU following octet 1 of the Explicit Pad, including octet 2 up to the *following length* count octets, are part of that Explicit Pad.

NOTE—The specification of the Explicit Pad component accommodates the possibility that the MPPDU includes fewer remaining octets than encoded in the length field in octets 1 and 2 when it is the last component in a MPPDU.

19.5.4 Frame Fragment

An MPPDU component is a Frame Fragment if:

a) Bit 8 of the first octet is one and bit 7 is zero, and

b) Bit 8 of the third octet is zero.

Figure 19-6 illustrates the format of Frame Fragments.

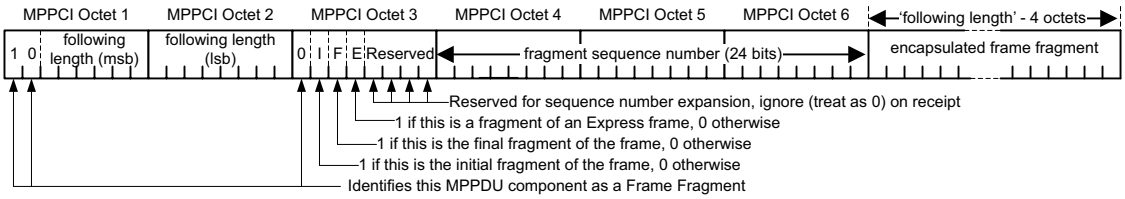


Figure 19-6—Frame Fragment

A user data frame can be conveyed in a sequence of Frame Fragments, with each successive Frame Fragment incrementing (modulo 2^{24}) a fragment sequence number encoded in the fourth (most significant) through sixth octets of the component. Two independent fragment sequence number spaces are supported, each identified by one of the two possible values (zero or one) of bit 5 of the third octet [the Express (E) bit]. The independent sequence number spaces allows the sequence of Frame Fragments encoding a given user data frame to be interrupted by, or interleaved with, Frame Fragments encoding another frame. By convention fragmented Express frames are encoded with the E bit set (i.e., when it is one) and fragmented Preemptable frames are encoded with the E bit clear.

Each encapsulated frame fragment comprises 64 or more octets (see 19.6, 19.7) i.e., the Frame Fragment's *following length* is greater than or equal to 68. A user data frame of less than 128 octets (MAC DA, SA, and MSDU) in length cannot be fragmented.

When bit 7 of the third octet [the Initial (I) bit] is set (i.e., when it is one), the encapsulated fragment is the initial fragment of a user data frame and the initial octets of the fragment (beginning with the seventh octet of the component) comprise that frame's MAC DA, SA, and Length/Type encoded protocol identifier as specified for an Encapsulated Frame (19.5.1).

When bit 6 of the third octet [the Final (F) bit] is set, the encapsulated fragment is the final fragment of a user data frame. A Frame Fragment can be encoded with both the I bit and the F bit set, in which case the entire user data frame (MAC DA, SA, and MSDU, but not any FCS) is present in the seventh octet and following octets of the Frame Fragment component.

The MPPDU format does not identify any given MPPDU as a Privacy Frame (17.4.1) or as supporting a Privacy Channel (17.4.2). This standard distinguishes Privacy Frames and Privacy Channels simply as part of specifying mandatory to implement PrY transmission capabilities. Any given received MPPDU can include Frame Fragments with the Express (E) bit set and Frame Fragments with the E bit clear.

Bits 4 through 1 of the third octet are reserved for future specification of an extended sequence number space, and are ignored on receipt by protocol implementations conformant to this standard.

NOTE—If encoded MACsec protected MPPDUs are received from an Ethernet link running at speeds up to 400 Gb/s, with each MPPDU capable of conveying a non-fragmented 1518 octet user data frame but in fact including two Frame Fragments, the 2^{24} fragment sequence number space specified by this standard takes at least 0.25 seconds to wrap. This remains true even if bits 4 through 1 of the component's third octet are used to encode the more significant bits of an extended sequence number space.

19.5.5 Unrecognized components

The MPPDU validation process (19.7) specified for this edition of this standard treats components with any of the following MPPCI bit patterns as unrecognized:

a) In the first octet bits 8 and 7 are both one, and a second octet is present in the MPPDU.
b) In the first octet bit 8 is one and bit 7 is zero, and in the third octet bit 8 is one.
c) The first octet has the value zero, and the value of the *following length* is between 1 and 13 (inclusive).

These MPPCI values are reserved for specification by future revisions of this standard.

19.5.6 Incorrectly encoded components

The MPPDU validation process (19.7) identifies components with any of the following bit patterns as incorrectly encoded:

a) In the first octet bit 8 is one and bit 7 is zero, a second octet is present in the MPPDU, but there is no third octet.
b) In the first octet bits 8 and 7 are both zero (matching the encoding of an Encapsulated Frame) and the *following length* is greater than the number of octets remaining in the MPPDU.

19.6 MPPDU generation

Mandatory to implement MPPDU generation capability is specified for PrY operation (Clause 20). While not specifying desired behavior, the following rules constrain and permit aspects of the valid generation of MPPDUs:

a) All MPPDUs shall include the MAC Privacy protection EtherType as the first two octets.
b) An MPPDU shall only contain MPPDU components specified in this standard.
c) Any Encapsulated Frame or Frame Fragment component included in the MPPDU shall be present in its entirety as indicated by an accurate *following length*.
d) An Explicit Pad component can be encoded with a *following length* that exceeds the number of octets remaining in the MPPDU.
e) All pad octets in a Trailing Pad or an Explicit Pad shall have the value zero.

   NOTE 1—This requirement guards against the accidental inclusion of data in an MPPDU, even if that data was previously transmitted to the same PrY(s). It is not validated on receipt (19.7), but is required for conformance.

f) A user data frame that is fragmented for transmission shall be encoded in a sequence of Frame Fragments having the same value of the E bit and consecutively numbered fragment sequence numbers, with the I bit set in the first Frame Fragments and the F bit set in the last.

g) Any type of MPPDU component can be present multiple times in a MPPDU (subject to the size of the MPPDU), and the order of individual components is not constrained with the following exceptions:
   1) Any Trailing Pad is the last component in the MPPDU. A receiver does not extract any further components or data from the MPPDU once the initial octets of a Trailing Pad are encountered.
   2) If more than one Frame Fragment with a given value of the E bit is present in an MPPDU, those components are encoded in fragment sequence number order.

   NOTE 2—A solitary Explicit Pad can provide padding that is as short as two octets (including the MPPCI). This permits, for example, the transmission of Encapsulated Frames starting on specific octet boundaries within an MPPDU for a best fit with stream gates (IEEE Std 802.1Q) with some variation in user data frame size. An immediately following Explicit Pad can occupy just those MPPDU octets that could have been taken by an Encapsulated Frame if a user data frame had been available for transmission. User data frames originating from stations attached to some media can be shorter than those sent by stations using IEEE Std 802.3, and do not have be padded before being encapsulated in an MPPDU.

The management counters specified by this standard for MPPDU transmission (20.14.1) reflect the mandatory to implement support of unprotected, Privacy Frame, or Privacy Channel transmission (17.4, 20.6–20.10). Separate counters are specified for Privacy Frames and for each Privacy Channel. If additional MPPDU generation and user data frame encapsulation algorithms are implemented, their configuration should be supported by appropriate performance monitoring.

19.7 MPPDU validation

A PrY recognizes a frame that is addressed to that PrY as an MPPDU if the first two octets of the frame's MSDU compose the MAC Privacy protection EtherType.

NOTE 1—A frame that is addressed to a PrY has a MAC Destination Address that is the individual address associated with the interface stack of which the PrY is part or is a Group Address that the PrY uses to receive frames (Clause 20). A PrY also verifies the MAC Source Address (SA) of each MPPDU. MPPDUs received with SAs that a PrY is not configured to receive are discarded and not validated or otherwise processed (Clause 20).

Each of the following possible MPPDU components is validated in the order they are encoded in the MPPDU, and the management counters specified in 20.14.2 updated, as follows:

a) If no octets remain to be processed, the validation of the MPPDU is complete.

b) If only one octet remains to be processed, InPadOctets is incremented, the octet is discarded and validation of the MPPDU is complete.

   NOTE 2—The value of the octet is not checked. If the MPPDU was correctly encoded it is zero, identifying the octet as a Trailing Pad.

c) If the component is incorrectly encoded (19.5.6) the octets of that component and any octets following in the MPPDU are discarded, inErroredMppdus is incremented, and validation of the MPPDU is terminated.

d) If the component is unrecognized (19.5.5) inUnknownMppcis is incremented, and the component (including any octets within the *following length* and in the MPPDU) is discarded.

e) If the component is an Encapsulated User Data Frame (19.5.1), inEncapsulatedFrames is incremented, the number of octets indicated by the following length is added to InUserOctets, and the component is extracted.

   NOTE 3—Case c) or d) above applies if the component has bits 8 and 7 of the first MPPCI octet both zero but has a *following length* that is too short for an encapsulated frame or too long for the MPPDU.

f) If component is a Trailing Pad (19.5.2), the number of pad octets (the Trailing Pad MPPCI and the remaining octets in the MPPDU) is added to InPadOctets and validation of the MPPDU is complete.

g) If the component is an Explicit Pad (19.5.3), the length of the Explicit Pad plus the number of octets following in that component (the *following length* or the number remaining in the MPPDU if less) is added to InPadOctets and the component is discarded.

h) If the component is a Frame Fragment (19.5.4), inExpressFragments (if the E bit is set) or inPreemptableFragments (if the E bit is clear) is incremented, the number of octets indicated by the following length is added to inUserOctets, and the component is extracted.

NOTE 4—The InUserOctets counter is incremented (even though the fragment can be discarded due to the loss of other fragments of the original user data frame) to allow link usage to be assessed by comparing InUserOctets and InPadOctets.

Those components that are unrecognized or recognized as padding are discarded, i.e., identified as requiring no further processing. Encapsulated Frame and Frame Fragment components are specified as being extracted, i.e., identified as requiring further processing as specified in Clause 18 and Clause 20. Components from each peer PrY that are extracted are retained in the order received, until processed. The model of PrY operation (Clause 20) is consistent with that processing taking place immediately after each component is validated and before components that follow in the same MPPDU are extracted, or with processing after components from several received MPPDUs have been extracted. If the processing of an MPPDU is terminated during validation due to an error in the encoding of a component it is unspecified as to whether earlier components encoded in that MPPDU are retained for further processing or not.

NOTE 5—The validation process makes no assumptions about the validity of unrecognized components, though any component with a following length that exceeds the number of octets remaining in the MPPDU is identified as the last component in the MPPDU.

The MPPDU reception and validation process, and the associated management counters specified in 20.14.2, are independent of the MPPDU transmission algorithms used by peer PrYs.

NOTE 6—A system can derive octet counts and counts of the number of frames received from other counters maintained for the interface stack, where those are suitable for this purpose. As specified in 20.2 conformant implementations exhibit the externally observable behavior specified by this standard.

This standard specifies constraints on the sizes of MPPDUs and of individual MPPDU components or combination of those components that can be generated (19.6). If any of those constraints are violated for a given MPPDU, a receiving PrY can terminate, but is not required to terminate, validation of that MPPDU. Terminating the validation of any given MPPDU shall not affect the processing of other MPPDUs.

Insert the following text (Clause 20) after Clause 19:

20. MAC Privacy protection Entity (PrY) operation

This clause

a) Provides an overview of the MAC Privacy protection Entity (PrY) and the service it provides (20.1).
b) Provides a model of operation (20.2) comprising an architecture (20.3) and its constituent processes that supports the PrY's functionality including management (20.14).

Clause 17 provides an overview of MAC Privacy protection and an introduction to PrY functionality. Clause 18 describes MAC Privacy protection protocol design, support, and addressing requirements, and Clause 19 describes the encoding of the MAC Privacy protection Protocol Units (MPPDUs) transmitted and received by PrYs. Clause 21 describes the position of PrYs in the interface stacks of particular systems.

20.1 PrY overview

A PrY is a shim (see IEEE Std 802.1AC) that transmits and receives MPPDUs conveying user data frames to and from peer PrYs in other systems. Each PrY is supported by a SecY (Clause 10) that confidentiality protects the MPPDUs transmitted by the PrY. That SecY is typically positioned immediately below the PrY in an interface stack (Figure 20-1, 21.1), but can be located in a separate system (21.8).

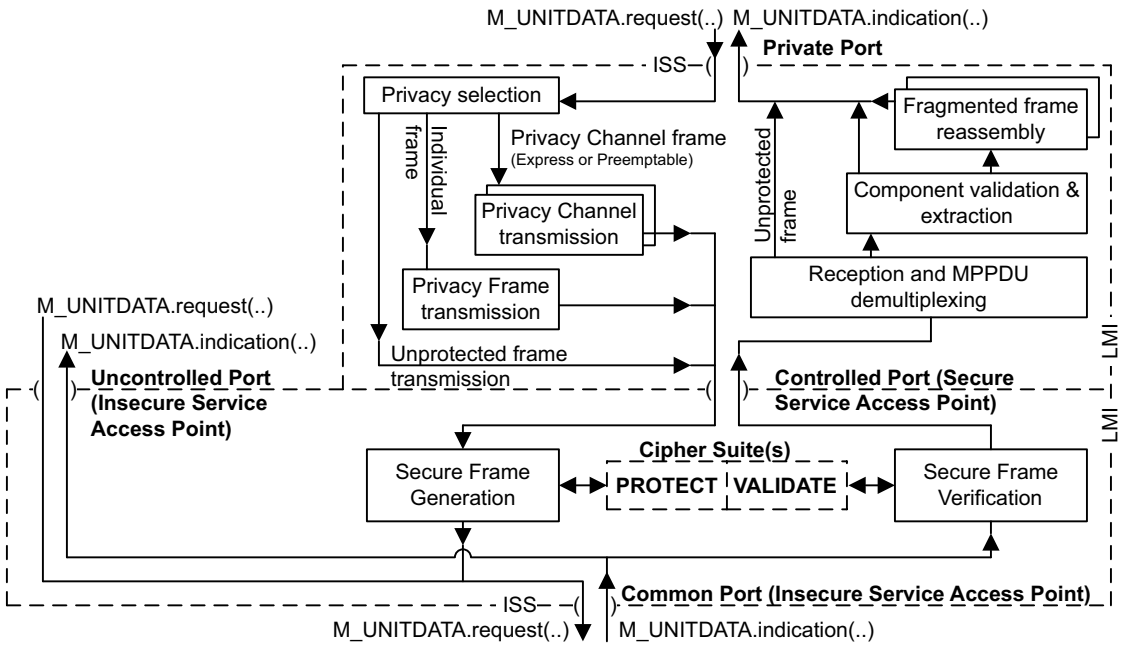


Figure 20-1—PrY and SecY

Each PrY provides the MAC Internal Sublayer Service (ISS, IEEE Std 802.1AC) at its Private Port which accepts user data frames for transmission and delivers received user data frames. The PrY uses an instance of the ISS at its Controlled Port to transmit and receive both MPPDUs that encapsulate those user data frames and user data frames that are not privacy protected. When the PrY is directly supported by a SecY, that Controlled Port is the SecY's Controlled Port, as shown in Figure 20-1. If the PrY is in a separate system or interface stack from its associated SecY, a physically secure data path between their respective Controlled Ports is required (21.8).

NOTE—Throughout this clause (Clause 20) the term *user* refers to the user of the PrY's Private Port.

20.2 Model of operation

The model of operation in this clause is simply a basis for describing the functionality of a PrY. It is not intended to overconstrain real implementations; these may adopt any internal model of operation compatible with the externally visible behavior that this standard specifies. Conformance of equipment to this standard is purely in respect of observable protocol.

NOTE—PrYs use and complement the QoS mechanisms specified for bridges in IEEE Std 802.1Q.

20.3 PrY architecture

The PrY architecture is illustrated in Figure 20-2.

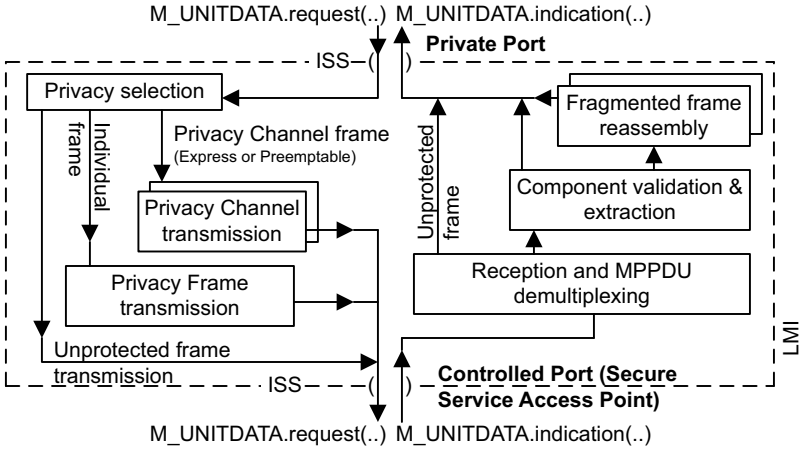


Figure 20-2—PrY architecture

The PrY's transmit and receive data paths are independent of one another, but both are supported by:

a) The PrY's Private Port and Controlled Port and their MAC status (6.4) and point-to-point (6.5) parameters (20.4).

b) The PrY's Management Entity (20.14) and Layer Management Interface (LMI).

The PrY's transmit data path is supported by the following:

c) A Privacy Selection process that uses a manageable Privacy Selection Table to determine if, and how, user data frames are protected (20.5).

d) Transmission of selected user data frames without privacy protection (20.6).

e) An individual frame transmission function that can transmit user data frames as individual Privacy Frames i.e., encapsulated in an MPPDU that conveys only that user data frame in an MPPDU, padded as specified by the Privacy Selection Table (20.7).

f) A Privacy Channel transmission process that can encapsulate multiple user data frames within a single MPPDU, can fragment individual user data frames so they are conveyed in more than one MPPDU, and can pad those MPPDUs to a fixed size (20.8).

The PrY's receive data path is supported by the following:

g) A reception process that recognizes and demultiplexes MPPDUs addressed to this PrY (20.11) for validation and component extraction, submitting other received frames directly to the PrY's user.

h) MPPDU validation (20.12), removing padding and unrecognized components and extracting Encapsulated Frames (19.5.1) and Frame Fragments (19.5.4) for reassembly.

i) A user data frame Reassembly process (20.13).

20.4 MAC status and point-to-point parameters

The MAC status parameters [MAC_Enabled and MAC_Operational, (6.4)] and point-to-point parameters [operPointToPointMAC and adminPointToPointMAC, (6.5)] for the Private Port are not independently controlled. Their values are the same as those for the PrY's Controlled Port, and are determined by the protocol entity supporting that port.

20.5 Privacy Selection

The PrY's user requests transmission of a single user data frame by issuing an ISS transmit request at the PrY's Private Port. The PrY uses the priority that the user has associated with that request, in conjunction with a manageable Privacy Selection Table (17.4, 17.4.3), to select one of the following ways of transmitting that user data frame through the Controlled Port:

a) Without privacy protection (20.6).
b) As an individual Privacy Frame (20.7, 17.4.1), i.e., in an MPPDU that encapsulates only that data frame (with the possible addition of padding).
c) In a Privacy Channel (20.8, 17.3.6) as an Express frame or as a Preemptable frame, with the possibility of fragmentation into successive MPPDUs.

The encapsulation and transmission of user data frames in MPPDUs can also be enabled and disabled for the PrY as a whole. If MPPDU encapsulation and transmission is enabled, the Privacy Selection Table is used as specified in a) through c) above. If disabled, all frames are transmitted without MAC Privacy protection (20.6). The Privacy Selection Table can be configured before MPPDU encapsulation is enabled. MPPDU encapsulation can also be disabled, at the risk of privacy information exposure, to support transparent operation and troubleshooting. MPPDU reception and demultiplexing (20.11) can be independently enabled and disabled, so a pair of communicating PrYs can be configured to receive MPPDUs before either PrY transmits MPPDUs.

20.5.1 Accepting user data frames for transmission

A PrY provides minimal buffering for transmitted user data frames, sufficient only to allow user data frames to be encoded in a Privacy Channel MPPDU awaiting transmission, and any remaining frame fragment to be retained for encapsulation in a following MPPDU for its selected Privacy Channel. A transmit request is only accepted if one of the following applies:

a) The user data frame is to be transmitted as an individual frame (without privacy protection or as a Privacy Frame), the immediately prior user data frame has been transmitted through the Controlled Port, and transmission of a previously generated Privacy Channel MPPDU (20.9) is not pending.
b) The user data frame is to be conveyed by a Privacy Channel, an MPPDU has been generated and not transmitted for that Privacy Channel, and the user data frame (or a fragment of that frame) can be encoded in that MPPDU.

A PrY's Private Port can accept a transmit request for a user data frame for transmission without privacy protection or as an individual Privacy Frame even if the effect of accepting that request is to delay the generation of a Privacy Channel MPPDU (see 20.9.4).

NOTE 1—The use of minimal PrY buffering allows the Transmission Selection processes specified for end stations and bridge (see IEEE Std 802.1Q) to retain control over the use of transmission bandwidth, so far as efficient use of Privacy Channel bandwidth permits. Conditional per-user priority acceptance of user data frames for transmission is similar to the use of Priority-based Flow Control (see IEEE Std 802.1Q), but without external dependencies.

NOTE 2—An ISS transmit request occurs at a service access point when a service user (in this case the user of a PrY's Private Port) wishes to transmit a frame and the service provider (in this case a PrY) can accept that request. The ISS transmit request and receive indication parameters describe the information conveyed between peer service users each time the service is used, and exclude information used only within one of the communicating systems

(IEEE Std 802.1AC). While a transmit request is commonly described as if it were an action taken unilaterally by the service user, it describes a system's externally observable behavior. If a service provider within a system was unable or unwilling to accept a transmit request, the request did not occur.

20.5.2 Stream-based Privacy Selection

A PrY may, in addition to supporting user priority based privacy selection via the Privacy Selection Table as specified in this standard, allow use of the stream identification function specified in IEEE Std 802.1CB.

NOTE 1—Configuration options for use with stream identification are not specified by this standard. However, where a Bridge supports both per-stream classification and metering (see IEEE Std 802.1Q) and MAC Privacy protection, it is not necessary for each PrY to duplicate the identification process. The stream_handle can be passed as a sub-parameter of the transmit request (see IEEE Std 802.1CB).

NOTE 2—While simple priority selection of MPPDU encapsulation (or not) is a coarse control, it can distinguish traffic for which the octet overhead of a Privacy Frame or the delay and jitter that can result from use of a Privacy Channel protection might be undesirable, e.g., priority 1 is commonly used for TSN streams with network bandwidth allocation.

20.6 Unprotected frame transmission

When the PrY's Private Port accepts a transmit request for a frame that is not to be privacy protected by encapsulation in an MPPDU, a corresponding transmit request is used to pass that unmodified user data frame to the service supporting the Controlled Port before any other frame transmitted as the result of accepting a later Private Port transmit request. The priority used to access the underlying service (the access priority) can differ from the priority requested by the PrY's user and is determined by the Privacy Selection Table.

NOTE 1—Ordering preservation for individually transmitted frames does not preclude support of preemption (e.g., as specified by IEEE Std 802.3 for Interspersing Express Traffic) as the underlying service (supporting the Controlled Port) can accept a transmit request for an express frame while physical transmission of an earlier preemptable request is still in progress.

NOTE 2—Use of a Privacy Selection Table determined priority to access an underlying service does not result in any change to user data. It does not change the value of any priority parameter encoded in that data, e.g., in a VLAN tag. The priority used to access the underlying service can be used to determine values in any tag added by that service.

20.7 Privacy Frame transmission

When the PrY's Private Port accepts a transmit request for a user data frame that is to be protected as an individual Privacy Frame, it is encoded as an MPPDU Encapsulated Frame (19.5.1).

The Privacy Selection Table is used to determine the MPPDU's access priority, whether padding is to be added to the MPPDU, and whether the user data frame's transmit request's drop_eligible parameter is hidden, i.e., always set False when making the Controlled Port transmit request for the MPPDU.

NOTE—Different values of PrY access priority are only relevant if they select differentiated services from a supporting SecY (between two transmit SCs, for example) or from the media access method or other protocol entity that supports the SecY's Controlled Port (providing frame preemption, for example). Similarly the drop_eligible parameter is only relevant if it is used by the underlying service. In a simple end station or single component bridge, it is not encoded with the Privacy Frame (any VLAN tag is encoded by support for the EISS, higher in the interface stack than a PrY or SecY which both support the ISS). However the network component of an EDE-CC (for example) can receive the internally communicated drop_eligible parameter for use in its subsequent Provider Network Port VLAN tag encoding.

If the privacyPadding parameter is non-zero, a Trailing Pad (19.5.2) is added (if required) to increase the size of the MPPDU (excluding its source and destination MAC addresses) to four octets (to allow for the MAC Privacy protection EtherType and MPPCI) plus a multiple of 16, 32, or 64 octets. If the revealDE parameter is Visible (True) the user data frame's transmit request's drop_eligible parameter is passed unchanged to the Controlled Port.

20.8 Privacy Channel transmission

The Privacy Channel transmission process supports two Privacy Channels, an Express Channel and a Preemptable Channel. A PrY can be configured to use both, one, or neither of these channels. Each channel is supported by its own instance of each of the following processes:

a) Privacy Channel MPPDU Generation (20.9)
b) Privacy Channel Encapsulation (20.10)

20.9 Privacy Channel MPPDU Generation

20.9.1 Objectives

Each instance of the Privacy Channel MPPDU generation process provides its channel's Privacy Channel MPPDU Encapsulation process with notionally empty MPPDUs that the latter can use to encapsulate user data frames. The objectives of the generation process(es) are to:

a) Initiate transmission of MPPDUs for the Privacy Channel at intervals that are independent of the user data frames and padding encoded in each of those MPPDUs.
b) Allow Privacy Channel MPPDUs to be padded to fixed sizes.
c) Support transmission scheduling decisions made by the processes that use the PrY's interface stack, e.g., transmission selection as specified for bridges by IEEE Std 802.1Q.
d) Minimize the impact of Privacy Channel encapsulation on total bandwidth and on the transit delay experienced by individual data flows.

Any particular Privacy Channel MPPDU generation algorithm necessarily reflects a balance between these objectives. This standard specifies a mandatory to implement default algorithm (20.9.4), suitable for use with the strict priority, credit-based shaper, enhanced transmission selection (ETS), and ATS transmission selection algorithms specified in IEEE Std 802.1Q. An optional algorithm, for use with the IEEE Std 802.1Q enhancements for scheduled traffic and the use of transmission gates is also specified (20.9.5).

20.9.2 Timeliness

Privacy Channel MPPDUs should be generated no earlier than required to provide the Controlled Port with transmit data at the maximum rate that port can support. Earlier Privacy Channel MPPDU generation can defeat the intended operation of the Controlled Port's user's transmission selection algorithm, by delaying a high priority frame that becomes available for transmission just prior to MPPDU transmission.

20.9.3 Competition for transmission bandwidth

Transmission of MPPDUs used to support a particular Privacy Channel can compete for transmission bandwidth with user data frames that are not privacy protected, are transmitted as individual Privacy Frames, or that are transmitted to support a second Privacy Channel. The SecY that supports a PrY's Controlled Port can also transmit frames sent by the users of its Uncontrolled Port, temporarily denying bandwidth to its Controlled Port and the PrY. All the above frames, including those conveying MPPDUs, can also be subject to later traffic shaping and consequent delay, e.g., by the Provider Network Port of an EDE-CC.

Provided that these delays and changes to Privacy Channel MPPDU transmission timing are independent of the content of Privacy Channel MPPDUs, they do not defeat the protection provided to user data frames conveyed by the Privacy Channel, except by signaling that not all traffic is carried by that channel. In particular traffic shaping and other frame timing changes can be made by the network component of an EDE-CC, EDE-SS, or EDE-CS.

The Privacy Channel MPPDU generation algorithms described in this standard take account of the possible bandwidth competition and the desire to provide more or less constant bandwidth long term while avoiding bursts in channel transmission. Additional algorithms may be implemented, subject to documentation of the way that they meet the objectives described above (20.9.1) and of their additional objectives.

20.9.4 Default Privacy Channel MPPDU generation algorithm

By default, all the MPPDUs generated for a given Privacy Channel are of the same size. To simplify management, the configurable userDataFrameSize parameter is the size of the largest user data frame that MPPDU could convey, i.e., the number of octets in the user data frame's MAC Destination Address (DA), MAC Source Address (SA), MSDU, and 4-octet FCS (17.4.2). The userDataFrameSize, is used by the PrY to calculate a channelFrameSize in bits. The channelFrameSize calculation takes into account the 12 octets of the source and destination MAC addresses of the MPPDU, the 2 octets of the MAC Privacy protection EtherType, the 2 octets of the MPPCI used to encode an EncapsulatedFrame (19.5.1), the omission of the 4 octets of the user data frame FCS, and the frameTransmissionOverhead (in octets):

$$\text{channelFrameSize} = 8 \times (\text{userDataFrameSize} + 12 + \text{frameTransmissionOverhead})$$

The frameTransmissionOverhead includes the octets of SecTAG and ICV (9.2) added by MACsec (if the PrY's Controlled Port is supported by a SecY) and any other media independent or media dependent overhead added by interface stack components (e.g., frame preamble).

NOTE 1—The userDataFrameSize is 1522 octets for a maximum sized Ethernet frame with a single VLAN tag. Since the user data frame's FCS is not encoded in the MPPDU, transmission of such a frame will add only 1518 octets to the management count chOutUserOctets (20.14.1).

NOTE 2—See 3.2.7 and 1.4.245 of IEEE Std 802.3-2022 for applicable maximum frame sizes when Ethernet is used in conjunction with encapsulation protocols including MAC Security.

Privacy Channel MPPDUs are generated by a simple token bucket algorithm. The notional bucket contains a number of bits. When a Privacy Channel MPPDU is transmitted, channelFrameSize bits are subtracted from the bucket. Bits are added to the bucket at the manageable requestedKbitRate, expressed in bits per second, up to a maximum of channelBurstSize bits, calculated by the PrY from the manageable parameter userBurstOctets as:

$$\text{channelBurstSize} = \text{channelFrameSize} \times (1 + (\text{userBurstOctets})/(\text{userDataFrameSize}))$$

If a single Privacy Channel is configured for transmission, a channel MPPDU is generated for use by the Privacy Channel's MPPDU Encapsulation process when the following conditions are satisfied:

a) The Privacy Channel's token bucket contains at least channelFrameSize bits when the MPPDU is transmitted, and

b) The user of the Private Port has:
   1) no user data frame eligible for immediate transmission, or
   2) the highest access priority for a user data frame eligible for immediate transmission is not numerically higher than the access priority used by the Privacy Channel.

c) The timeliness condition (20.9.2) is satisfied, i.e., any later generation of the MPPDU could result in its later transmission on the medium supporting the PrY's interface stack.

If both an Express and a Preemptable Privacy Channel are configured, and conditions a) through c) in this subclause (20.9.4) are met for both Privacy Channels, a Channel MPPDU is only generated for the Express Privacy Channel.

NOTE 3—If delays in the Privacy Channel's MPPDU Encapsulation process, in the operation of a supporting SecY, and in the latter's use of the transmission medium are ignored, condition c) is equivalent to specifying that the transmission of the immediately prior MPPDU or any competing frame has been completed.

20.9.5 Transmission gate MPPDU generation algorithm

A PrY that is part of an interface stack that supports the use of transmission gates and a gate control list (IEEE Std 802.1Q) may be configured to use the Privacy Channel MPPDU generation algorithm specified in this subclause (20.9.5).

20.9.5.1 Constraints

Successful operation of the algorithm constrains the assignment of user data frames to traffic class queues, the configuration of the gate control list (IEEE Std 802.1Q), and the configuration of the PrY's Privacy Selection Table as follows:

a) All frames queued for a given traffic class and thus subject to the operation of the same transmission gate are either all transmitted in a given Privacy Channel or are transmitted as individual frames.

b) The per traffic class transmission gates open at any given time are either all for user data frame transmission in a given Privacy Channel or all for transmission as individual frames.

c) Following execution of each entry in the gate control list that specifies an open state for a traffic class with Privacy Channel transmission, sufficient time elapses prior to the execution of the next entry in the list for transmission of an MPPDU containing:
   1) A Frame Fragment, if fragmentation is enabled for that channel, or
   2) An Encapsulated Frame of the maximum size that can be assigned to that traffic class.

20.9.5.2 Algorithm

A Privacy Channel MPPDU, for a given Privacy Channel, is transmitted when either of the following conditions apply:

a) An entry in the gate control list is executed and specifies that a transmission gate is open for a traffic class whose user data frames are encapsulated in that channel.

b) The transmission gate is open for such a traffic class, one or more prior MPPDUs have been generated as permitted by these conditions [a) and b)], generation and transmission of the MPPDU would not delay the subsequent transmission of an MPPDU or user data frame generated [see a)] or permitted by the execution of a gate control list entry, and later generation of the MPPDU could delay its transmission.

The size of each generated MPPDU is the maximum possible without delaying subsequent transmissions [see b)] subject to an upper bound capable of conveying an unfragmented user data frame of userDataFrameSize octets, without any pad.

NOTE 1—If possible delays in the Privacy Channel's MPPDU Encapsulation process, in the operation of a supporting SecY, and in the latter's use of the transmission medium are ignored, condition b) is equivalent to the following: transmission of any prior MPPDU or competing frame has been completed, and the transmission of this MPPDU will be complete prior to execution of the next gate control list entry.

NOTE 2—The conditions [a) and b)] apply even if execution of a gate control list entry does not change the state of a transmission gate. The corresponding reduction in the MPPDU size (for specified control list entry execution intervals) can facilitate earlier reception and validation of the MACsec ICV protecting those MPPDUs and thus reduce the effective transit delay of the encapsulated user data frames.

20.10 Privacy Channel Encapsulation

Once a Privacy Channel MPPDU has been generated and prior to its transmission through the PrY's Controlled Port, the PrY's Private Port and the encapsulation process accepts (and only accepts) transmit requests for user data frames that are to be conveyed by that Privacy Channel. The delay (if any) between the generation of an MPPDU and its subsequent transmission, following a transmit request to the Controlled

Port, should be independent of the content (user data frames or padding) encapsulated in the MPPDU, or of any similar operation on any other port in the system of which the PrY is a part, in order to avoid disclosing the level of activity to an adversary.

A Privacy Channel encapsulation algorithm can fragment user data frames to make better use of Privacy Channel bandwidth. The Privacy Selection Table classifies each user data frame that can be conveyed by a Privacy Channel as an Express frame or as a Preemptable frame, and each encapsulated Frame Fragment is identified as an Express or a Preemptable frame fragment. After a Frame Fragment has been encoded in an MPPDU (19.5.4), the encapsulation process can hold the remainder of a single fragmented Express frame or of a single fragmented Preemptable frame (or the remainder of one user data frame of each classification, if a single Privacy Channel is configured) for later encoding in the same or a subsequent MPPDU.

Each Privacy Channel shall be capable of being configured to use the default encapsulation algorithm (20.10.1) and may be configured to support additional encapsulation algorithms as specified in (18.5). Fragmentation is expected to be used in most network scenarios but can be disabled on transmission, per Privacy Channel. A PrY is always capable of receiving and reassembling fragmented user data frames. Enabling or disabling transmit fragmentation can result in a temporary loss of connectivity with the PrY's Private Port's MAC_Operational parameter value temporarily false.

20.10.1 Default Privacy Channel encapsulation algorithm

The default encapsulation algorithm ensures that no fragment comprises less than 64 octets of a user data frame. A frame that can be fragmented (or the remainder of a frame than can be subject to further fragmentation) comprises at least 128 octets and its fragmentation removes the greatest multiple of 64 octets that can be encoded from its initial octets, leaving a remainder of at least 64 octets.

NOTE 1—An entire frame can comprise fewer than 64 octets, either as a result of its initial transmission using a medium access method other than that specified by IEEE Std 802.3 or after the removal of one or more tags (e.g., a VLAN tag).

NOTE 2—The minimum userDataFrameSize (20.9.4) that can be used with this algorithm is 132 octets, unless the maximum size of frames to be conveyed by the channel is restricted to be less than 130 octets.

When a new MPPDU is generated for a Privacy Channel that conveys both Express and Preemptable frames, user data frames (or fragments of those frames) are added in the following priority order:

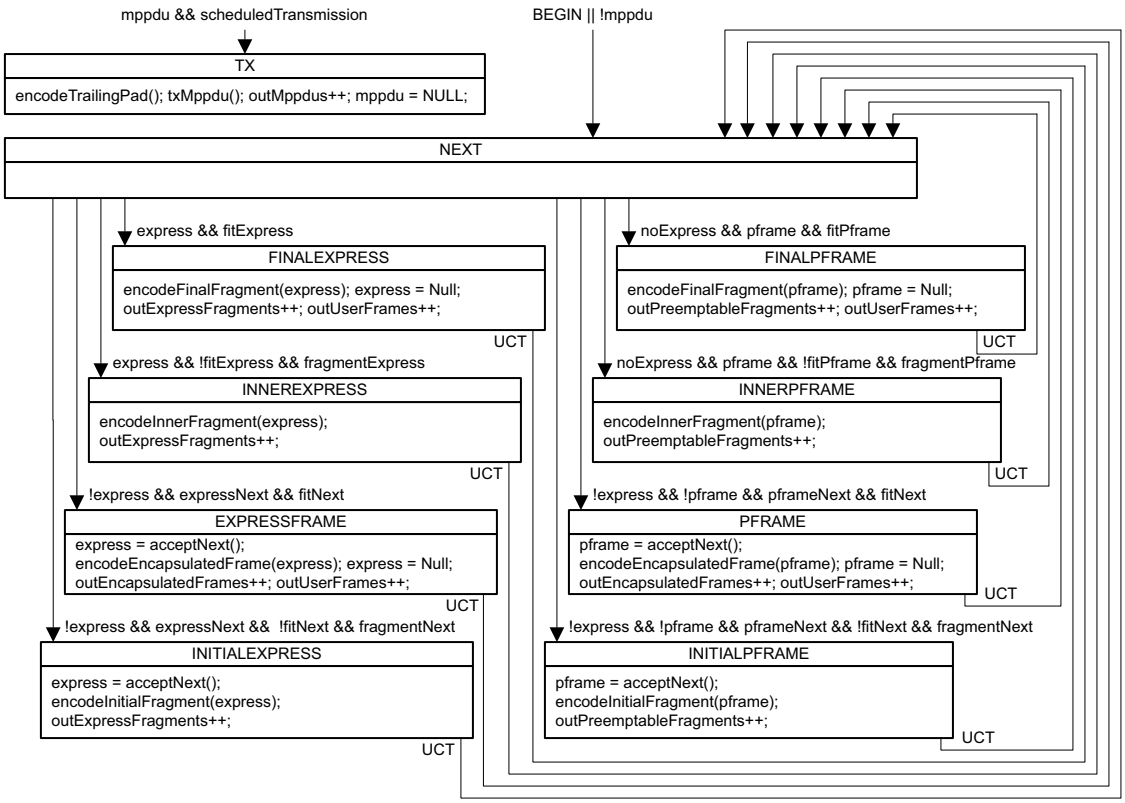
a) The remaining octets (if any) of a previously fragmented Express Frame.
b) The highest priority user data frame (if any) that the PrY's user wishes to transmit that is identified (by the Privacy selection process, 20.5) as an Express frame.
c) The remaining octets (if any) of a previously fragmented Preemptable Frame.
d) The highest priority user data frame (if any) that the PrY's user wishes to transmit that is identified (by the Privacy selection process, 20.5) as a Preemptable frame.

Once encapsulation is complete, the value zero is encoded in each of the remaining octets of the MPPDU, and the MPPDU is transmitted.

Figure 20-3 specifies the default encapsulation algorithm using the state machine conventions used in this standard and takes precedence if there is any discrepancy between it and the text of this subclause.

If a single Privacy Channel is configured, the encoding of an Express frame (or its fragments) can interrupt the encoding of a series of fragments of a default class frame (19.5.4). If two Privacy Channels are configured, conditions c) and d) in this subclause do not apply to the Express Privacy Channel and conditions a) and b) do not apply to the Preemptable Privacy Channel.

NOTE 3—When fixed sized Privacy Channel MPPDUs are to be sent at constant rate, the use of a single Privacy Channel can reduce the bandwidth required. Interrupting the encoding of a Preemptable frame to encode an Express frame can reduce the delay experienced by the latter if that allows its encoding to be completed in an earlier MPPDU. In contrast, the use of both Privacy Channels allows Express user data frames to use a higher access priority and take advantage of any preemption capabilities provided by the underlying service that supports the Controlled Port.



State machine conditions, each True (not Null) iff (if and only if):

mppdu : An MPPDU has been generated and not yet transmitted.
express : The PrY is holding the remainder or all of an Express user data frame.
expressNext: The PrY's user has selected, and has available, an Express user data frame as the next frame for transmission.
noExpress: express and expressNext are both False.
pframe : The PrY is holding the remainder or all of a Preemptable user data frame.
pframeNext : The PrY's user has selected, and has available, a Preemptable user data frame as the next frame for transmission.
fitExpress : The Express frame (or all of the remainder of the fragmented Express frame) can be encoded in the remaining MPPDU octets.
fragmentExpress : The Express frame or its remainder can be fragmented, and the next fragment encoded in the remaining MPPDU octets.
fitPframe : The Preemptable frame (or all of the remainder of the fragmented Preemptable frame) can be encoded in the remaining MPPDU octets.
fragmentPframe : The Preemptable frame remainder can be fragmented, and the next fragment encoded in the remaining MPPDU octets.
fitNext : The next user data frame (Express or Preemptable) can be encoded in the remaining octets of the MPPDU without fragmentation.
fragmentNext : The next user data frame can be fragmented, and the first fragment encoded in the remaining MPPDU octets.
scheduledTransmission: The MPPDU will be eligible for transmission once the Trailing Pad, if any has been added.
Note: The 'open' transition to the TX state takes precedence over other transitions from NEXT, apart from BEGIN.

State machine procedures:

express = acceptNext() : Accept the next user data frame (an Express frame) for transmission, similarly frame = acceptNext() for a Preemptable frame.

encodeEncapsulatedFrame(express), encodeEncapsulatedFrame(pframe) : Encode the user data frame in the MPPDU, and add the number of user data octets encoded (not including the MPPCI) to outUserOctets.

encodeInitialFragment(express), encodeInitialFragment(pframe) : Encode an Initial Fragment, encapsulating the greatest multiple of 64 octets from the user data frame that will fit in the MPPDU leaving at least 64 octets of the user data frame as a remainder, and add the number of user data frame octets encoded (not including the MPPCI) to outUserOctets.

encodeInnerFragment(express), encodeInnerFragment(pframe) : Encode a Frame Fragment (with Initial and Final bits clear), encapsulating the greatest multiple of 64 octets that will fit in the MPPDU leaving at least 64 octets of the frame as a remainder, and add the number of user data frame octets encoded (not including the MPPCI) to outUserOctets.

encodeFinalFragment(express),encodeFinalFragment(pframe) : Encode the remainder of the user data frame in a Final Fragment.

encodeTrailing Pad() : Encode the value 0 in all the remaining octets (if any) of the MPPDU, add the number of pad octets to outPadOctets. Changes in the number of octets to be added shall not affect time at which the MPPDU is transmitted.

txMppdu() : Transmit (complete the transmission of) the MPPDU through the PrY's Controlled Port.

Figure 20-3—Privacy Channel Encapsulation state machine

An implementation of the default encapsulation algorithm (Figure 20-3) can complete MPPDU encoding prior to transmission or can begin transmission while later octets of the MPPDU are still being encoded. An MPPDU component need only be available for encoding when that encoding takes place.

20.11 MPPDU reception and demultiplexing

The underlying service (typically that provided by a SecY) that supports a PrY's Controlled Port issues an ISS service indication to signal receipt of a frame. The frame is recognized as an MPPDU addressed to this PrY if, and only if both:

a) The destination MAC Address of the frame is:
   1) The address used by the PrY as the destination MAC address of transmitted MPPDUs (ifMppduDA, 22.6), and that address is a group MAC address; or
   2) The individual MAC Address used by the PrY as the source MAC address of transmitted MPPDUs (ifAddr, 22.6).

and

b) The first two data octets encode the MAC Privacy protection EtherType (19.3, Table 19-1).

MPPDUs addressed to this PrY are processed by the MPPDU component validation and extraction logic, unless MPPDU reception has been disabled by management, in which case they are discarded.

Otherwise receipt of the frame results in a corresponding ISS service indication at the PrY's Private Port to signal receipt of that (unmodified) frame to the PrY's user.

When the PrY's Controlled Port is directly supported by a SecY, the destination MAC address of transmitted MPPDUs is the group MAC Address used by the supporting PAE's KaY (IEEE Std 802.1X) to discover peer SecYs and distribute SAKs. SecY support of the PrY is reported by ifSecySupport (22.6).

NOTE 1—The PrY does not recognize or otherwise process received frames whose initial octets are those of a VLAN tag or any other type of tag. Similarly a PrY does not add VLAN tags to transmitted MPPDUs. VLAN tags can be added or removed by other protocol components in the system of which the PrY is part, e.g., an EDE.

Received frames that are not recognized as MPPDUs addressed to this PrY are delivered to the Private Port in the order that they are received from the Controlled Port. Their delivery is not supressed if MPPDU reception and demultiplexing has been disabled for this PrY. The relative order of the delivery of those frames and of frames decapsulated from received MPPDUs is not specified.

NOTE 2—An interface stack containing a PrY for which MPPDU reception has been disabled can be on the path between another pair of communicating PrYs. Its presence does not disrupt that communication unless they are using a destination address associated with the disabled PrY, or filtered by the system of which the PrY is a part.

20.12 MPPDU component validation and extraction

MPPDUs received from the Controlled Port and addressed to the PrY are processed in the order received. Each MPPDU is processed, and its components extracted and validated, as specified in (19.7). Encapsulated Frames (19.5.1) and Frame Fragments (19.5.4) are both delivered to the reassembly process in the order they were encoded in each received MPPDU and in the order those MPPDUs are received.

20.13 Protected frame reception and reassembly

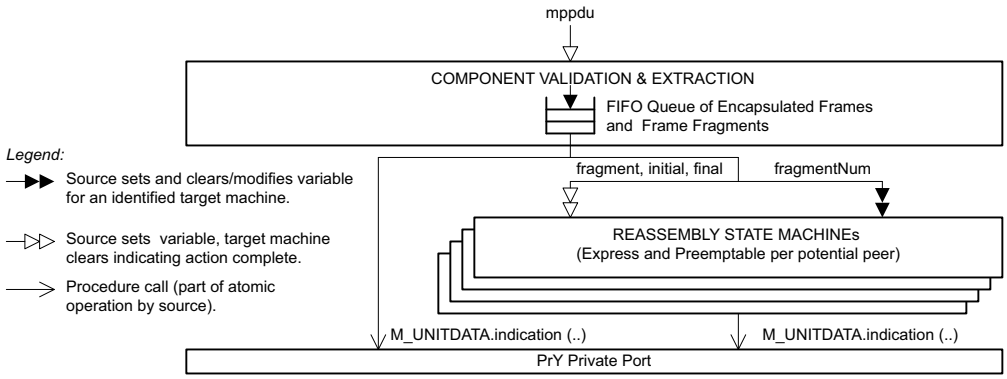
Each user data frame is delivered to the user of the PrY's Private Port when its reception by the PrY (in an Encapsulated Frame or in a set of Frame Fragments) is complete, and is delivered before any other user data frame whose complete reception depends on an Encapsulated Frame or a Frame Fragment encoded later in the same MPPDU or in a subsequently received MPPDU.

Delivery of a complete user data frame is not postponed pending the (possible) arrival of other Frame Fragments or Encapsulated Frames.

A PrY shall be capable of the simultaneous reassembly of one Express frame and one Preemptable frame. A PrY may be capable of reassembling fragmented frames from more than one peer PrY, and shall then support simultaneous reassembly of one Express and one Preemptable frame for each of a specified maximum number of peers (ifMaxPeers, 22.6). If the PrY is supported by a SecY, ifSecYSupport (22.6) is true and MKA updates the actual number (ifNumPeers, 22.6) and their source MAC addresses (peers, 22.6), avoiding the need for management configuration of that information.

Any given frame shall be reassembled from fragments received in MPPDUs with same, verified, source MAC address, unless MKA has confirmed that there is a single peer. If fragments are received from more than the supported number of peers, all, some, or none of the frames encoded in a sequence of fragments are successfully reassembled. A failure to reassemble frame fragments shall not affect the successful reception of unfragmented frames. Any received fragment shall be retained for no longer than 0.1 second pending reassembly.

Figure 20-4 provides an overview of the relationship between the component validation and extraction process, delivery of unfragmented frames, and the specification of frame reassembly state machines.



MPPDU components are validated and extracted in MPPDU reception order, and in the order encoded in those MPPDUs as specified in 19.7. Padding and unrecognized components are extracted and discarded. The attributes of each received frame fragment are communicated to the appropriate reassembly state machine using the state machine variables **fragment**, **initial**, **final**, and **fragmentNum**. If there is no such machine (e.g., if the receiving PrY supports only point-to-point connectivity, but has more than one peer PrY), the frame fragment is discarded. The variable **fragment** references an implementation dependent construct for the user data frame fragment, and is PtrToNull (0, 'clear', and FALSE) when no fragment is referenced. User data frames received in Encapsulated Frames are delivered direct to the user of the PrY's Private Port.

The default reassembly algorithm state machine (20.13.1, Figure 20-5) clears **fragment** and **initial** when the fragment is added to a user data frame reassembly or discarded, and clears **final** when the fragment is discarded or delivered to the PrY's Private Port as part of a successfully reassembled frame. The component validation and extraction process maintains the required user data frame delivery order by not presenting any following fragment to a reassembly state machine, or delivering a user data frame direct to the Private Port until **fragment**, **initial**, and **final** for each machine are clear.

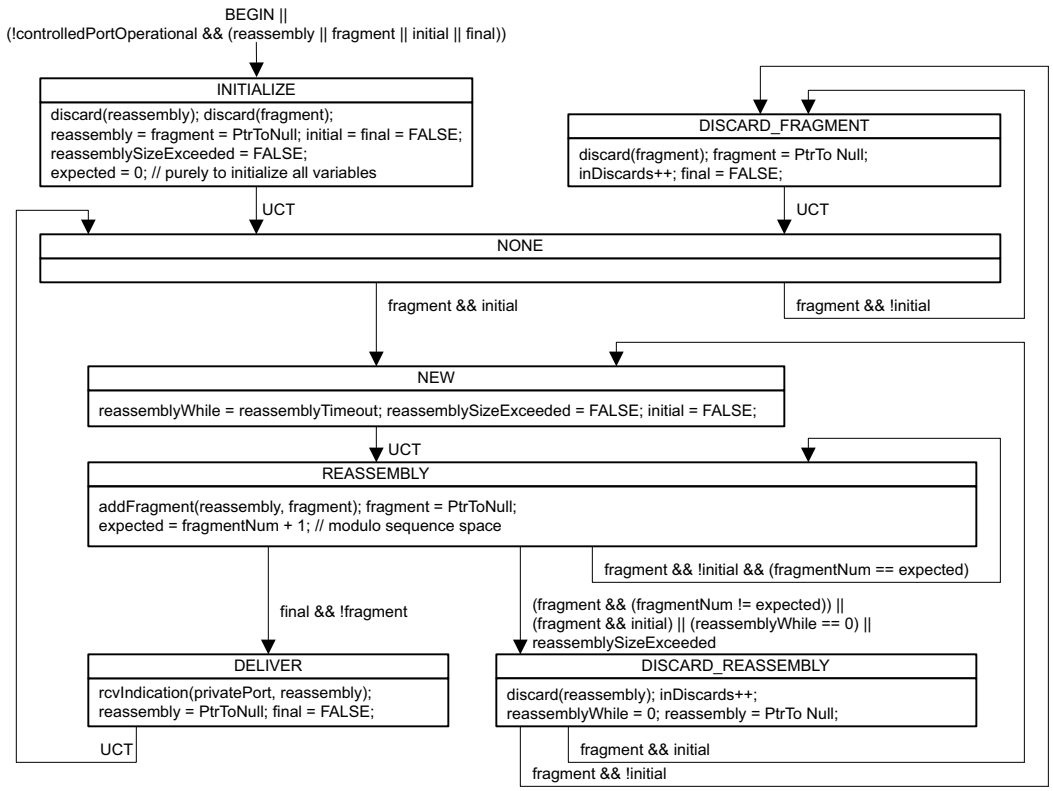
Figure 20-4—Protected frame reception and reassembly

This standard specifies a mandatory to implement default reassembly algorithm (20.13.1). This algorithm reassembles user data frames from Frame Fragments received in order from a given peer, as indicated by the relevant fragment sequence number (Express or Preemptable, 19.5.4), and discards out of order Frame Fragments. A PrY may support one or more additional algorithms subject to the constraints specified in this clause (20.13). Current use of the default algorithm is reported by ifDefaultReassembly (22.6).

NOTE 3—A privacy protection system that uses Link Aggregation and MACsec collocates PrYs with the SecYs (see 21.4) supporting individual links and transmits the fragments of any given user data frame on a single link with consecutive Frame Fragment (19.5.4) fragment sequence numbers. A supporting SecY that is configured to provide strict replay protection discards MPPDUs received out of order.

20.13.1 Default reassembly algorithm

Figure 20-5 specifies the mandatory to implement default reassembly algorithm using the state machine conventions used in this standard, IEEE Std 802.1X and IEEE Std 802.1Q. This state machine takes precedence if there is any discrepancy between it and the text of this clause (20.13).



The entry conditions to any state are true when the actions in that state are executed and those actions are executed atomically (with respect to any other state machine or process) on entry.
The component validation process sets the state machine variables **fragment**, **initial**, and **final** atomically to pass a fragment (with its initial and final indications) to the appropriate reassembly state machine, and does not pass its next Frame Fragment or Encapsulated Frame to the same or any other state machine or the Private Port's user until **fragment**, **initial**, and **final** are all False (zero or PtrToNull).
The **fragment** and **reassembly** state machine variables reference implementation dependent constructs for the fragment or reassembled frame, and are PtrToNull (0 and FALSE) if there is no current fragment or reassembly.
The **addFragment(reassembly, fragment)** procedure adds the fragment to the current reassembly, and transfers responsibility for tracking any resources previously associated with **fragment** to **reassembly**. If the size limit on the reassembled frame would be exceeded the state machine variable **reassemblySizeExceeded** is set. The **reassemblyWhile** timer comprises an implementation dependent **reassemblyTimeout** number of clock ticks equal to 0.1 second, and decremented by the system.
The **discard(fragment)** procedure recovers any associated resources associated with **fragment**.
The **discard(reassembly)** procedure discards the current reassembly and recovers any associated resource, including any associated with a fragment that has caused reassemblySizeExceeded to be set.
If the reassembly state machine is reponsible for reassembling Express Frame Fragments, the action shown as **inDiscards++** increments the inExpressDiscards count. If Preemptable Frame Fragments are being ressembled, inPreemptableDiscards is incremented.

Figure 20-5—Reassembly state machine

If (for a given peer and given class of frames) a fragment is received that is not the next in sequence, the results of any reassembly already in progress is discarded. If there is no reassembly in progress, the received fragment shall also be discarded unless it is marked as an initial fragment. Otherwise the received fragment is added to the reassembly. If the received fragment was marked as a final fragment, the reassembled frame is delivered to the Private Port.

20.14 PrY management

The PrY management process controls, monitors, and reports on the operation of the PrY, providing access to operational controls and statistics for network management through the LMI. It:

a) Maintains the MAC status and point-to-point MAC parameters (20.4) for the Private Port.

NOTE 1—Interface status reporting can be management protocol specific. See 22.3.2 for IF-MIB related detail.

b) Allows the encapsulation of user data frames in MPPDUs, and the decapsulation of received user data frames to be separately enabled or disabled for the PrY as a whole (20.5, 22.6, 20.11).

c) Allows the destination MAC Address of the PrY to be explicitly configured [5.10 n), p)].

NOTE 2—A PrY that is directly supported by a SecY uses the PAE Group Address used to support that SecY as the destination MPPDU MAC Address and does not need to be separately configured.

d) Allows the source MAC addresses used by peer PrYs to be explicitly configured, if the PrY supports (20.13) more than one peer [5.11 a)].

NOTE 3—When a PrY supports more than one peer [5.11 a)] and the connectivity provided by the Controlled Port is not point-to-point, the source MAC address of each peer is used in fragment reassembly (20.13). A PrY that is directly supported by a SecY uses the initial 6 octets of each of the peer SCIs discovered by MKA, and does not need to be separately configured.

e) Supports configuration of the Privacy Selection Table (20.5, 17.4, Figure 17-3), including the privacyType (None, PrivacyFrame, ExpressChannel, PreemptableChannel) and the parameters for each table entry that specifies Privacy Frame transmission (privacyPadding, revealDE, accessPriority), see 20.7.

f) Allows the use of the PrY's Express Channel and Preemptable Channel to be separately enabled or disabled (20.8, 17.4.2).

g) Supports configuration of each Privacy Channel's userDataFrameSize (17.4.2, 20.9.4, 20.9.5), and allows the use of user data frame fragmentation with the channel to be enabled or disabled (20.10).

h) Supports selection of the default Privacy Channel MPPDU generation algorithm (20.9.4), configuration of its requestedKbitRate and userDataBurstSize parameters, and reporting of the channelFrameSize and channelBurstSize calculated by the PrY for use with that algorithm.

i) Supports selection of the transmission gate Privacy Channel MPPDU generation (20.9.5) algorithm.

j) Maintains transmission statistics for the PrY and its Private Port interface (20.14.1).

k) Maintains reception statistics for the PrY and its Private Port interface (20.14.2).

Figure 20-6 illustrates the management information that represents a PrY's capabilities and provides control over and reporting on its operation. For convenience the figure uses UML 2.0 conventions together with C++ language constructs. For an explanation of these conventions, see *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition* [B1].

Figure 20-6 reflects the model of SecY operation described in this clause (Clause 20). It includes data made available, through the LMI, to a PrY by other components (notably the PAE and the KaY) within the same system, in addition to operations and parameters directly accessible by remote management protocols. Security considerations can restrict management access, and the use of a PrY can depend on the configuration of those other system components. For particular managed object specifications based on this clause and figure, see Clause 13 (MAC Security Entity MIB) and Clause 23 (YANG Data Models).

NOTE 4—Figure 20-6 is aligned with the operational model, while remote management models group objects for ease of access. Object names are chosen to make correspondence clear where there are organizational differences.

In Figure 20-6 the management information for each PrY is shown as indexed by privatePortNumber within a PrY system. This containment relationship complements that specified in 10.7, where management information for each SecY is indexed by controlledPortNumber. In the PrY MIB (Clause 22), each PrY is identified by an interface index (ifIndex). In the YANG module (Clause 23), each PrY is indexed by an interfaceName.

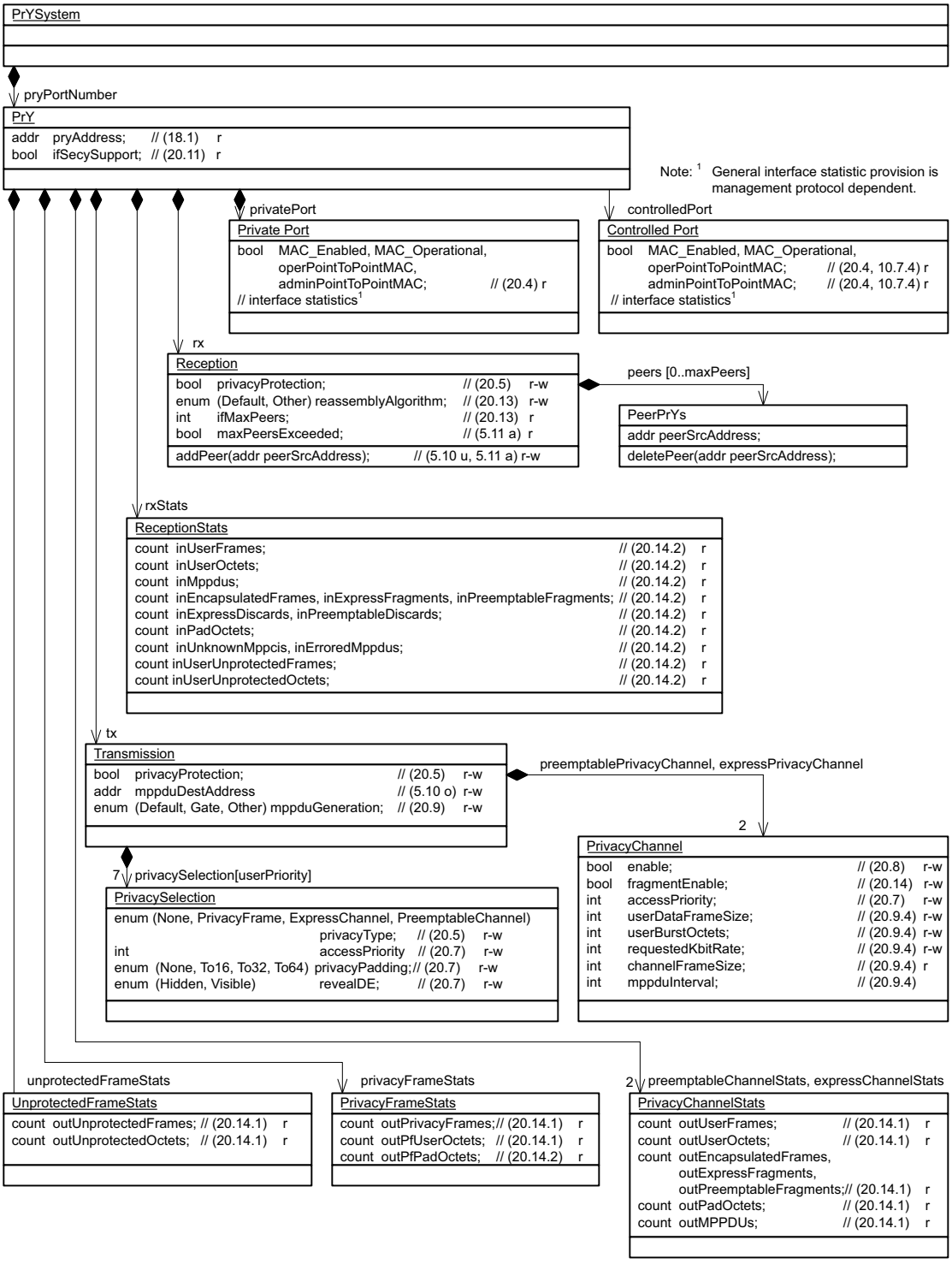


Figure 20-6—PrY Managed objects

A managed system is responsible for setting the values of certain PrY management objects when possible. When the appropriate values are available from other system components, this reduces the requirement for explicit network management of PrYs and the chance of configuration error. The values of objects that are writable by network management, and not updated by the system, shall be persistent and remain unchanged across re-initialization of the system and its management entity.

20.14.1 PrY transmission statistics

PrY transmission statistics support assessment of the efficiency of PrY operation and of the adequacy of Privacy Channel provision.

The following transmission count totals are recorded for Privacy Frames:

a) outPrivacyFrames (22.6), the number of user data frames in Privacy Frames (equally, the number of Privacy Frame MPPDUs and the Encapsulated Frame components in those MPPDUs).

b) outPfUserOctets (22.6), the number of user data frame octets.

c) outPfPadOctets (22.6), the number of padding octets.

The following transmission count totals are recorded separately for each Privacy Channel:

d) chOutUserFrames (22.6), Encapsulated Frames and Frame Fragments with the Final (F) bit set.

NOTE—A Frame Fragment with the F bit set updates both frame and fragment counts.

e) chOutUserOctets (22.6), the number of user data frame octets.

f) chOutPadOctets (22.6), the number of padding octets added to Privacy Channel MPPDUs.

g) chOutMppdus (22.6), the number of transmitted MPPDUs for the Privacy Channel.

h) chOutEncapFrames (22.6), the number of Encapsulated Frames (19.5.1) encoded in MPPDUs.

i) chOutExpFragments (22.6), the number of Express Frame Fragments (19.5.4) encoded.

j) chOutPreFragments (22.6), the number of Preemptable Frame Fragments (19.5.4) encoded.

The following transmission count totals are recorded for user data frames that are not privacy protected:

k) outUnprotectedFrames (22.6), the number of user data frames transmitted unprotected.

l) outUnprotectedOctets (22.6), the number of octets in those user data frames.

Counts of user data frame octets include the user MAC DA and SA, but not any encapsulating MPPDU's MAC DA and SA, MAC Privacy protection EtherType, MPPCIs, or FCS. The octets of each user data frame's FCS (which is not encoded in an MPPDU) are also excluded. Counts of padding octets include the MPPCI octets for Trailing Pad and Explicit Pad components. An Explicit Pad that is the last component in an MPPDU can have a *following length* that goes beyond the end of the MPPDU, only those octets that are actually sent are included in this count.

NOTE—The names of management counters are those used in Figure 10-5 and Figure 20-6. In the MIB modules these are preceded by a short names for the module and for the group or table of which they are part, and can be abbreviated to satisfy maximum name length conventions. In YANG hyphenation rather than case stropping is used to identify name components. The description clauses in both MIB and YANG modules provides a full identification of each object.

In the context of a particular system, some of this statistical information can be obtained from other system components for PrY reporting. If, for example, the user of the PrY's Private Port maintains separate frame and octet transmission counters for each user priority, then all of the counts a), b), c), d), e), k), and l) can be obtained by an appropriate summation of those user priority counters when the configuration of the Privacy Selection Table is taken into account. The sum of outMppdus and outUserFrames [g), a), and k)] is equal to the number of frames transmitted through the Controlled Port. If that count is available from its supporting protocol entity, outUserFrames [k)] need not be separately counted.

SMIv2 IF-MIB (see 22.3.2) and YANG interface statistics (see 23.4.2) can be obtained from the above counts [d) through l)].

20.14.2 PrY reception statistics

PrY reception statistics can be used, in conjunction with transmission statistics from peer PrYs, to detect network loss, peer PrY configuration errors, and PrY implementation errors. See 19.4, 19.7, 20.11, and 20.12 for additional detail.

The following counts are recorded, in total, for received MPPDUs:

a) inUserFrames (22.6), the number of user data frames successfully received from correctly encoded Encapsulated Frames or reassembled from Frame Fragments [e), f) and g), below].

b) inUserOctets (22.6), the number of octets in those user data frames.

c) inPadOctets (22.6), pad octets received in MPPDUs.

d) inMppdus (22.6), the number of MPPDUs received.

e) inEncapsulatedFrames (22.6), valid Encapsulated Frames (19.5.1, 19.7).

f) inExpressFragments (22.6), valid Frame Fragments with the Express (E) bit set (19.5.4, 19.7).

g) inPreemptableFragments (22.6), valid Frame Fragments with the E bit clear (19.5.4, 19.7).

h) inExpressDiscards (22.6), the number of times reassembly (20.13) discards an Express Frame Fragment or an in progress user data frame reassembly of those Frame Fragments.

i) inPreemptableDiscards (22.6), the number of times reassembly (20.13) discards a Preemptable Frame Fragment or an in progress user data frame reassembly of those Frame Fragments

   NOTE—A reassembly state machine is not required to maintain a count of fragments previously added to an in progress reassembly, so discard counters are not necessarily a count of received but discarded fragments. Discarding can result in the loss of fragments previously successfully added to an in progress reassembly.

j) inUnknownMppcis (22.6), MPPDU components that are not recognized (19.5.5).

k) inErroredMppdus (22.6), MPPDUs containing an incorrectly encoded component (19.5.5).

The following counts are recorded, in total, for user data frames that are not privacy protected:

l) inUserUnprotectedFrames (22.6), received user data frames that were not privacy protected.

m) inUserUnprotectedOctets (22.6), the number of octets in those user data frames.

The sums of the inUserFrames and inUserUnprotectedFrames [a) and l)] and the inUserOctets and the inUserUnprotectedOctets counts [b) and m)] reflect number of user data frames delivered to the user of the PrY's Private Port. In the absence of frame loss and network congestion due to increased use of bandwidth they should be unaffected by the decision to privacy protect, in a Privacy Channel or a Privacy Frame, any particular user data frames.

In the context of a particular system, some of these reception statistics can be obtained from other system components for PrY reporting.

The sum of inMppdus [d)] and inUserUnprotectedFrames is the number of frames received from the Controlled Port. If that total is available from the protocol entity supporting the Controlled Port, one of those counts need not be separately maintained by the PrY, although it is reported as part of the reception statistics.

If the inUnknownMppcis count is non-zero, the peer PrY has either been incorrectly implemented or is using a MPPDU component identifier reserved for future standardization.

20.15 PrY performance requirements

Time-sensitive networking (TSN) applications can benefit from or further constrain delays and delay variances experienced by relayed and transmitted frames (see IEEE Std 802.1AS, IEEE Std 802.1Q).

Insert the following text (Clause 21) after Clause 20:

21. MAC Privacy protection in Systems

This clause specifies how MAC Privacy protection is supported by the following:

a) Interface stacks in general (21.1), and end station (21.2) and bridge interfaces (21.3) in particular.

b) Link Aggregation (21.4).

c) Ethernet Data Encryption devices (EDEs, Clause 15) and related systems (21.5).

d) Shared media (21.6).

e) Multi-access LANs (21.7).

f) MACsec and MAC Privacy protection in separate systems (21.8).

Interoperability between systems requires not only interoperability between MAC Privacy protection Entity (PrY) implementations and use of the same LAN MAC technology, but also use of the same, or compatible, functions in the same relative position within interface stacks (21.1), as specified in this clause.

NOTE 1—An understanding of architectural concepts common to this and other IEEE 802.1 standards is essential to understanding this standard's specification of MAC Privacy protection. The reader is encouraged to review Clause 7 of IEEE Std 802.1AC-2016 and Clause 11 of this standard. Clause 17 provides an overview of MAC Privacy protection, and Clause 20 specifies the internal operation of PrYs. Some, but not all, of the provisions of those clauses are repeated to reduce the burden on the reader of frequent cross-referencing.

NOTE 2—The list of systems described in this clause is not intended to be exhaustive. Privacy protection can be provided wherever MACsec is used.

21.1 MAC Privacy protection interface stacks

Each PrY uses a MAC Internal Sublayer Service (ISS, IEEE Std 802.1AC) access point and provides the ISS at its secure Private Port. This allows use of MAC Privacy protection with other media-independent functions. Privacy protection depends not only on the operation of the MAC Privacy protection Entities (PrYs) that generate and validate the MPPDUs that convey user data frames, but also on the confidentiality, data integrity, and data origin authenticity of those MPPDUs. Each PrY should be directly supported by a MAC Security Entity (SecY) as shown in Figure 21-1. A PrY and its supporting SecY can also be located in adjacent systems (21.8) to support deployment of MAC Privacy protection in networks where MACsec is already deployed, or MACsec protection is separately managed.

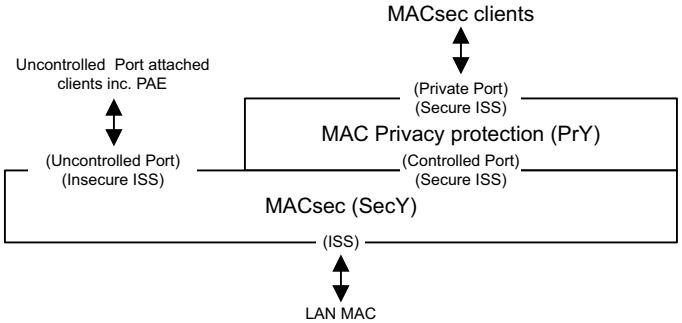


Figure 21-1—A MAC Privacy protection interface stack

Where the PrY is directly supported by a SecY (as shown in Figure 21-1), all MAC Clients that would otherwise be attached to the SecY's Controlled Port should be attached to the PrY's Private Port. The overhead of privacy protection is acceptable for most protocols (see 17.3 for discussion of the Quality of Service impact and mitigation) and a receiving peer PrY delivers the original user data frame to its client.

NOTE 1—The transmission order of frames is determined by the PrY's client. The PrY does not buffer user data frames, except as required to fill the next MPPDU to be transmitted, with the possibility of a fragment that does not fit being held over to a subsequent MPPDU. However if the PrY's client requests transmission of an express user frame before the PrY has completed transmission of the fragments of a default user frame, the express user frame can be encoded in an MPPDU that is transmitted prior to the MPPDU used to encode those fragments. This handling of express user frames supports use of the preemption as provided by the IEEE Std 802.3 MAC or equivalent capabilities.

A transmitting PrY can selectively protect protocols (17.2) without the need to coordinate the selection of those protocols with a peer PrY: a PrY that receives a user data frame that has been validated by its SecY but is not identified as an MPPDU to be processed by that PrY is simply passed to its client. Once a received MPPDU addressed to a PrY has been validated by its SecY, that PrY's processing of each user data frame component (an entire frame or a fragment) conveyed by the MPPDU is independent of the processing of other components in that MPPDU (with the exception of fragment reassembly) and independent of the addressing and priority parameters of the MPPDU. This independence allows a transmitting PrY to use a range of privacy protection and encapsulation strategies, again without the need to coordinate their selection with a peer PrY. This standard specifies management controls based on the user priority of each data frame, as the user priority is commonly used to distinguish between broad application classes (see IEEE Std 802.1Q) such as network critical frames or frames associated with flows requiring bandwidth allocation. A PrY may support additional management controls, including the use of flow classification (IEEE Std 802.1Q) to selectively or differentially protect user data frames of different protocols. Protection parameters can be associated with a stream_handle sub-parameter of the ISS connection_identifier parameter (see IEEE Std 802.1CB). The PICS should include Additional Information (A.3.2) for any additional management controls provided by a PrY implementation.

NOTE 2—The use of selective protection, or of varying MPPDU parameters (priority and size), can compromise privacy by revealing the characteristics of, and changes in, protected traffic. Maintaining a constant frequency, size, and bandwidth for observable traffic types can have a greater impact on quality of service than treating those types as an undifferentiated whole. This standard accordingly specifies the use of at most two Privacy Channels to control transmission timing and fragmentation of user data frames.

The use of MAC Privacy protection does not change the use or selection of MAC Clients attached to the SecY's Uncontrolled Port.

21.1.1 PrY Addressing

System interoperability also depends on the suitable selection of the destination address of the MAC Privacy protection Data Units (MPPDUs) that each PrY uses to encapsulate user data frames. The operation of networks depends on access by the intermediate systems that make up that network (bridges, routers, and switches) to the destination and source addresses of frames to be forwarded. Forwarding systems can require access to the user data conveyed to associate frames with data flows and bandwidth reservations, and to participate in control protocols. The intermediate systems' use of MACsec to validate frames, confining frames that have been corrupted or introduced by an attacker to individual LANs and protecting control protocol operation, also means that those systems have access to user data and addresses. Privacy protection depends not only on the operation of the MAC Privacy protection Entities (PrYs) that generate and validate the MPPDUs that convey user data frames, but also on the MACsec confidentiality-protection of those MPPDUs. The extent of each privacy protected region of a network is thus aligned with the MACsec protection of that region, which is typically an individual LAN but can be an equivalent LAN service supported by a provider network. To maintain this alignment without depending on additional configuration, the destination MAC address of each MPPDU is the PAE Group Address (Table 15-2) configured for MKA support of that SecY, except as specified for non-collocated SecY and PrY (21.8).

21.2 Privacy protection for end station interfaces

An end station can use the interface stack illustrated by Figure 21-1. If the end station transmits frames that are VLAN tagged, tagging (for transmission) and de-tagging (on reception) is carried out by the MACsec Client, just as if privacy protection was not being provided.

In network scenarios where the end station identity is not strongly correlated with location, and all frames are privacy protected, the use of a locally assigned MPPDU source MAC address can help to maintain privacy. The encapsulated user data frame's source address can continue to be used (as is common practice) by a system that incorporates the receiving PrY to index end station related information.

NOTE—A media access control method can use control frames, or have other operational properties that are correlatable with station identity. See IEEE Std 802E-2020 [B5].

21.3 MAC Privacy protection for bridge interfaces

MAC Bridges are specified in IEEE Std 802.1Q. The MAC Relay Entity forwards frames between the interface stacks supported by each of the Bridge Ports. Figure 21-2 shows MAC privacy protection interface stacks for a VLAN-unaware Bridge (on the left) and a VLAN-aware Bridge (on the right), compare with Figure 11-5 and Figure 11-7.

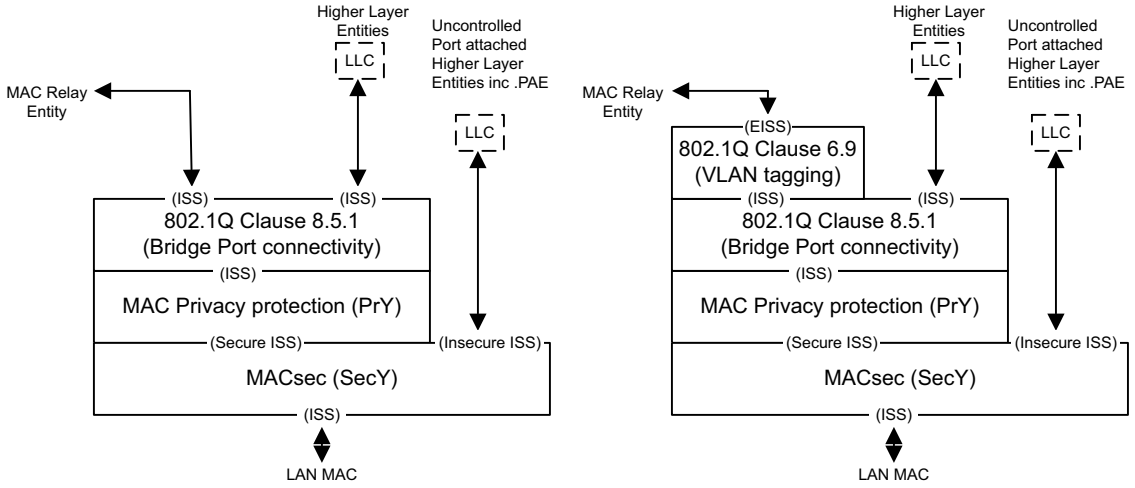


Figure 21-2—MAC Privacy protected Bridge Ports

NOTE—If the MAC Bridge aggregates multiple LANs to support a single Bridge Port, each individual LAN supports its own PrY and SecY, which together provide a secure privacy protected MAC Service to the Link Aggregation sublayer, as specified in 21.4. Each aggregated port then provides that service to the Bridge Port transmit and receive functions.

Figure 17-2 shows the MPPDU frame format supported by these interface stacks, including the relative placement of the PrY MAC Addresses, MAC Privacy protection EtherType, SecTAG, and user data frame (including the VLAN tag, if present). The use of MACsec and privacy protection is not necessarily enabled for, or supported by, all ports of any given bridge. The scope of privacy protection is, as for MACsec, from a bridge port to its nearest peer neighbor, e.g., from Customer Bridge Port to Customer Bridge Port, from Customer Bridge Port to an end system, or from Provider Bridge Port to Provider Bridge Port.

21.4 Privacy protection for Link Aggregation

Link Aggregation is specified in IEEE Std 802.1AX™. The service provided by two separate point-to-point LANs is combined to provide a single service interface. To provide MAC Security for such a system, two independent SecYs operate below the link aggregation sublayer (see 11.5). Privacy protection is provided for each SecY by a PrY that is a user of the secure MAC Service (ISS) provided by that SecY, and in turn supports the link aggregation sublayer (Figure 21-3, compare with Figure 11-8). Privacy Channel(s) for each PrY can be configured to schedule MPPDU transmission, eliminating or reducing the potential correlation between MPPDU transmission and changes in the traffic flows that link aggregation assigns to one individual link or another.

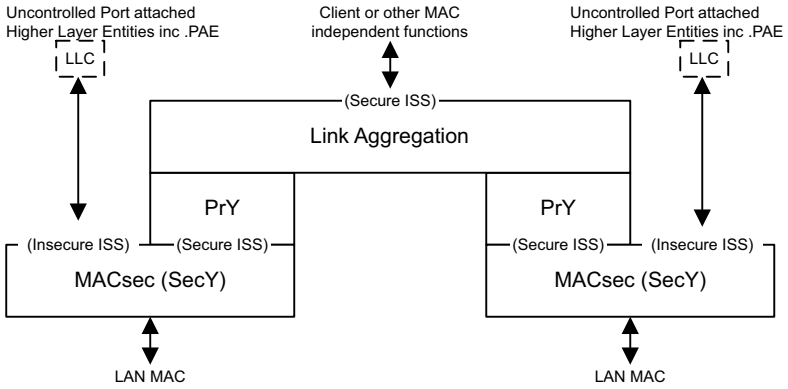


Figure 21-3—Privacy protection and Link Aggregation

If a given PrY fragments a user data frame, all the fragments of that frame are conveyed in MPPDUs protected by the SecY that supports that PrY. This facilitates the simple reassembly function (20.13.1) mandated for fragmentation support, assuming that MPPDUs transmitted with a given access priority (and thus all MPPDUs for a given Privacy Channel) and protected by a given SecY are received in order.

NOTE—MPPDU fragment encoding (19.5.4) can support reassembly of out of order fragments, but this is not required PrY reception capability (20.13).

21.5 EDEs with MAC Privacy protection

An EDE (Clause 15) can provide MAC Privacy protection for traffic transiting a Provider Bridged Network (PBN) by incorporating a PrY in the interface stack with a SecY, as illustrated in Figure 21-4 for an EDE-CC (compare with Figure 15-9).

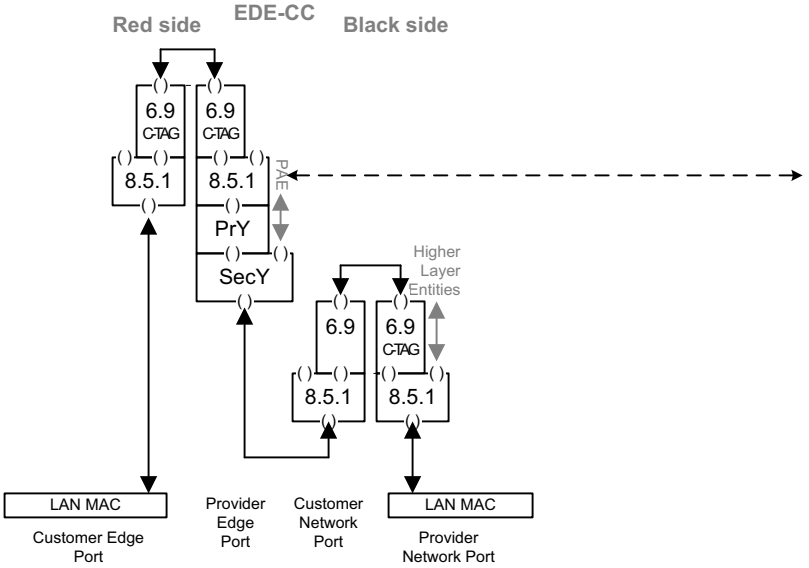


Figure 21-4—EDE-CC with MAC privacy protection

Figure 21-4 provides a sectional view of the interface stacks, showing a single Provider Edge Port (PEP. An EDE-CC can include multiple PEPs so the PBN can use each frame's C-TAG to select the destination interface. In Figure 21-5, the member set for each VID (as specified for the Forwarding Process of a VLAN Bridge component by IEEE Std 802.1Q) includes at most one PEP so the edge C-VLAN Bridge component (shown on the left of EDE-CC1, connecting the CEP and PEPs) can forward a frame received from bridge B1 to just one PEP. That PEP can then protect the user data frame, encapsulating it within an MPPDU that is integrity and confidentiality protected by MACsec before it is forwarded to one of the Customer Network Ports (CNP) of the network C-VLAN Bridge component (shown on the right of EDE-CC1, connecting the CNPs and the Provider Network Port).

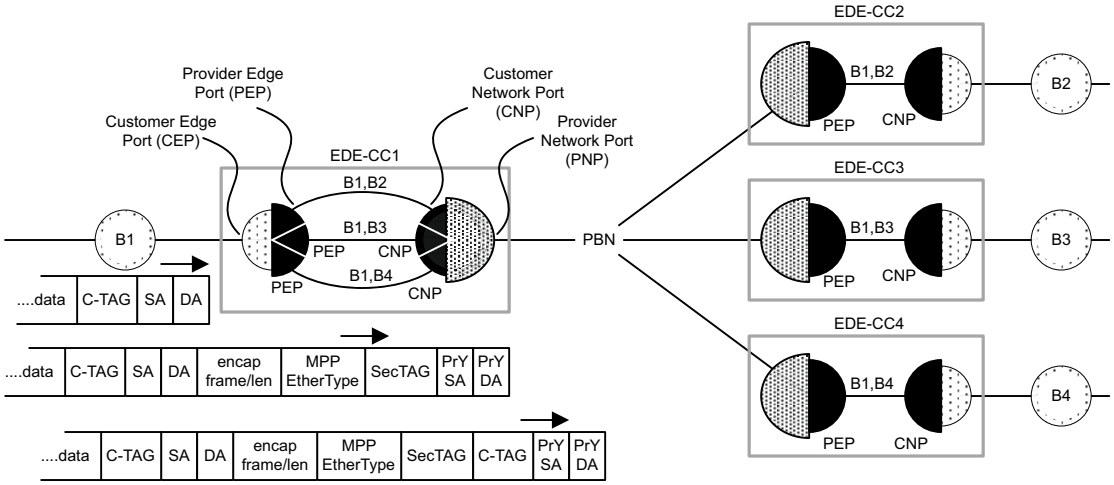


Figure 21-5—EDE-CCs communicating over a PBN

The initial octets of the protected MPPDU compose the SecTAG added by the PEP, so the CNP assigns it to the C-VLAN identified by the Port VLAN Identifier (PVID, as specified by IEEE Std 802.1Q for Support of the EISS) for that Bridge Port. The frame is then transmitted, C-tagged with that PVID, through the Provider Network Port (PNP). The PBN can then use that outer C-TAG to identify the destination PBN interface, modifying that tag if necessary. The original user data frame C-TAG is delivered, privacy protected within the MPPDU, to the distant customer equipment.

NOTE 1—IEEE Std 802.1AEcg-2017 and IEEE Std 802.1AE-2018 required an EDE-CC and EDE-SS to tag frames transmitted by the PNP with the same VID as received by the CEP [5.8 e) and 5.9 e) of IEEE Std 802.1AE-2018]. This configuration constraint simplified management, and avoided potential network forwarding changes as a result of enabling or disabling MACsec protection. EDEs implementing this constraint remain conformant to IEEE Std 802.1AE, but user data frames for different customer VLANs that are destined for the same remote PBN interface can also be forwarded through the same PEP by a conformant EDE. Those user data frames can then be encapsulated in a common set of MPDUs, and use the same SC and outer VID in the interests of privacy.

When an EDE-CC's PNP receives a frame, its network C-VLAN component uses the member set for the VID in the outer C-TAG to select a PNP. The frame is transmitted untagged by that PNP, allowing the SecY for the attached PEP to recognize the frame as MACsec protected and to validate the frame. An associated PrY, above the SecY in the PEP interface stack can then decapsulate the original user data frames (including their original VLAN tags, if present) and pass them up the interface stack to the edge component's MAC Relay Entity.

Privacy protection can be enabled or disabled on a PEP by PEP basis: the customer equipment associated with a remote PBN interface accessible through a given EDE is not all necessarily capable of privacy protection, and can have MACsec disabled.

NOTE 2—The use of two bridge components to model the EDE behavior follows the approach used in IEEE Std 802.1Q to model Provider Edge Bridges. In the simplest cases this is elaborate, but has the advantage that a system whose behavior can always be explained in terms of the valid interconnection of existing network components does not introduce new interoperability challenges. In other words, bundling any part of a valid network of components into a distinct physical system always yields a valid (if not necessarily desirable) system. The model has the further advantage of explaining how further functionality, standardized separately for individual bridge components, can be added to an EDE or EDE-like system, and what the resulting externally observable behavior of the system should be. Connectivity Fault Management, as specified by IEEE Std 802.1Q, is one example of such functionality. The internal implementation of an EDE is not over-constrained by the two component model, conformance to this standard and to the provisions of IEEE Std 802.1Q is only to the externally observable behavior implied by the model.

NOTE 3—This standard's specification of EDEs is not intended to encompass all the possible functionality of a two bridge component device. Restricting the edge component's member set for each C-VLAN to include at most one PEP means that the edge component need not learn from customer MAC Addresses (see the enhanced filtering utility criteria specified by IEEE Std 802.1Q) and that the outer VID is determined by the inner VID. More general functionality could include learning the association of MAC Addresses with a particular PEP, with the possibility of dynamic determination of the PEP and thus of the outer VID in the VLAN tag added prior to transmission by the PNP.

21.6 Privacy protection with shared media

Privacy protection can be used with shared media and group CAs. The use of virtual shared media with a single VLAN and explicitly configured PrY addresses to optimize delivery of MPPDUs is not recommended, as enabling or disabling privacy protection would then result in a change in the underlying connectivity, possibly introducing or revealing configuration problems and making their diagnosis more difficult. If the simple use of VLANs to partition the connectivity provided by the shared media is not sufficient to meet scaling goals, the use of Provider Bridging or Provider Backbone Bridging as specified by IEEE Std 802.1Q is indicated, with privacy protection provided by the Provider Edge Port (PEP) interface stacks.

21.7 Privacy protection and multi-access LANs

Privacy protection can be used in conjunction with MACsec to support multi-access LANs (see 11.8). The MAC Address used by each PrY as the source MAC address of transmitted MPDUs is that used by protocols (EAPOL and MKA, see IEEE Std 802.1X) using the Uncontrolled Port to establish secure connectivity.

NOTE—As with end station interfaces in general (21.2) MAC Privacy protection does not prevent an observer from determining which MPPDUs are destined for which station, but does allow the end station user and a service provider to continue to use a permanent address associated with the station to identify information (e.g., access rights) for the station, without exposing that address to an unauthorized observer.

21.8 Separate privacy protection devices

MPPDUs are identified by a distinct EtherType (19.3) and can be transmitted and received without MACsec protection, so that protection and validation can be performed by separate devices or systems, possibly already deployed in a network or separately administered. Figure 21-6 illustrates the encapsulation of a user data frame, shown for simplicity as the initial component of an MPPDU, by a pair of such separate systems each on the red-side of a pair of interconnected EDEs. These systems can be modeled in the same way as a two bridge-component EDE, with the substitution (rather than the addition) of a PrY for the SecY in each PEP interface stack. The use of these separate privacy protecting systems is predicated on a desire to maintain separate MACsec protecting and validating systems for communication between all the connected PBN interfaces: if the associated EDE retains its default configuration it adds a copy of the outer C-TAG after the SecTAG prepended to each MPPDU transmitted across the PBN (compare Figure 21-6 with Figure 21-5), and does not inter-operate with an EDE that has both PrY and SecY in the PEP interface stack. An MPPDU destined to the group MAC Address used by the MACsec SecY is filtered by the receiving PEP, so a PrY in a privacy protecting system can be configured to use the individual MAC Address of its peer as the destination PrY MAC Address. Since an existing EDE or system on the path between the receiving EDE and the destination decapsulating device can be presumed not to recognize MPPDUs, the EDE and that device are best placed adjacent to one another, without intervening devices.

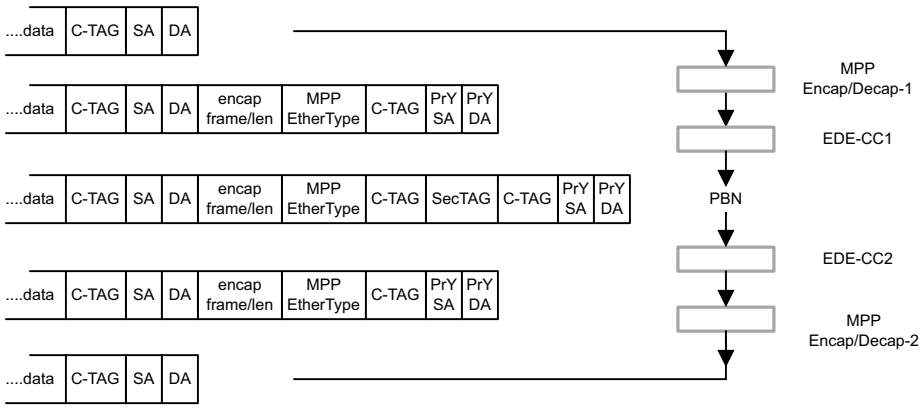


Figure 21-6—MAC Privacy protection using existing EDEs

The addressing and MPPDU format constraints described in this clause (21.7) for the deployment of MAC Privacy protection and MACsec confidentiality protection in separate systems do not apply when the functionality is provided in a single system comprising multiple components. The information that a receiving SecY needs to pass to its associated PrY is all present as parameters of an ISS indication, and each receive indication from the PrY to its client comprises the parameters expected with receipt of a user data frame. Individual MPPDU components (19.5) are self describing and can be extracted and processed separately by the PrY, with the exception of frame fragments received from a group CA. In that case the PrY needs to use the MPPDU's PrY source MAC Address to reassemble user data frame fragments received from different members of the CA.

Insert the following text (Clause 22) after Clause 21:

22. MAC Privacy protection Entity (Pry) MIB

22.1 Introduction

This clause contains an SMIv2 Management Information Base (MIB) module, based on the specification in Clause 20, 20.14, and Figure 20-6, for managing the operation of a MAC Privacy protection Entity (PrY).

22.2 The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of IETF RFC 3410 [B7].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This MIB module is compliant to SMIv2, as described in IETF RFC 2578, IETF RFC 2579, and IETF RFC 2580.

22.3 Relationship to other MIBs

22.3.1 System MIB Group

It is assumed that a system implementing this MIB also implements the “system” group defined in IETF RFC 3418 (or at least that subset of the system group defined in IETF RFC 1213).

22.3.2 Relationship to the Interfaces MIB

It is assumed that a system implementing this MIB module implements the “interfaces” group defined in IETF RFC 2863, the Interfaces Group MIB. This MIB includes the clarifications mandated by IETF RFC 2863 for any MIB that is medium-specific or an adjunct of the Interfaces Group MIB.

A MAC Privacy protection Entity (PrY) is a shim layer in an interface stack, and is integrated with the IF-MIB as shown in Figure 22-1.

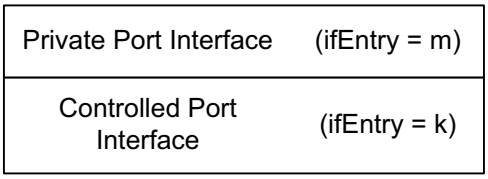


Figure 22-1—PrY Interfaces

The PrY’s Private Port is a service access point that provides one instance of the secure privacy protected MAC Service. Managed objects in this PrY MIB are indexed by the ifIndex for the PrY’s Private Port. The interface type for the Private Port is macSecControlledIF(231).

The PrY’s Controlled Port is a service access point that supports the PrY with a secure confidentiality protected instance of the MAC Service. Management parameters for the Controlled Port are as specified for

the protocol entity supporting that Controlled Port (typically a SecY, see Clause 10, 10.7, and Clause 13), and are referenced by the ifIndex for that interface.

The ifTable and ifStackTable makes the interface stack layering shown in Figure 22-1 explicit, with separate interfaces and interface indices for the Private and the Controlled Port, to allow the interface statistics for both Ports to accurately represent the numbers of packets they send and receive. These statistics can differ significantly: if a Privacy Channel (17.4.2, 20.9) is configured, ifOutOctets and ifOutUcastPkts for the Private Port remain unchanged for periods during which the PrY's user is not transmitting frames while the same statistics for the Controlled Port continue to reflect the regular transmission of Privacy Channel MPPDUs. The Private Port's interface statistics are of interest when considering the load presented by processes sending user data frames, while the Controlled Port's interface statistics reflect the load place on the supporting network elements.

The Private Port's ifAdminStatus is read-only and is `up(1)` if, and only if, MAC_Enabled (6.4, 20.4) is True. Otherwise ifAdminStatus is `down(2)`. The ifOperStatus is read-only and is `up(1)` if, and only if, MAC_Operational (6.4, 20.4) is True. Otherwise ifOperStatus is `lowerLayerDown(7)`.

NOTE 1—The Private Port's MAC_Enabled and MAC_Operational parameter values are the same as those for the Controlled Port.

The attributes in Table 22-1 are part of the required ifGeneralInformationGroup object group specified in IETF RFC 2863, and are not duplicated in the PrY MIB.

Table 22-1—Use of ifGeneralInformationGroup Objects

ifGeneralInformationGroup Objects	Use for PrY
ifDescr	See IETF RFC 2863.
ifType	macSecControlledIF(231).
ifSpeed	same as the Controlled Port interface ifSpeed.
ifPhysAddress	This object should have an octet string with zero length.
ifAdminStatus	Read-only, see above (22.3.2).
ifOperStatus	Read-only, see above (22.3.2).
ifLastChange	See IETF RFC 2863.
ifName	See IETF RFC 2863.
ifLinkUpDownTrapEnable	See IETF RFC 2863. Default: disabled(2).
ifHighSpeed	Same as the Controlled Port's ifHighSpeed.
ifConnectorPresent	Read-only: false(2).
ifAlias	See IETF RFC 2863.
ifTableLastChange	See IETF RFC 2863.

The attributes in Table 22-2 are part of the required ifCounterDiscontinuityGroup object group specified in IETF RFC 2863 and are not duplicated in the PrY MIB.

Table 22-2—Use of ifCounterDiscontinuityGroup Object

ifCounterDiscontinuityGroup Object	Use for MACsec
ifCounterDiscontinuityTime	See IETF RFC 2863. Always 0, no discontinuity.

The ifStackTable is used to identify the layer relationships between PrY's Private Port (upper) interface, with an ifIndex value that identifies the PrY's managed objects, and the PrY's Controlled Port (lower) interface (with an ifIndex that identifies the protocol entity and managed objects supporting that port.

The use of the ifPacketGroup object group specified in IETF RFC 2863 is described in 20.14.

The ifRcvAddressTable is not applicable to PrY operation, as a PrY is always capable of receiving user data frames that are not MAC Privacy protected. The MAC Destination Address of those frames is the MAC address of the intended recipient, and not necessarily an address associated with the PrY or with the interface stack of which the PrY is a part.

NOTE 2—As specified, a PrY always operates in 'promiscuous mode'. As specified in IETF RFC 2863, entries in the ifRcvAddressTable are only required for those addresses for which the system would receive frames were it not operating in promiscuous mode. Other components in the system of which the PrY is a part can be configured to selectively forward, filter, or receive frames.

22.4 Security considerations

MAC Privacy protection is supported by the MAC Security protocol (MACsec) which, when properly deployed, can help to protect against inadvertant transmission to or from unauthenticated and unauthorized parties. Unauthorized access to management objects defined in this MIB module with MAX-ACCESS read-only, as well as read-write or read-create, can compromise privacy as follows:

a) The `ieee8021PryOutTable`, `ieee8021PryChannelOutTable`, and `ieee8021PryInTable` together provide direct statistical information (though not real-time timing information) about the true level of data traffic passing through the interface.

b) The configuration of the `ieee8021PrySelectionTable` , `ieee8021PryFrameTable`, and the `ieee8021PryChannelTable` can reveal information that can be correlated with the intended use of the interface, even if it is not carrying that traffic at present.

c) The `ieee8021PrySelectionTable`, in combination with the `ieee8021PryFrameTable` and the `ieee8021PryChannelTable` provides detailed information about the Privacy Frame, Privacy Channel, and access priority (potentially visible in a VLAN tag added by an EDE-CC or EDE-SS) support of user data frames of each priority. This information can be useful to an adversary seeking information that is correlated with the use and changing use of particular applications.

Some of this information, and further detailed information, can also be obtained by accessing read-only, as well as read-write or read-create, objects in other MIB modules supported by the system. It is important, if privacy its to be protected, that all MIB module access is restricted to authorized parties.

Access to read-write objects can compromise basic network operation as well as privacy, as follows:

d) Privacy Channels can be configured, through the `ieee8021PryChannelTable`, to use excessive network bandwidth or to use so little bandwidth that service is effectively denied to some or all traffic passing through the interface. If the PrY is not collocated with a SecY, the

`ieee8021PryIfMppduDA` in the `ieee8021PryIfTable` can be configured to transmit excess traffic throughout the network.

e) If privacy protection is not desired, it is advisable to disable both `ieee8021PryIfTxProtection` and `ieee8021PryIfRxProtection` in `ieee8021PryIfTable`. If an adversary can enable receive privacy protection and can transmit a data frame that is relayed through the network (as if from an authorized source) and addressed to an PrY, that frame can be processed by that PrY and forwarded with a destination or source MAC address that would have resulted in prior frame filtering if not hidden by privacy protection.

f) The `ieee8021PryFrameTable` can be configured to change, or even reorder, the priority used to transmit user data frames, impacting network performance.

This subclause (22.4) does not provide a comprehensive description of of privacy, security, and operational exposures that can result from unauthorized access to the PrY MIB and other system MIB objects. Additional vulnerabilities of greater or lesser significance can exist.

22.5 Structure of the MIB module

A single MIB module is defined in this clause (Clause 22). In the MIB module each PrY is identified by the InterfaceIndex used by the Interfaces MIB (22.3.2, Figure 22-1) for its Private Port. This facilitates identification of the PrY when investigating an interface stack, and discovery of the other interface stack entities (via the Interfaces MIB) related to a particular PrY, including a supporting SecY (if present) and its associated PAE. Figure 22-1 illustrates the structure of the MIB module.

At the top level, the MIB module comprises management and statistics objects, both with an initial OID (object identifier) of ieee8021PryMIBObjects, and MIB conformance information with an initial OID of ieee8021PryMIBConformance. No notifications are defined.MIB objects are arranged in a number of tables (in MIB terms a SEQUENCE OF entries) with each entry in the table comprising an number of basic objects (in MIB terms a SEQUENCE OF objects such as truth values, integers, text strings).

The entries in each of the management and statistics objects tables are indexed in one of the following ways:

a) By the interface index, if the objects in each entry are for the PrY identified by that index).
b) By the interface index and user priority, for PrY objects that depend on user priority based Privacy Selection.
c) By the interface index and Privacy Channel (Express or Preemptable) for Privacy Channel management and statistics.

   NOTE 1—Privacy Channel objects only apply to PrY transmission. A PrY's peers' transmit behavior is only constrained by this standard as needed for interoperability (17.4, 18.5).

d) By interface index and MAC Address to identify the peers of a given PrY.

The MIB conformance objects are organized into compliance statements and conformance groups.

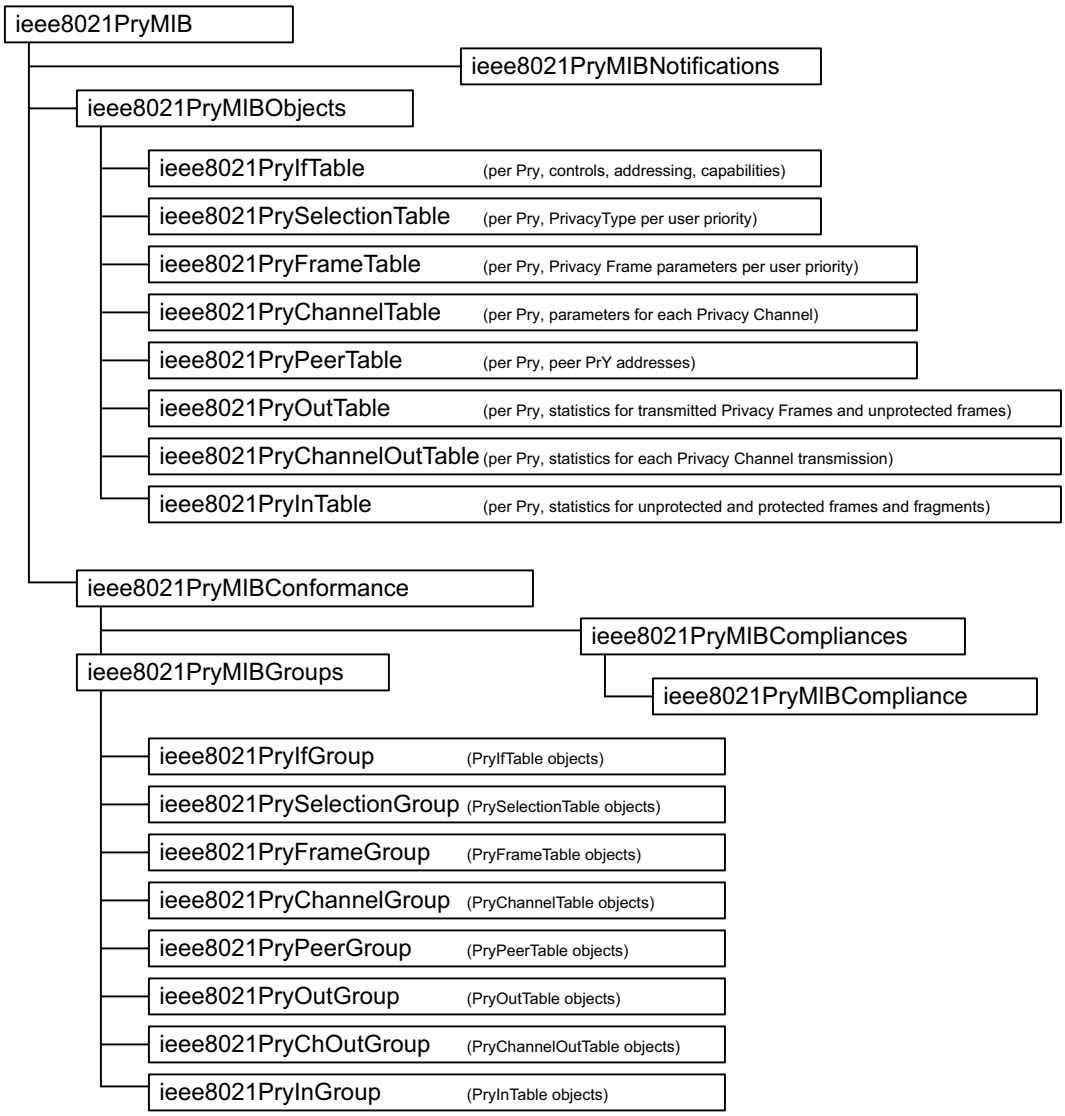


Figure 22-2—PrY MIB structure

22.6 MAC Privacy protection Entity (PrY) MIB definition[12, 13]

```
IEEE8021-PRY-MIB DEFINITIONS ::= BEGIN

-- ============================================================================
-- IEEE802.1AE MAC Privacy protection Entity (PrY) MIB
-- ============================================================================

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    Integer32, Unsigned32, Counter64
        FROM SNMPv2-SMI
--  SnmpAdminString

--      FROM SNMP-FRAMEWORK-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    MacAddress, TruthValue, RowStatus
        FROM SNMPv2-TC
    InterfaceIndex
        FROM IF-MIB
    IEEE8021PriorityValue
        FROM IEEE8021-TC-MIB
;
ieee8021PryMIB MODULE-IDENTITY
    LAST-UPDATED "202210310000Z" -- October 31, 2022
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
       "WG-URL: http://www.ieee802.org/1
        WG-EMail: stds-802-1-L@ieee.org/1/
         Contact: IEEE 802.1 Working Group Chair
           Postal: C/O IEEE 802.1 Working Group
                   IEEE Standards Association
                   445 Hoes Lane
                   Piscataway, NJ 08854
                   USA
           E-mail: stds-802-1-chairs@ieee.org"
    DESCRIPTION
       "The MAC Privacy protection Entity (PrY) MIB module.

        Unless otherwise indicated, the references in this MIB module are to
        IEEE Std 802.1AE-2018 as amended by IEEE Std 802.1AEdk-2023.

        Copyright (C) IEEE (2022).
        This MIB module is part of IEEE Std 802.1AE; see that standard and its
        amendments for full legal notices.

        A MAC Privacy protection Entity (PrY) is a protocol shim in an
        interface stack that encapsulates user data frames in MAC Privacy
        protection Data Units (MPPDUs). Once those MPPDUs are confidentiality
        protected by MACsec, the ability of potential adversaries to draw
        conclusions from the source and destination MAC addresses, sizes, and
        transmission timing and frequency of user data frames is reduced or
        eliminated.

        Each PrY in a system and its managed objects is indexed by the
        InterfaceIndex(ifIndex) of its upper interface (Private Port), which
        provides a privacy protected service to its user, typically a Bridge
        Port(IEEE Std 802.1Q) or an end station protocol stack. Object names
        can be conveniently pronounced by rendering 'Pry' as 'Privacy ', and
        'If' as 'Interface '.
        "
```

[12] *Copyright release for MIBs:* Users of this standard may freely reproduce the MIB modules in this standard so that they can be used for their intended purpose.

[13] The MIB text in this clause includes clickable cross-references to the other clauses of this standard and to MIB objects (highlighted). A plain text (UTF-8) version of this MIB is attached to the PDF version of this standard, and can be obtained by Web browser from the IEEE 802.1 Website at htttps://1.ieee802.or/mib-modules/.

```
    REVISION "202210310000Z" -- October 31, 2022
    DESCRIPTION "Initial Revision"
    ::= { iso (1) iso-identified-organization (3) ieee (111)
    standards-association-numbered-series-standards (2) lan-man-stds (802)
    ieee802dot1(1) ieee802dot1mibs(1) 36 }
-- ============================================================================
-- subtrees in the PrY MIB

ieee8021PryMIBNotifications OBJECT IDENTIFIER ::= { ieee8021PryMIB 1 }
ieee8021PryMIBObjects       OBJECT IDENTIFIER ::= { ieee8021PryMIB 2 }
ieee8021PryMIBConformance   OBJECT IDENTIFIER ::= { ieee8021PryMIB 3 }
-- ============================================================================
--ieee8021PryMIBObjects
--  ieee8021PryIfTable
--  ieee8021PrySelectionTable
--  ieee8021PryFrameTable
--  ieee8021PryChannelTable
--  ieee8021PryPeerTable
--  ieee8021PryOutTable
--  ieee8021PryChannelOutTable
--  ieee8021PryInTable
-- ============================================================================
--ieee8021PryIfTable
ieee8021PryIfTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PryIfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "A table with entries for each MAC Privacy protection capable interface in
    the system. Configured values of writable objects in table entries MUST be
    persistent and remain unchanged across re-initialization of the system's
    management entity."
    REFERENCE "20.14, "
    ::= { ieee8021PryMIBObjects 1 }

--ieee8021PryIfEntry
ieee8021PryIfEntry OBJECT-TYPE
    SYNTAX     Ieee8021PryIfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table entry with privacy controls for a particular PrY."
    INDEX { ieee8021PryIfIndex }
        ::= { ieee8021PryIfTable 1 }

--Ieee8021PryIfEntry
Ieee8021PryIfEntry ::=
   SEQUENCE {
        ieee8021PryIfIndex                 InterfaceIndex,
        ieee8021PryIfRxProtection          TruthValue,
        ieee8021PryIfTxProtection          TruthValue,
        ieee8021PryIfSecySupport           TruthValue,
        ieee8021PryIfAddr                  MacAddress,
        ieee8021PryIfMppduDA               MacAddress,
        ieee8021PryIfDefaultReassembly     TruthValue,
        ieee8021PryIfMaxPeers              Integer32,
        ieee8021PryIfNumPeers              Integer32
    }

--ieee8021PryIfIndex
ieee8021PryIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The Interface Index for this PrY's Private Port."
    REFERENCE   "20.14"
    ::= { ieee8021PryIfEntry 1 }
```

```
--ieee8021PryIfRxProtection
ieee8021PryIfRxProtection OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "When true(1), the PrY processes received MPPDUs addressed to
                 ieee8021PryAddr and ieee8021IfMppduDA (if that is a Group
                 address). When false(2) they are passed directly to the PrY's
                 Private Port. All other  MPPDUs are passed to the Private
                 Port, unprocessed, irrespective of this control's value."
    REFERENCE   "20.11"
    DEFVAL      { true }
    ::= { ieee8021PryIfEntry 2 }

--ieee8021PryIfTxProtection
ieee8021PryIfTxProtection OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "When true(1), the PrY protects transmitted user data frames
                 as configured in the Privacy Selection Table. When false(2),
                 all user data frames are passed directly to the PrY's
                 Controlled Port."
    REFERENCE   "20.5"
    DEFVAL      { true }
    ::= { ieee8021PryIfEntry 3 }

--ieee8021PryIfSecySupport
ieee8021PryIfSecySupport OBJECT-TYPE
  SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Set true(1) by the system if the PrY is directly supported by
                 a SecY and MKA, and false(2) otherwise. When true, the value
                 of ieee8021PryIfMppduDA and the entries in the PrY's peer
                 address table (ieee8021PryPeerTable) are determined by the
                 Key Agreement Entity (KaY) operating MKA, and are not
                 writable by network management."
    REFERENCE   "18.1, 20.11"
    DEFVAL      { true }
    ::= { ieee8021PryIfEntry 4 }

--ieee8021PryIfAddr
ieee8021PryIfAddr OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The individual MAC address associated with the PrY and other
                 components of the PrY's interface stack. Allocated by the
                 system. Used by PrY as the source address of MPPDUs and by a
                 supporting SecY (if present) for SCI assignment. The PrY
                 receives and processes MPPDUs with this destination address."
    REFERENCE   "18.1"
    ::= { ieee8021PryIfEntry 5 }

--ieee8021PryIfMppduDA
ieee8021PryIfMppduDA OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The destination MAC address used to transmit MPPDUs.
                 Also used to receive MPPDUs (if a Group address) when
                 ieee8021PryIfRxProtection is true. Set by the KaY if
                 ieee8021PryIfSecySupport is true, otherwise writable. If
                 ieee8021PryIfSecySupport transitions from true to false,
                 defaults to the Nearest non-TPMR Bridge Group address."
    REFERENCE   "18.1, 20.11"
    ::= { ieee8021PryIfEntry 6 }
```

```
--ieee8021PryIfDefaultReassembly
ieee8021PryIfDefaultReassembly OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Set true(1) by the system to indicate that the default
                 reassembly algorithm is used. Set false(2), otherwise. If
                 the system supports additional reassembly algorithms it shall
                 also support selection of the default algorithm. The maximum
                 size of the user data frame (DA, SA, MSDU) that can be
                 reassembled for delivery to the Private Port is the value of
                 ifMtu (as provided by the IF-MIB plus 22 octets."
    REFERENCE   "20.13, 20.13.1"
    ::= { ieee8021PryIfEntry 7 }

--ieee8021PryIfMaxPeers
ieee8021PryIfMaxPeers OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The maximum number of peer PrYs supported by the configured
                 reassembly algorithm."
    REFERENCE   "20.13"
    ::= { ieee8021PryIfEntry 8 }

--ieee8021PryIfNumPeers
ieee8021PryIfNumPeers OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of peer PrYs detected by the system."
    REFERENCE   "20.13"
    ::= { ieee8021PryIfEntry 9 }

-- ============================================================================
--ieee8021PrySelectionTable
ieee8021PrySelectionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PrySelectionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table for Privacy selection by transmit request priority"
    REFERENCE   "17.4, 17.4.3, 20.5"
    ::= { ieee8021PryMIBObjects 2 }

--ieee8021PrySelectionEntry
ieee8021PrySelectionEntry OBJECT-TYPE
    SYNTAX      Ieee8021PrySelectionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Privacy type selection for a given user priority"
    REFERENCE   "17.4.3, 20.5"
    INDEX { ieee8021PryIfIndex, ieee8021PrySelectionPriority }
      ::= { ieee8021PrySelectionTable 1 }

--Ieee8021PrySelectionEntry
Ieee8021PrySelectionEntry ::=
    SEQUENCE {
        ieee8021PrySelectionPriority    IEEE8021PriorityValue,
        ieee8021PrySelectionPrivacyType INTEGER
    }

--ieee8021PrySelectionPriority
ieee8021PrySelectionPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Transmit request user priority."
    REFERENCE   "17.4.3, 20.5"
    ::= { ieee8021PrySelectionEntry 1 }
```

```
--ieee8021PrySelectionPrivacyType
ieee8021PrySelectionPrivacyType OBJECT-TYPE
    SYNTAX      INTEGER {
                    none (1),
                    privacyFrame (2),
                    preemptableChannel (3),
                    expressChannel (4)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Privacy protection: none, privacyFrame, preemptableChannel,
                 or expressChannel."
    REFERENCE   "17.4.3, 20.5"
    ::= { ieee8021PrySelectionEntry 2 }

-- ============================================================================
--ieee8021PryFrameTable
ieee8021PryFrameTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PryFrameEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of Privacy Frame parameters for each possible value
                 of Controlled Port transmission priority (PrY user priority).
                 Each table entry can be configured even if Privacy Frame
                 transmission is not currently selected for user data frames
                 of that user priority."
    REFERENCE   "17.4.3, 20.7"
    ::= { ieee8021PryMIBObjects 3 }

--ieee8021PryFrameEntry
ieee8021PryFrameEntry OBJECT-TYPE
    SYNTAX      Ieee8021PryFrameEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Privacy Frame parameters for a given user priority."
    REFERENCE   "20.14.4"
    AUGMENTS { ieee8021PrySelectionEntry }
         ::= { ieee8021PryFrameTable 1 }

--Ieee8021PryFrameEntry
Ieee8021PryFrameEntry ::=
    SEQUENCE {
        ieee8021PryFrameAccessPriority   IEEE8021PriorityValue,
        ieee8021PryFrameRevealDE         TruthValue,
        ieee8021PryFramePadding          INTEGER
    }

--ieee8021PryFrameAccessPriority
ieee8021PryFrameAccessPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The Controlled Port priority (access priority) used to
                 transmit Privacy Frames with the Private Port transmission
                 priority (user priority) that selects this table entry."
    REFERENCE   "17.4.3, 20.7"
    ::= { ieee8021PryFrameEntry 1 }
```

```
--ieee8021PryFrameRevealDE
ieee8021PryFrameRevealDE OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "When true(1), the drop_eligible parameter value for
                 Controlled Port transmission of a Privacy Frame of the
                 specified user priority matches that used for the
                 encapsulated user data frame's Private Port transmit request.
                 Otherwise the Controlled Port transmit request has
                 drop_eligible false, and the DEI bit of a VLAN tag added as a
                 consequence of that transmit request (e.g., by the network
                 component of an EDE-CC) will be clear."
    REFERENCE   "20.7"
    ::= { ieee8021PryFrameEntry 2 }

--ieee8021PryFramePadding
ieee8021PryFramePadding OBJECT-TYPE
    SYNTAX      INTEGER {
                  one (1),
                  sixteen (16),
                  thirtyTwo(32),
                  sixtyFour(64)
                 }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Specifies padding of the Privacy Frame MPPDU (excluding
                 its source and destination MAC addresses) to four octets
                 (to allow for the MAC Privacy protection EtherType and the
                 MPPCI for an Encapsulated Frame) plus the nearest
                 multiple of one(1) (for no padding), sixteen(16),
                 thirtyTwo(32), or sixtyFour (64) octets. The specified size
                 excludes any octets to be added by supporting components
                 lower in the interface stack (e.g. a MACsec SecTAG and ICV,
                 and the Ethernet FCS) or other bridge components (e.g. an
                 outer VLAN tag added by an EDE's network component)."
    REFERENCE   "17.4.3, 20.7"
    ::= { ieee8021PryFrameEntry 3 }

-- ============================================================================
--ieee8021PryChannelTable
ieee8021PryChannelTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PryChannelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table with entries for the Express and the Preemptable
                 Privacy Channel's parameters."
    REFERENCE   "20.8, 20.9, 20.10"
    ::= { ieee8021PryMIBObjects 4 }

--ieee8021PryChannelEntry
ieee8021PryChannelEntry OBJECT-TYPE
    SYNTAX      Ieee8021PryChannelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Parameters for a given Privacy Channel."
    INDEX { ieee8021PryIfIndex, ieee8021PryChType }
        ::= { ieee8021PryChannelTable 1 }
```

```
--Ieee8021PryChannelEntry
Ieee8021PryChannelEntry ::=
    SEQUENCE {
        ieee8021PryChType              INTEGER,
        ieee8021PryChEnable            TruthValue,
        ieee8021PryChFragmentEnable    TruthValue,
        ieee8021PryChAccessPriority    IEEE8021PriorityValue,
        ieee8021PryChUserDataFrameSize Integer32,
        ieee8021PryChMppduGeneration   INTEGER,
        ieee8021PryChRequestedKbitRate      Unsigned32,
        ieee8021PryChMppduBitsOnWire   Unsigned32,
        ieee8021PryChMppduInterval     Unsigned32,
        ieee8021PryChUserBurstOctets   Unsigned32
    }

--ieee8021PryChType
ieee8021PryChType OBJECT-TYPE
    SYNTAX      INTEGER {
                    express (1),
                    preemptable (2)
                }
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Privacy Channel Type, express(1) or preemptable(2)."
    REFERENCE   "20.8"
    ::= { ieee8021PryChannelEntry 1 }

--ieee8021PryChEnable
ieee8021PryChEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "When true(1), user data frames assigned to this Privacy
                 Channel by a ieee8021PrySelectionEntry are transmitted using
                 this channel's parameters. When false(2), they are
                 transmitted using the other channel if ieee8021PryChEnable
                 is true for that channel and transmitted as Privacy Frames
                 using the relevant ieee8021PryFrameEntry otherwise."
    REFERENCE   "20.8"
    ::= { ieee8021PryChannelEntry 2 }

--ieee8021PryChFragmentEnable
ieee8021PryChFragmentEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "When true(1) permits user data frame fragmentation in this
                 Privacy Channel. Should be true, for bandwidth efficiency and
                 delay minimization. Provided to allow simple performance
                 testing and fragmentation benefit analysis."
    REFERENCE   "20.10"
    ::= { ieee8021PryChannelEntry 3 }

--ieee8021PryChAccessPriority
ieee8021PryChAccessPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The Controlled Port priority (access priority) used to
                 transmit MPPDUs for this Privacy Channel."
    REFERENCE   "20.8, 20.9.3"
    ::= { ieee8021PryChannelEntry 4 }
```

```
--ieee8021PryChUserDataFrameSize
ieee8021PryChUserDataFrameSize OBJECT-TYPE
    SYNTAX      Integer32 (128 .. 32768)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "User specification of MPPDU size. The largest user data
                 frame, at the Private Port interface (i.e. prior to MAC
                 Privacy protection)that can be transmitted as an MPPDU
                 Encapsulated Frame without fragmentation. Includes the user
                 data frame DA, SA, MSDU with EtherType, and a four octet FCS
                 allowance.  Excludes octets subsequently added by MACsec, or
                 other supporting interface stack components. Physical media,
                 and the configuration of other system components can impose
                 an upper bound lower than the configured value of this
                 parameter."
    REFERENCE   "20.9.3"
    ::= { ieee8021PryChannelEntry 5 }

--ieee8021PryChMppduGeneration
ieee8021PryChMppduGeneration OBJECT-TYPE
    SYNTAX      INTEGER { default (1), transmissionGate (2), other (3)}
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The MPPDU generation algorithm for this Privacy Channel. When
                 default(1), the (maximum) bandwidth is requested, with
                 a catch up (burst)parameter to recovers lost bandwidth if
                 an MPPDU transmission has been delayed by another frame sent
                 with higher access priority or by another component of the
                 same interface stack. When transmissionGate(2), MPPDU
                 transmission timing is gated, see 20.9.5 and IEEE Std 802.1Q."
    REFERENCE   "20.9, 20.9.4, 20.9.5"
    ::= { ieee8021PryChannelEntry 6 }

--ieee8021PryChRequestedKbitRate
ieee8021PryChRequestedKbitRate OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The physical medium bit rate (kilobits per second) to be
                 used by this Privacy Channel and the default MPPDU generation
                 algorithm in the absence of higher priority traffic or other
                 resource competition."
    REFERENCE   "20.9.4"
    ::= { ieee8021PryChannelEntry 7 }

--ieee8021PryChMppduBitsOnWire
ieee8021PryChMppduBitsOnWire OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of bit times required to transmit an MPPDU that
                 conveys a single, Private Port transmitted, user data frame
                 of ieee8021PryChUserDataFrameSize encoded as an Encapsulated
                 Frame(19.5.1). Calculated by the system, including all fields
                 added by the interface stack."
    REFERENCE   "20.9.4"
    ::= { ieee8021PryChannelEntry 8 }

--ieee8021PryChMppduInterval
ieee8021PryChMppduInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The approximate interval (as calculated by the system) in
                 nanoseconds between the transmission of MPPDUs for this
                 Privacy Channel, in the absence of competing higher priority
                 traffic or other resource competition.
"
    REFERENCE   "20.9.4"
    ::= { ieee8021PryChannelEntry 9 }
```

```
--ieee8021PryChUserBurstOctets
ieee8021PryChUserBurstOctets OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The number of additional user data frame burst for use by the
                 default MPPDU generation algorithm to recover channel
                 bandwidth lost to competing higher priority traffic."
    REFERENCE   "20.9.4"
    ::= { ieee8021PryChannelEntry 10 }

-- ============================================================================
--ieee8021PryPeerTable
ieee8021PryPeerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PryPeerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of peer PrYs. Frame Fragments received in MPPDUs with
                 source MAC addresses not in this table are discarded. When
                 ieee8021PryIfSecySupport is true, table entries are created
                 and deleted by the supporting Key Agreement Entity. When false
                 the system automatically creates an entry for
                 ieee8021PryIfMppduDA if that is not a Group address, and other
                 entries can be created by management."
    REFERENCE   "20.13"
    ::= { ieee8021PryMIBObjects 5 }

--ieee8021PryPeerEntry
ieee8021PryPeerEntry OBJECT-TYPE
    SYNTAX     Ieee8021PryPeerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the peer PrY table for one peer."
    INDEX { ieee8021PryIfIndex, ieee8021PryPeerAddr }
    ::= { ieee8021PryPeerTable 1 }

--Ieee8021PryPeerEntry
Ieee8021PryPeerEntry ::=
    SEQUENCE {
        ieee8021PryPeerAddr       MacAddress,
        ieee8021PryPeerRowStatus  RowStatus
    }

--ieee8021PryPeerAddr
ieee8021PryPeerAddr OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "MAC address of a peer PrY."
    REFERENCE   "20.13"
    ::= { ieee8021PryPeerEntry 1 }

--ieee8021PryPeerRowStatus
ieee8021PryPeerRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "Indicates the status of an entry in the ieee8021PryPeerTable,
                 can be used to create entries if ieee8021PryIfSecySupport
                 is false."
    REFERENCE   "20.13"
    ::= { ieee8021PryPeerEntry 2 }
```

```
-- ============================================================================
--ieee8021PryOutTable
ieee8021PryOutTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PryOutEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Transmission statistics for Private Port user data frames
                 transmitted as Privacy Frames or unprotected."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryMIBObjects 6 }

--ieee8021PryOutEntry
ieee8021PryOutEntry OBJECT-TYPE
    SYNTAX      Ieee8021PryOutEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Transmission statistics, for frames not in a Privacy Channel."
    AUGMENTS { ieee8021PryIfEntry }
        ::= { ieee8021PryOutTable 1 }

--Ieee8021PryOutEntry
Ieee8021PryOutEntry ::=
    SEQUENCE {
        ieee8021PryOutPrivacyFrames Counter64,
        ieee8021PryOutPfUserOctets  Counter64,
        ieee8021PryOutPfPadOctets   Counter64,
        ieee8021PryOutUnprtFrames   Counter64,
        ieee8021PryOutUnprtOctets   Counter64
    }

--ieee8021PryOutPrivacyFrames
ieee8021PryOutPrivacyFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of user data frames sent as Privacy Frames (each
                 in a separate MPPDU)."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryOutEntry 1 }

--ieee8021PryOutPfUserOctets
ieee8021PryOutPfUserOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of user data octets sent in Privacy Frames (each
                 user data frame in a separate MPPDU). Not counting pad octets."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryOutEntry 2 }

--ieee8021PryOutPfPadOctets
ieee8021PryOutPfPadOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of pad octets sent in Privacy Frames (each
                 conveying a single Private Port user data frame)."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryOutEntry 3 }

--ieee8021PryOutUnprtFrames
ieee8021PryOutUnprtFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of MAC Privacy PDUs sent."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryOutEntry 4 }
```

```
--ieee8021PryOutUnprtOctets
ieee8021PryOutUnprtOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of MAC Privacy PDUs sent."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryOutEntry 5 }

-- ============================================================================
--ieee8021PryChannelOutTable
ieee8021PryChannelOutTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PryChannelOutEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Transmission statistics for the Express and Preemptable
                 Privacy Channels."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryMIBObjects 7 }

--ieee8021PryChannelOutEntry
ieee8021PryChannelOutEntry OBJECT-TYPE
    SYNTAX      Ieee8021PryChannelOutEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Transmission statistics for a Privacy Channel."
    AUGMENTS { ieee8021PryChannelEntry }
        ::= { ieee8021PryChannelOutTable 1 }

--Ieee8021PryChannelOutEntry
Ieee8021PryChannelOutEntry ::=
    SEQUENCE {
        ieee8021PryChOutUserFrames      Counter64,
        ieee8021PryChOutUserOctets      Counter64,
        ieee8021PryChOutPadOctets       Counter64,
        ieee8021PryChOutMppdus          Counter64,
        ieee8021PryChOutEncapFrames     Counter64,
        ieee8021PryChOutExpFragments    Counter64,
        ieee8021PryChOutPreFragments    Counter64
    }

--ieee8021PryChOutUserFrames
ieee8021PryChOutUserFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of user data frames sent in this Privacy Channel."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryChannelOutEntry 1 }

--ieee8021PryChOutUserOctets
ieee8021PryChOutUserOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of user data octets sent in this Privacy Channel.
                 Not counting pad octets."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryChannelOutEntry 2 }

--ieee8021PryChOutPadOctets
ieee8021PryChOutPadOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of pad octets sent in this Privacy Channel."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryChannelOutEntry 3 }
```

```
--ieee8021PryChOutMppdus
ieee8021PryChOutMppdus OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of MPPDUs sent in this Privacy Channel."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryChannelOutEntry 4 }

--ieee8021PryChOutEncapFrames
ieee8021PryChOutEncapFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of Encapsulated Frames encoded for this Privacy
                 Channel."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryChannelOutEntry 5 }

--ieee8021PryChOutExpFragments
ieee8021PryChOutExpFragments OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of Express Fragments encoded for this Privacy Channel."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryChannelOutEntry 6 }

--ieee8021PryChOutPreFragments
ieee8021PryChOutPreFragments OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of Preemptable Fragments encoded for this Privacy
                 Channel."
    REFERENCE   "20.14.1"
    ::= { ieee8021PryChannelOutEntry 7 }

-- ============================================================================
--ieee8021PryInTable
ieee8021PryInTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PryInEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Reception statistics for all user data frames, MPPDUs, their
                 components,and fragmented user data frame reassembly."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryMIBObjects 8 }

--ieee8021PryInEntry
ieee8021PryInEntry OBJECT-TYPE
    SYNTAX      Ieee8021PryInEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Transmission statistics, for frames not in a Privacy Channel."
    AUGMENTS { ieee8021PryIfEntry }
        ::= { ieee8021PryInTable 1 }
```

```
--Ieee8021PryInEntry
Ieee8021PryInEntry ::=
    SEQUENCE {
        ieee8021PryInUserFrames       Counter64,
        ieee8021PryInUserOctets       Counter64,
        ieee8021PryInPadOctets        Counter64,
        ieee8021PryInMppdus           Counter64,
        ieee8021PryInEncapFrames      Counter64,
        ieee8021PryInExpFragments     Counter64,
        ieee8021PryInPreFragments     Counter64,
        ieee8021PryInExpDiscards      Counter64,
        ieee8021PryInPreDiscards      Counter64,
        ieee8021PryInUnknownMppcis     Counter64,
        ieee8021PryInErroredMppdus      Counter64,
        ieee8021PryInUnprtFrames      Counter64,
        ieee8021PryInUnprtOctets      Counter64
    }

--ieee8021PryInUserFrames
ieee8021PryInUserFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of protected user data frames received in
                 MPPDUs, encoded as Encapsulated Frames or reassembled from
                 Frame Fragments."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 1 }

--ieee8021PryInUserOctets
ieee8021PryInUserOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of user data frame octets received. Excludes
                 padding."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 2 }

--ieee8021PryInPadOctets
ieee8021PryInPadOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of pad octets received in MPPDUs."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 3 }

--ieee8021PryInMppdus
ieee8021PryInMppdus OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of MAC Privacy protection PDUs received."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 4 }

--ieee8021PryInEncapFrames
ieee8021PryInEncapFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of Encapsulated Frame components received in MPPDUs."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 5 }
```

```
--ieee8021PryInExpFragments
ieee8021PryInExpFragments OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of correctly encoded Express Frame Fragments
                 received in MPPDUs. Includes fragments discarded by
                 reassembly (unknown peer, too many peers, out of order,
                 reassembled frame too large)."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 6 }

--ieee8021PryInPreFragments
ieee8021PryInPreFragments OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of correctly encoded Preemptable Frame Fragments
                 received in MPPDUs. Includes fragments discarded by
                 reassembly (unknown peer, too many peers, out of order,
                 reassembled frame too large)."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 7 }

--ieee8021PryInExpDiscards
ieee8021PryInExpDiscards OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of Express Frame Fragment discard events (discarding a
                 fragment and/or a partially reassembled user data frame)."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 8 }

--ieee8021PryInPreDiscards
ieee8021PryInPreDiscards OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of Preemptable Frame Fragment discard events
                 (discarding a fragment and/or a partially reassembled
                 user data frame)."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 9 }

--ieee8021PryInUnknownMppcis
ieee8021PryInUnknownMppcis OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of of unknown MPPDU components received."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 10 }

--ieee8021PryInErroredMppdus
ieee8021PryInErroredMppdus OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Number of received MPPDUs containing an incorrectly
                 encoded component."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 11 }
```

```
--ieee8021PryInUnprtFrames
ieee8021PryInUnprtFrames OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of MAC Privacy user frames received."
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 12 }

--ieee8021PryInUnprtOctets
ieee8021PryInUnprtOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Total number of MAC Privacy user octets received. Does not
                 include padding"
    REFERENCE   "20.14.2"
    ::= { ieee8021PryInEntry 13 }

-- ============================================================================
--ieee8021PryMIBConformance

ieee8021PryMIBCompliances  OBJECT IDENTIFIER ::= { ieee8021PryMIBConformance 1 }
ieee8021PryMIBGroups       OBJECT IDENTIFIER ::= { ieee8021PryMIBConformance 2 }

-- ============================================================================
--ieee8021PryMIBCompliances

--ieee8021PryMIBCompliance
ieee8021PryMIBCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION "IEEE8021-PRY-MIB compliance statement, IEEE Std 802.1AEdk."
    MODULE  -- this module
        MANDATORY-GROUPS {
            ieee8021PryIfGroup,
            ieee8021PrySelectionGroup,
            ieee8021PryFrameGroup,
            ieee8021PryChannelGroup,
            ieee8021PryPeerGroup,
            ieee8021PryOutGroup,
            ieee8021PryChOutGroup,
            ieee8021PryInGroup
        }
    ::= { ieee8021PryMIBCompliances 1 }

-- ============================================================================
--ieee8021PryMIBGroups
--ieee8021PryIfGroup
ieee8021PryIfGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PryIfRxProtection,
        ieee8021PryIfTxProtection,
        ieee8021PryIfSecySupport,
        ieee8021PryIfAddr,
        ieee8021PryIfMppduDA,
        ieee8021PryIfDefaultReassembly,
        ieee8021PryIfMaxPeers,
        ieee8021PryIfNumPeers,
        ieee8021PryIfNumPeers
    }
    STATUS      current
    DESCRIPTION "PrY service management (mandatory ieee8021PryIfTable objects)."
    ::= { ieee8021PryMIBGroups 1 }
```

```
--ieee8021PrySelectionGroup
ieee8021PrySelectionGroup    OBJECT-GROUP
    OBJECTS {
        ieee8021PrySelectionPrivacyType
    }
    STATUS      current
    DESCRIPTION "Privacy type selection, by Private Port transmitted frame
                 priority."
    ::= { ieee8021PryMIBGroups 2 }

--ieee8021PryFrameGroup
ieee8021PryFrameGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PryFrameAccessPriority,
        ieee8021PryFrameRevealDE,
        ieee8021PryFramePadding
    }
    STATUS      current
    DESCRIPTION "Parameters for individual Privacy Frame transmission."
    ::= { ieee8021PryMIBGroups 3 }

--ieee8021PryChannelGroup
ieee8021PryChannelGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PryChEnable,
        ieee8021PryChFragmentEnable,
        ieee8021PryChAccessPriority,
        ieee8021PryChUserDataFrameSize,
        ieee8021PryChMppduGeneration,
        ieee8021PryChRequestedKbitRate,
        ieee8021PryChMppduBitsOnWire,
        ieee8021PryChMppduInterval,
        ieee8021PryChUserBurstOctets
    }
    STATUS      current
    DESCRIPTION "PrY service management Group"
    ::= { ieee8021PryMIBGroups 4 }

--ieee8021PryPeerGroup
ieee8021PryPeerGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PryPeerRowStatus
    }
    STATUS      current
    DESCRIPTION "PrY peer addresses."
    ::= { ieee8021PryMIBGroups 5 }

--ieee8021PryOutGroup
ieee8021PryOutGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PryOutPrivacyFrames,
        ieee8021PryOutPfUserOctets,
        ieee8021PryOutPfPadOctets,
        ieee8021PryOutUnprtFrames,
        ieee8021PryOutUnprtOctets
    }
    STATUS      current
    DESCRIPTION "PrY service management Group"
    ::= { ieee8021PryMIBGroups 6 }
```

```
--ieee8021PryChOutGroup
ieee8021PryChOutGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PryChOutUserFrames,
        ieee8021PryChOutUserOctets,
        ieee8021PryChOutPadOctets,
        ieee8021PryChOutMppdus,
        ieee8021PryChOutEncapFrames,
        ieee8021PryChOutExpFragments,
        ieee8021PryChOutPreFragments
    }
    STATUS      current
    DESCRIPTION "PrY service management Group"
    ::= { ieee8021PryMIBGroups 7 }

--ieee8021PryInGroup
ieee8021PryInGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PryInUserFrames,
        ieee8021PryInUserOctets,
        ieee8021PryInPadOctets,
        ieee8021PryInMppdus,
        ieee8021PryInEncapFrames,
        ieee8021PryInExpFragments,
        ieee8021PryInPreFragments,
        ieee8021PryInExpDiscards,
        ieee8021PryInPreDiscards,
        ieee8021PryInUnknownMppcis,
        ieee8021PryInErroredMppdus,
        ieee8021PryInUnprtFrames,
        ieee8021PryInUnprtOctets
    }
    STATUS      current
    DESCRIPTION "PrY service management Group"
    ::= { ieee8021PryMIBGroups 8 }

END
```

Insert the following text (Clause 23) after Clause 22:

23. YANG Data Models

This clause specifies YANG (IETF RFC 7950) data models that provide control and monitoring of systems and system components that implement functionality specified in this standard. These data models are based on the managed objects and their functionality specified in Clause 10 and Clause 20, and their use as specified in Clause 11, Clause 15, and Clause 21. The specifications in those clauses take precedence if there is any discrepancy with the text of this clause (Clause 23).

This clause

a) Introduces the YANG framework that governs the naming and hierarchy of configuration and operational data structures in the data models, and the modeling of network interfaces (23.1).

b) Describes each data model and its relationship to other YANG data models, and to the operational processes and managed objects specified in the other clauses of this standard (23.2, 23.4, 23.8).

c) Reviews security considerations particular to each of the data models, with specific reference to data nodes in the YANG modules that compose the model (23.3, 23.5, 23.9).

d) Includes a schema tree diagram for each YANG module (23.10).

e) Includes each YANG module (23.11).

YANG data models are specified for the addition of MAC Security (23.2) and MAC Privacy protection (23.4) capabilities to general interface stacks. Both explicit and augmented interface stack models are supported, and relevant modeling considerations described (23.6). These general interface models are used in models for EDEs (23.8.2, 23.8.3, 23.8.4, 23.8.5) and to support the addition of MAC Security and MAC Privacy protection to other systems as described in 23.8.1, Clause 11 and Clause 21.

The MIB modules specified in Clause 13 and Clause 22 were similarly derived from the other clauses of this standard. Consequently, the capabilities and structure of the YANG data models are closely aligned with those MIB modules. However, the YANG data modules were not derived from the MIB modules and do not include data or modeling constructs particular to MIB module specification and not in the information model. A system may support MACsec and MAC Privacy protection management using both MIB and YANG modules. However any given system is expected to be managed using either YANG or SNMP, rather than a combination of the two.

The YANG modules defined in this clause are designed to be accessed via a network configuration protocol, e.g., NETCONF protocol (IETF RFC 6241 [B12]). In the case of NETCONF, the lowest NETCONF layer is the secure transport layer and the mandatory to implement secure transport is SSH (IETF RFC 6242 [B13]). The NETCONF access control model (IETF RFC 6536 [B14]) provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content. It is the responsibility of a system’s implementor and administrator to ensure that the protocol entities in the system that support NETCONF, and any other remote configuration protocols that make use of these YANG modules, are properly configured to allow access only to those principals (users) that have legitimate rights to read or write data nodes. This standard does not specify how the credentials of those users are to be stored or validated.

23.1 YANG Framework

This clause has been developed in accordance with the YANG guidelines published in IETF RFC 6087 [B11] as applicable to IEEE standards.

The YANG framework applies hierarchy in the following areas:

1) The uniform resource name (URN), as specified in IEEE Std 802d-2017.
2) The YANG objects form a hierarchy of configuration and operational data structures that define the YANG model.

Figure 23-1 outlines the organization, within the YANG hierarchy, of data models and objects that support and complement data models supported by this standard.

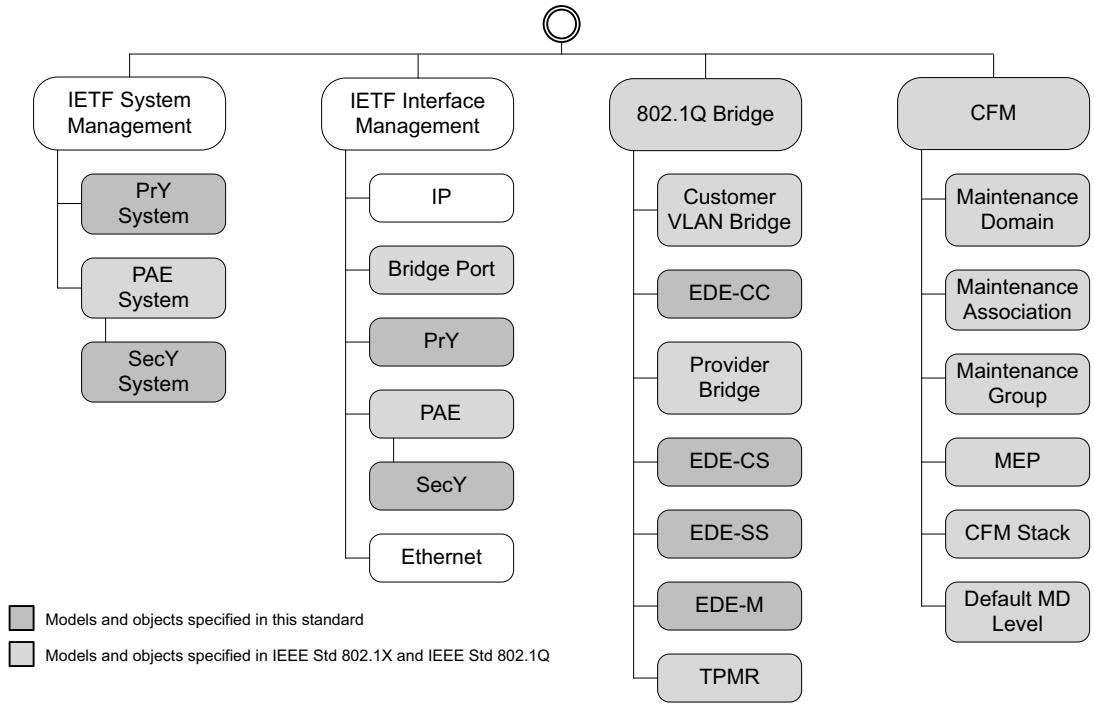


Figure 23-1—YANG hierarchy, models and objects

Figure 23-1 is not intended to be comprehensive. It shows data models and objects specified or referenced by this standard. The YANG modules specified augment both the IETF System Management YANG data model (IETF RFC 7317), with a hierarchy of nodes under **/ietf-system:system**, and the IETF Interface Management YANG data model (IETF RFC 8343), with a hierarchy of nodes under **/ietf-interfaces:interfaces/ietf-interfaces:interface**. EDE specifications make use of Bridge components specified in IEEE Std 802.1Q, with YANG data nodes under **/ieee:std:802.1Q:yang:ieee802-dot1q-bridge**.

NOTE—The prefixes **if** and **sys** are used to abbreviate **/ietf-interfaces:interfaces/ietf-interfaces:interface** to **/if:interfaces/if:interface** and **/ietf-system:system** to **/sys:system** in the YANG modules.

IEEE Std 802.1Q and IEEE Std 802.1X provide additional information on the IETF System Management and Interface Management models and their relationship to data models specified by IEEE 802.1 standards.

23.2 MAC Security Entity (SecY) model

The SecY YANG data model is based on Figure 10-5 [14], which summarizes the information model used by the SecY management process (10.7). Each SecY supports port-based network access control for a particular port and is associated with a Port Access Entity (PAE, IEEE Std 802.1X) which includes the SecY's Key Agreement Entity (KaY) as described in 7.1. The SecY model augments the PAE YANG data model, specified in IEEE Std 802.1X, with two primary YANG sub-trees. The **secy-system** sub-tree augments **pae-system**, which augments /**sys:system**. The **secy** sub-tree augments **pae**, which augments **/if:interfaces/if:interface**.

The SecY data model supports explicit and augmented interface stacks, including interfaces that include an augmentation for a protocol shim above a SecY. When the SecY is the only or the top-most protocol shim in an interface, nodes in the basic IETF interface management model (in **/if:interfaces/if:interface**) can duplicate or summarize information from the **secy** sub-tree (see 23.2.2, 23.6).

The PAE YANG data model includes nodes that provide information about or control the operation of MACsec capable interfaces (see 23.2.3). The **secy-system** and **secy** sub-trees do not duplicate this information.

Figure 23-2 shows the **secy-system** and **pae-system** nodes that support the SecY model, and references to those nodes from **secy** and **pae** interface nodes, as follows:

— **pae-system/system-access-control** allows a network manager to enable or disable port-based network access control for a system as a whole. When disabled this control overrides **secy/verification/validate-frames** and **secy/generation/protect-fram**es causing all SecYs to behave as if it were 'null'.

— **secy-system/cipher-suites** lists Cipher Suites that might be used by the system, and their attributes. The Cipher Suites that a given SecY supports are referenced by **secy/cipher-suite-control**, and that currently used by **secy/current-ciphersuite**.

— **pae-system/nid-group** lists the NIDs supported by the system. On system initialization and prior to (or in the absence of) supporting authentication and authorization protocol, the use of MACsec on a particular interface can be controlled by a **pae/logon-nid.selected** reference to a particular NID (Network Identity). The scope of the NID's identifier (**pae-system/nid-group.nid**) is that of the **/sys:system,** so a simple system can use a null NID for a configuration applicable to all MACsec capable interfaces, while a system that naturally has ports of particular types (such as an EDE-CC) can use a local convention to identify the configuration for ports of that type. Equally a network manager can use a network wide NID naming scheme.

— **pae-system** also contains EAPOL and MKA version information.

NOTE—Objects in the SecY YANG model are named using the normal YANG lower case convention with hyphens separating words or acronyms. These names correspond simply to those in the Clause 10 protocol independent description, which use 'camelCase' (capitalizing the start of each word other than the first and omitting hyphens).

Figure 23-3 shows the **secy**, **/if:interfaces/if:interface** and **pae** nodes that support the SecY model and their mutual dependencies, as follows:

— **pae/port-capabilities/macsec** and **pae/port-capabilities/mka** are true if an interface supports MACsec and MKA respectively. A PAE can, but does not necessarily, have other capabilities (such as support for authentication using EAP).

[14] Figure 10-5 and figures in this clause (Clause 23) use UML [B1] conventions and C++ language constructs to present information in a common compact way, independent of its inclusion in MIB modules, YANG data models, or operational use within a system. These figures are not intended to support automated object encoding, and are simplified to allow significant information and relationships to be presented in a single figure, reducing the need to temporarily memorize detail when reading the standard.

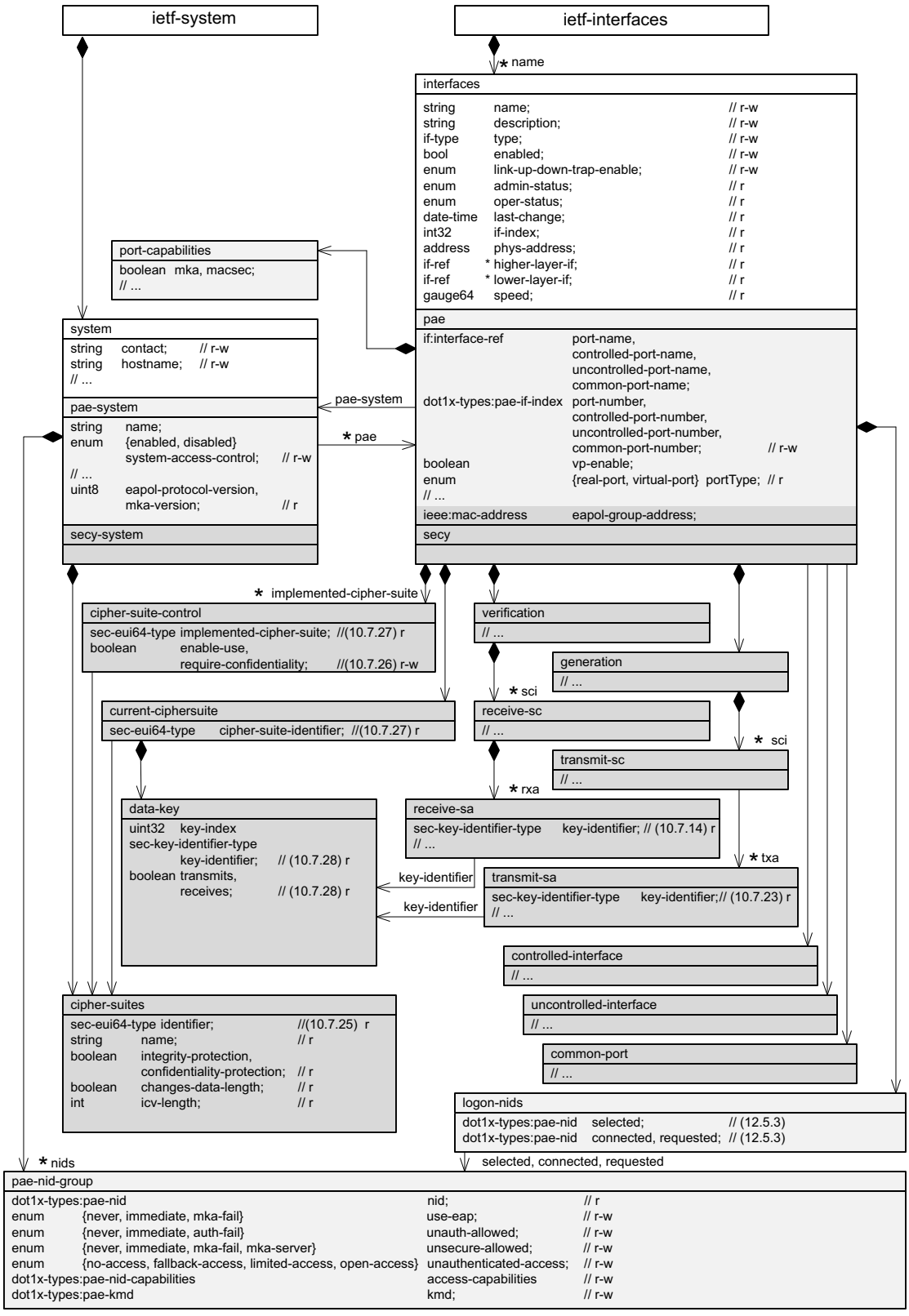
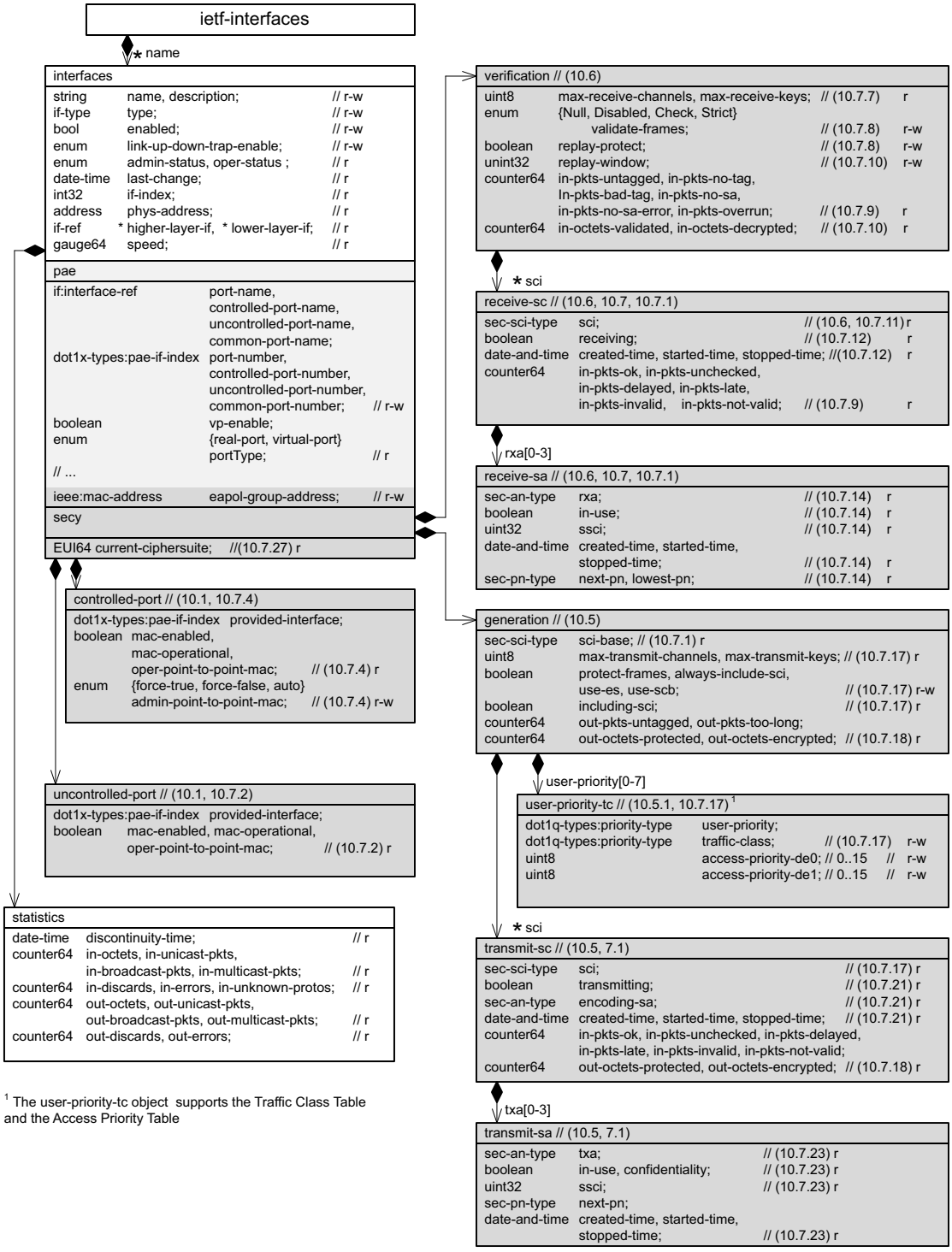


Figure 23-2—SecY model system nodes and references



[1] The user-priority-tc object supports the Traffic Class Table and the Access Priority Table

Figure 23-3—SecY model interface nodes

The relationship between the SecY data model and the IETF System Management, IETF Interface Management, and PAE models is further detailed in 23.2.2, 23.4, and 23.6.

The SecY data model does not include objects and operations from Figure 10-5 that are only made available by the SecY to other components (notably the Key Agreement Entity, KaY) within the same system. These inaccessible items include cryptographic keys (see 23.3) and their purely local key identifiers, controls over individual receive SCs and SAs [enableReceive, updtNextPN, updtLowestPN(10.7.15)], and controls over individual transmit SCs and SAs.

23.2.1 Relationship to the IETF System Management model

IEEE Std 802.1X describes the relationship between the IETF System Management model and the **pae-system**, which augments **/sys:system**. The same considerations that also apply to the **secy-system** which augments the **pae-system**. In particular a system that implements the SecY YANG data model shall also implement the IETF System Management data model defined in IETF RFC 7317).

23.2.2 Relationship to the IETF Interface Management model

IEEE Std 802.1X describes the relationship between the IETF Interface Management model and the **secy** sub-tree augments **pae**, which augments **/if:interfaces/if:interface**. The same considerations also apply to the **secy-system** that augments the **pae-system**. In particular, a system that implements the SecY YANG data model shall also implement the IETF Interface Management data model defined in IETF RFC 8343 for each interface that is MACsec capable.

IEEE Std 802.1X further describes both explicit and augmented interface stack models with both PAE and SecY functionality. An explicitly modeled interface represents a single shim or sublayer in an interface stack. An augmented interface combines adjacent shims or sublayers into a single interface to simplify configuration. In both cases the **/if:interfaces/if:interface/statistics** in-octets, in-unicast-pkt, in-broadcast-pkts, in-multicast-pkts, out-octets, out-unicast-pkt, out-broadcast-pkts, and out-multicast-pkts, reflect the contents of frames transmitted and received by the user(s) of the uppermost shim or sublayer, which might or might not be the SecY. The SecY's Controlled Port interface statistics are always available from the **secy/controlled-interface** container. Statistics for the SecY's Common Port are recorded by shim or interface directly supporting that port, and are not separately recorded.

In both explicit and augmented interface stacks, a SecY and its associated PAE augment the same interface, as described in IEEE Std 802.1X and as a consequence of the **secy** node augmenting its associated **pae** node. The PAE uses the SecY's Uncontrolled Port to transmit and receive EAPOL PDUs. Detailed statistics for those EAPOL frames are provided by the **pae/eapol-statistics** container, and are not otherwise broken down by unicast, broadcast, and multicast or recorded by the SecY model.

Unless otherwise specified, and unlike a PAE, a protocol entity that is represented by an augment or reference (e.g., in the case of a CFM MIP or MEP shim) to an interface that is augmented by a SecY uses the service provided by the SecY's Controlled Port (possibly indirectly if the interface has been augmented by a number of shims). An instance of LLDP (for example), when represented by and configured as an augment to the same interface as a SecY, transmits frames that are protected and validated by that SecY. Interface stack configurations are described in 23.6.

To assist in the isolation and diagnosis of connectivity issues, the current status of MAC_Enabled and MAC_Operational (6.4) are reported for the SecY's Controlled Port and Common Port in **secy/controlled-interface** and **secy/common-interface**. The value of MAC_Enabled for a port (interface) reflects the state of all the administrative controls for sublayers in the interface that support each port. The values of MAC_Operational and MAC_Enabled for the Uncontrolled Port are the same as those for the Common Port.

23.2.3 Relationship to the PAE model

The **secy-system** and **secy** nodes augment the PAE model as described above (23.2, 23.2.2). A system that implements the SecY YANG data model shall also implement the PAE model defined in IEEE Std 802.1X. The **pae-system** and **pae** node objects implemented as part of that model can be as simple as those required to identify particular interfaces as MACsec capable, and to support MKA operation with pre-shared CAKs. The PAE model can also support EAP operation (as a Supplicant or an Authenticator, or both) and the transmission and reception of announcements. IEEE Std 802.1X further describes the relationship of the PAE model to other models for authentication, authorization, and the use of credentials.

23.2.4 Structure of the SecY model

The SecY model comprises the **ieee802-dot1ae-secy** (23.11.1) and **ieee802-dot1x-eapol** (23.11.3) YANG modules specified in this standard, the **ieee802-dot1x** YANG module specified in IEEE Std 802.1X, the **ietf-system** and **ietf-interfaces** YANG modules, and the types modules that they import.

NOTE—The ieee802-dot1x-eapol module is specified in this standard to provide access the EAPOL address used by a PAE. A future revision of the ieee802-dot1x module could add that missing functionality.

23.3 Security considerations for the SecY model

A SecY exists to secure communication, providing connectionless user data confidentiality, frame data integrity, and data origin authenticity (1.2, Clauses 6, 7, and 8). All the data nodes in the SecY model and supporting data nodes in the PAE model that are writable/creatable/deletable (i.e., config true) exist to control the way that the SecY does (or does not) secure communication and need to be considered to be sensitive in all network environments. In particular, config access can either compromise security directly or disrupt network operation as part of an effort to mislead a network administrator into compromising security, as follows:

- **pae-system/system-access-control** changes can disable the use of MACsec.
- **pae-system/pae-nid-group** and/or **pae/logon-nids** changes can disable or constrain the use of MACsec on particular ports.
- **pae.eapol-group-address** change can cause MKA operation to fail, preventing the use of a MACsec protected link.
- **secy/verification/validate-frames** and **secy/generation/protect-frames** changes can disable MACsec protection and validation of received frames.
- **secy/verification/replay-window** increases can render the network vulnerable to overload from repeated frames while decreases can reduce performance on links that reorder frames.

An implementation can remove or restrict access to these nodes (e.g., by requiring direct access to the system rather than through a network configuration protocol). Read access to these nodes can also allow an adversary to choose indirect attacks.

The SecY YANG model does not provide access to cryptographic keys, authentication credentials, and authorization parameters that would allow an adversary to introduce apparently valid frames on a secured network link.

Access to data nodes that are specified as readable (i.e., config false) in the SecY model does not elicit information that an adversary can use directly to compromise network operation. In principle that information could be gathered by an adversary that has direct access to the MACsec protected link. However, access using a network configuration protocol can provide that information to a remote adversary.

23.4 MAC Privacy protection (PrY) model

The PrY YANG data model is based on Figure 20-6, which summarizes the information model used by the PrY management process (20.14). Each PrY provides MAC Privacy protection for a particular port. The PrY model augments the IETF Interface Management model: the **pry** sub-tree augments **pae**, which augments **/if:interfaces/if:interface**.

Each PrY is supported by a SecY [or by an equivalent protocol entity providing connectionless user data confidentiality, frame data integrity, and data integrity for MAC Clients (1.2)]. The supporting SecY can be part of the same interface stack or located in an adjacent system and connected by a physically secure connection. When the SecY is part of the same interface stack, the PrY is modeled as part of (i.e., an augment of) the same interface and supported directly by the SecY, i.e., the PrY's Controlled Port is the SecY's Controlled Port (20.1, Figure 20-1).

Figure 23-4 shows the **pry** nodes augmenting the **interface** nodes, including the following:

- **pry/reception/privacy-protection** and **pry/transmission/privacy-protection** allow a network manager to enable MAC privacy protection on reception and transmission independently.
- **pry/if-secy-support** is true if the PrY is directly supported by a SecY.
- **pry/pry-address** is the address the PrY uses as the source address of MPPDUs, **pry/pry-mppdu-dest-address** is the destination address used for transmitted MPPDU, and **pry/peer-entry*peer-src-address** lists the MAC addresses of the PrY's peers (MPPDUs received with unrecognized addresses are discarded). If **pry/if-secy-support** is true these addresses are supplied by MKA and are not configurable in the data model.
- **pry/transmission/privacy-selection** provides per-user priority control (privacy-type) of the way MAC Privacy protection is applied to any given user data frame on transmission (by encapsulation in a Privacy Channel, by transmission as an individual Privacy Frame, or not at all). For Privacy Frame transmission it also provides control over the access priority, whether the user data frame DE bit value is revealed, and the use of padding.
- **pry/transmission/channel*channel-id** allows each Privacy Channel (Express, Preemptable) and its use of user data fragmentation to be enabled or disabled, and controls Privacy Channel parameters (MPPDU size, bandwidth utilization).
- **pry/transmission/frame-tx-statistics**, **pry/transmission/channel*channel-id/tx-statistics**, and **pry/reception/rx-statistics** provide counts of transmitted and received user data frames, user data octets, padding octets, unprotected frames, MPPDUs, and user data frame fragments.

23.4.1 Relationship to the IETF System Management model

The IETF System Management model defines the scope of the if-index parameter values that identify each interface that is augmented by a **pry**. The PrY model is not otherwise dependent on **/sys:system**.

23.4.2 Relationship to the IETF Interface Management model

The **pry** sub-tree augments **/if:interfaces/if:interface**. Unless otherwise specified, and unlike a PAE, a protocol entity that is represented by an augment or reference to an interface that is augmented by a PrY, or both a PrY and a SecY, uses the service provided by the PrY's Private Port (possibly indirectly if the interface has been augmented by a number of shims). Interface stack configurations are described in 23.6.

23.4.3 Relationship to the PAE and SecY models

If the PrY is not in the same system as its supporting SecY, the PrY model is independent of PAE and SecY models, or rather it is the responsibility of the network administrator to ensure compatible configuration of the two systems. From a practical point of view this means that **pry/transmission/privacy-protection** and **pry/reception/privacy/protection** can only be enabled if the connectivity provided by the SecY is protected

by MACsec. The network administrator also has to choose and configure appropriate PrY source MAC address, PrY peer source MAC addresses, and MPPDU destination MAC addresses.

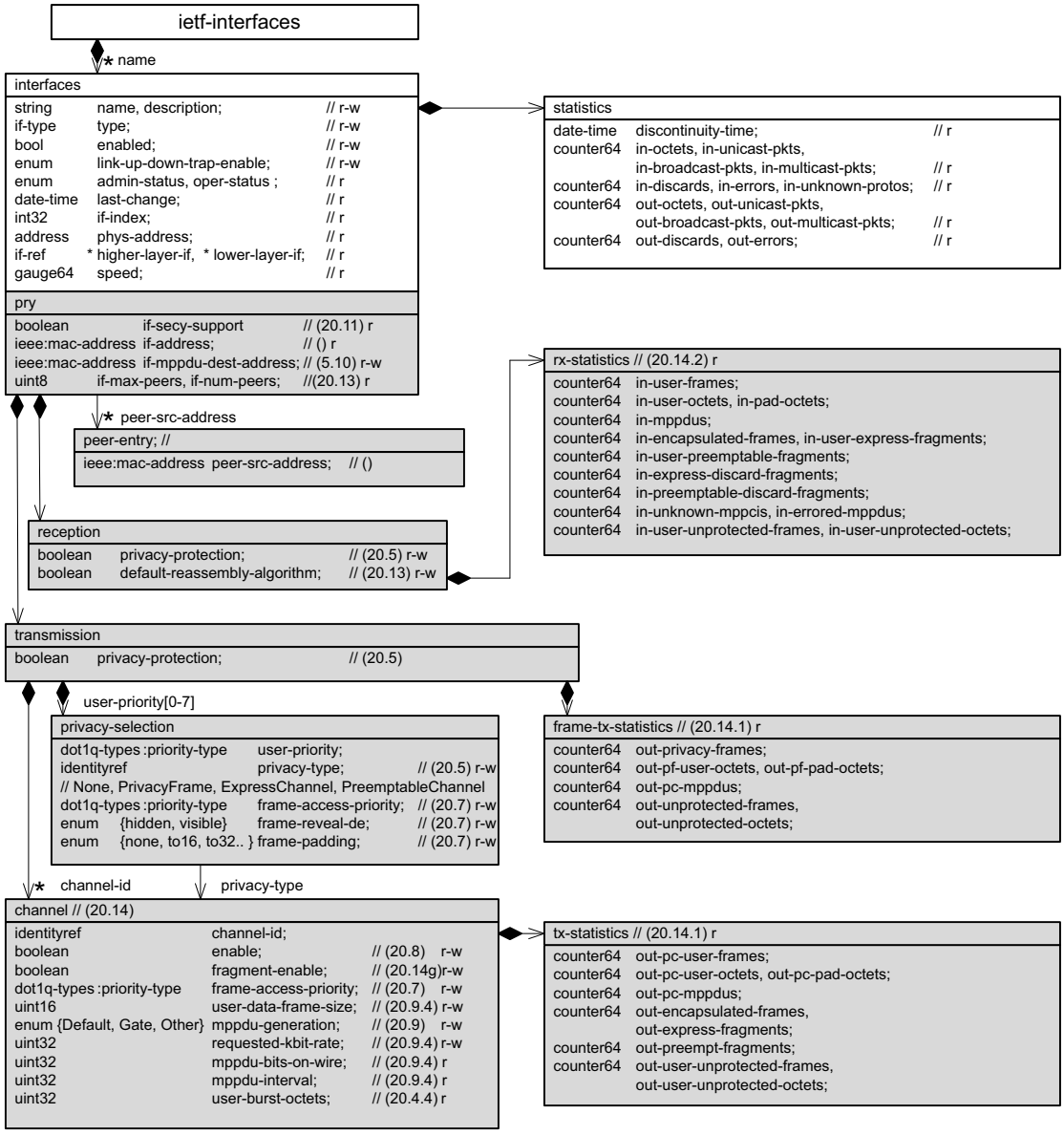


Figure 23-4—PrY model interface nodes

If the PrY is in the same system (and thus in the same interface) as its supporting SecY, those MAC addresses are automatically configured by the SecY's supporting MKA instance (in the PAE model) and are config false in the PrY model. MAC Privacy protection (on transmission and reception) is only enabled if the SecY is protecting transmitted frames, **i.e., secy/generation.protect-frames** is true, and is validating all received frames, **i.e., secy/verification/validate-frames** is Strict. Changes to the use of MAC Privacy protection resulting from changes to MACsec protection of transmitted and received frames do not change the configured values of **pry/reception/privacy-protection** and **pry/transmission/privacy-protection**.

23.4.4 Structure of the PryY model

The PrY model comprises the **ieee802-dot1ae-pry** (23.11.2) YANG module specified in this standard, the **ieee802-dot1x** YANG module specified in IEEE Std 802.1X, the **ietf-system** and **ietf-interfaces** YANG modules, and the types modules that they import.

23.5 Security considerations for the PrY model

A PrY, when used in conjunction with MACsec, reduces the ability of adversaries to correlate the MAC addresses, sizes, and timing of user data frames with users and their use of the network (1.1). MACsec is responsible for frame data confidentiality and integrity protection, and data origin authenticity. Accordingly while access to the data and controls provided by the PrY data model does not allow an adversary to compromise security directly, by injecting frames, modifying frame data, or accessing that data, the model includes information that the PrY is intended to keep private. This private data includes the following read-only (config false) objects:

— **pry/transmission/frame-tx-statistics**, **pry/transmission/channel*channel-id/tx-statistics**, and **pry/reception/rx-statistics** counts.

— **/if:interfaces/if:interface/statistics**.

In addition any frame or octet statistics maintained by data models for entities higher in the PrY's interface stack, or in other interface stacks to which frames from that interface stack are forwarded (in the case of a bridge or router), or in higher layer entities in the system, can include information that the network administrator intends to be private.

Configuration (read-write) access to the following objects can also remove or reduce the level of privacy protection by allowing frames of different user priorities and sizes to be more clearly distinguished:

— **pry/transmission*user-priority/privacy-selection** (all objects).

— **pry/transmission/channel*channel-id** (enable, frame-access-priority, user-data-frame-size, and to a lesser extent other objects at this node).

The data model includes controls that an adversary can use to impact network performance and capability, as well as revealing correlatable information, including the following:

— **pry/transmission/privacy-selection/frame-padding**, if increased can reduce the available network bandwidth.

— **pry/transmission/channel*channel-id/requested-kbit-rate** controls the bandwidth available to user data frames allocated to the Privacy Channel when the default MPPDU generation algorithm is used. This bandwidth is used irrespective of the bandwidth taken by those user data frames. If set so high as to result in the MPPDU loss for the Privacy Channel, communicating user entities can reduce the number of user data frames they send but this 'back-off' does not reduce the bandwidth used by the Privacy Channel and the consequent probability of frame loss.

NOTE—A PrY using the default (20.9) MPPDU generation algorithm does not attempt to send MPPDUs faster than they can be transmitted by the interface of which it is part. Therefore, if connectivity to peer PrYs capable of line rate reception is provided by a simple LAN without intermediate buffering, Privacy Channel MPPDUs are not lost. However their transmission can preclude the transmission of other, lower priority frames.

— **pry/transmission*user-priority/privacy-selection/privacy-type**, if changed can change the access priority selected for MPPDUs for frames of a given user priority or change their allocation to a Privacy Channel, Privacy Frame, or unprotected transmission.

23.6 Interface stack models

The SecY and PrY YANG models support both explicit and augmented interface models. An explicitly modeled interface represents a single shim or sublayer in an interface stack. An augmented interface combines adjacent shims or sublayers into a single interface to simplify configuration. IEEE Std 802.1X provides both general examples and examples that are particular to the use of port-based network access control supported by MACsec, and the same notation is used in the following description.

Figure 23-5 illustrates these modeling choices for an end station interface with both MACsec or MAC Privacy protection. The small alphanumeric circles (labeled A, B, C) in the figures represent the if-index used to identify each interface (represented by the larger circles).

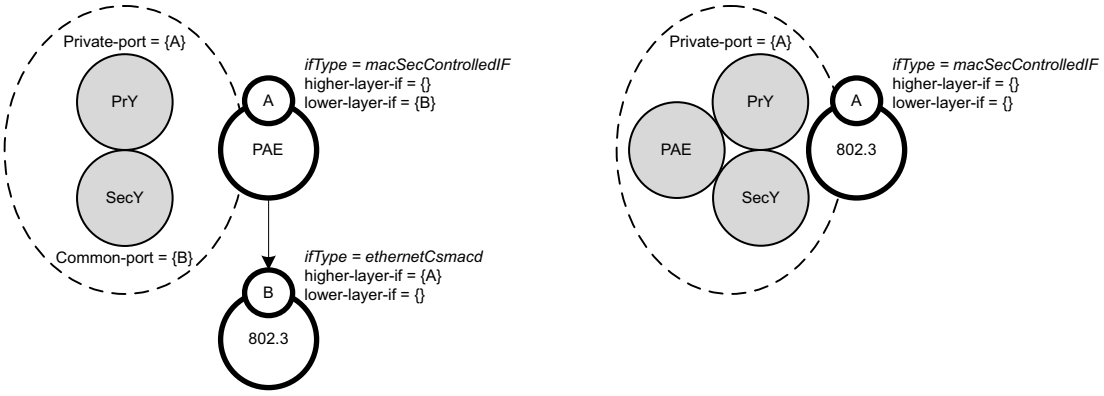


Figure 23-5—Explicit and augmented interface stack models for an end station

Figure 23-6 illustrates two further modeling choices for the interface.

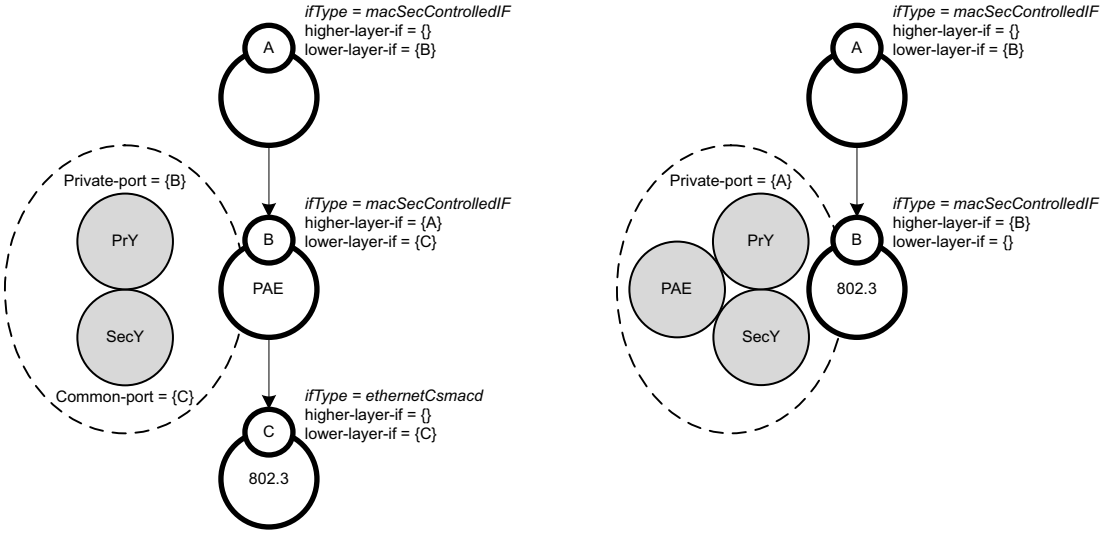


Figure 23-6—Two further interface stack modeling choices

The controller (e.g., NETCONF client) can dictate which interface stack model (explicit or augmented) is used. Consideration needs to be given to selecting the interface stack model that is right for the usage scenario. For example, starting with either of the two models shown in Figure 23-5 and transitioning to an interface stack that supports IEEE Std 802.1AX Link Aggregation (see Figure 23-7), requires the initial

interfaces to be torn down (e.g., deleted) and re-instantiated (e.g., created). This is not the case when starting from either of the two models shown in Figure 23-6 (see, for example, Figure 23-7).

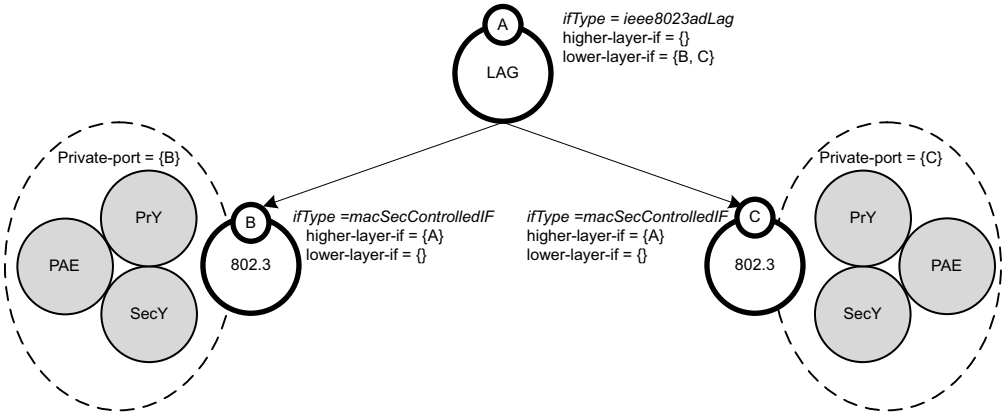


Figure 23-7—An interface stack model for link aggregation and MACsec

In Figure 23-5, Figure 23-6, and Figure 23-7, it is assumed that the network manager has overwritten the ifType to reflect the service provided by the top-most element of the interface stack. These models could be augmented by the Bridge Port model specified in IEEE Std 802.1Q, with an ifType of **bridge** for interface A.

Unless otherwise specified, and unlike a PAE, a protocol entity that is represented by an augment or reference to an interface that is augmented by a PrY uses the service provided by the PrY's Private Port (possibly indirectly if the interface has been augmented by additional shims). Similarly a protocol entity represented by an augment or reference to an interface augmented by a SecY, but not by a PrY, uses the secure MAC Service supported by that SecY. Some protocol entities, notably LLDP, can be usefully instantiated to make use of the insecure MAC service provided by a SecY's Common Port or the secure MAC Service provided by its Controlled Port. Figure 23-8 shows an interface stack supporting both.

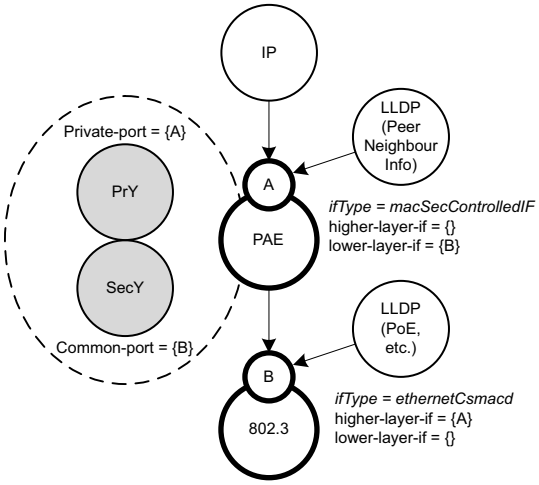


Figure 23-8—An interface stack with LLDP instances

In Figure 23-8, the lower instance of LLDP supports Power over Ethernet (PoE) operation prior to the establishment of MACsec secured connectivity which is used, by the upper LLDP instance, to share neighbor configuration information.

23.7 Security considerations for interface stack models

The security considerations that apply to use of interface stacks incorporating SecYs or PrYs are generally those applicable to those interface stacks (23.6) and the basic SecY and PrY models (23.3, 23.5), with the additions and exceptions described in this clause (23.7).

23.7.1 Interface stacks with link aggregation

When an interface stack includes link aggregation as specified by IEEE Std 802.1AX, the SecY and PrY managed objects for each Aggregation Port (e.g., the subinterface stacks below interfaces B and C in Figure 23-7) need to have different values of the Aggregation Key (so they are not attached to the same Aggregator) or identical values for the following objects:

- **pae/logon-nids/selected**
- **pae/logon-nids/connected**
- **pae/logon-nids/requested**

Different values for Aggregation Ports attached to the same Aggregator can compromise the security by allowing a port with different (or no) authentication, confidentiality, and integrity requirements to carry part of the data traffic for an Aggregator that had supported stricter security requirements. The **pae** for simple systems might only support a single null NID (identified by the null string), in which case the above object values are necessarily the same for all Aggregation Ports.

23.8 System models

A given system can comprise end station functionality, one or more bridge components, and other protocol components (e.g., supporting IP routing).

In principle each of the interfaces within a system can be supported by MACsec, MAC Privacy protection, or both. However the utility of such support depends on the connectivity from the SecY and PrY instances to their potential peers, and the existence of those peers. Correct configuration of each SecY's associated **pae/eapol-group-address**, together with the frame filtering provided by VLAN and MAC Bridge component Reserved Addresses (IEEE Std 802.1Q) guards against the attempted creation of inappropriate secured connectivity and (where a PrY is directly supported by a SecY) inappropriately instantiating privacy protection.

Where a system comprises multiple groups of interfaces each with naturally or potentially different SecY and PrY characteristics, each of those groups can be identified by a NID (**dot1x-types:pae-nid**) with **pae-system/pae-nid-group*nids** objects (possibly augmented) listing the attributes for each group. The attributes associated with the Customer Edge Ports of a Provider Edge Bridge (PEB) could, e.g., be associated with a NID that is the value of the **pae/logon-nids/selected** object for each of those port's PAEs. If a particular PEB supports distinct groups of Customer Edge Ports, e.g., in locations with different physical access characteristics, the **pae/logon-nids/selected** object could then differ accordingly. A **pae-nid** is a UTF-8 string and can be chosen to match a **pae/controlled-port-name** or a prefix that is part of the system's port naming convention for ports of a particular type.

This subclause (23.8) further describes aspects of the YANG model for particular systems, including those for the various types of EDE as specified in Clause 15 of this standard.

23.8.1 EDE models

An EDE-M is modeled (15.2, 15.4) as specified by IEEE Std 802.1Q for a VLAN-unaware MAC Bridge, with a SecY, and optionally a PrY, in the black-side interface stack (5.5). An EDE-CS is modeled (15.5) as specified by IEEE Std 802.1Q for a Provider Edge Bridge, with a SecY, and optionally a PrY, in the Provider Edge Port (PEP) interface stack (5.5). An EDE-CC and EDE-SS uses the IEEE Std 802.1Q Provider Edge Port model with the exception that both VLAN Bridge components are C-VLAN components in the EDE-CC model and both are S-VLAN components in the EDE-SS model.

23.9 Security considerations for system models

The general security considerations for interface stack models (23.7) apply to each interface that is MACsec or MAC Privacy protection capable. Additional consideration for EDEs are specified in 23.9.1.

23.9.1 EDE model security considerations

23.9.1.1 Use of C-VLAN multiplexing for MAC Privacy protection

An EDE-CC that also supports MAC Privacy protection is not constrained to use the same value for the outer VLAN tag added to a frame transmitted through its Provider Network Port [5.8 g), 15.6, Figure 15-8]. This optional capability is provided by managing the PVID (assigning the VID for that outer VLAN tag) for the internal Customer Network Port attached to the Provider Edge Port (PEP) that provides edge component egress for frames with the mapped inner VID. The purpose of this capability is to enhance privacy when frames for two or more different C-VLANs (as received at the Customer Edge Port) are to be conveyed across the Provider Bridged Network (PBN) to the same destination. However, if MACsec is not operational for that PEP, a SecTAG is not added by the PEP so an outer VLAN tag is not added. Not only is integrity, confidentiality, and privacy not provided in the absence of MACsec (as would be expected) but the customer's C-VLANs (as opposed to an outer C-VLAN added by the EDE) are used by the service provider to forward traffic. Configuring **pae-system/pae-nid-group*nids/unsecure-allowed** to never for each PEP's **pae/logon-nids/selected** can guard against both possibilities.

23.10 YANG module schema

The YANG data modules specified by this standard are summarized by simple tree diagrams with the following notation:

— Brackets "[" and "]" enclose list keys.
— Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).
— Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
— Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
— Ellipsis ("...") stands for contents of subtrees that are not shown.

23.10.1 ieee802-dot1ae-secy schema

```
module: ieee802-dot1ae-secy

  augment /if:interfaces/if:interface/dot1x:pae:
    +--rw secy
       +--rw controlled-port
       |  +--ro mac-enabled?               boolean
       |  +--ro mac-operational?           boolean
       |  +--ro oper-point-to-point-mac?   boolean
       |  +--rw admin-point-to-point-mac?  enumeration
       |  +--ro controlled-port-enabled?   boolean
       +--rw uncontrolled-port
       |  +--ro mac-enabled?               boolean
       |  +--ro mac-operational?           boolean
       |  +--ro oper-point-to-point-mac?   boolean
       |  +--rw admin-point-to-point-mac?  enumeration
       +--rw verification
       |  +--ro max-receive-channels?   uint8
       |  +--ro max-receive-keys?       uint8
       |  +--rw validate-frames?        enumeration
       |  +--rw replay-protect?         boolean
       |  +--rw replay-window?          uint32
       |  +--ro in-pkts-untagged?       yang:counter64
       |  +--ro in-pkts-no-tag?         yang:counter64
       |  +--ro in-pkts-bad-tag?        yang:counter64
       |  +--ro in-pkts-no-sa?          yang:counter64
       |  +--ro in-pkts-no-sa-error?    yang:counter64
       |  +--ro in-pkts-overrun?        yang:counter64
       |  +--ro in-octets-validated?    yang:counter64
       |  +--ro in-octets-decrypted?    yang:counter64
       |  +--ro receive-sc* [sci]
       |     +--ro sci                  sec-sci-type
       |     +--ro created-time?        yang:date-and-time
       |     +--ro started-time?        yang:date-and-time
       |     +--ro stopped-time?        yang:date-and-time
       |     +--ro receiving?           boolean
       |     +--ro in-pkts-ok?          yang:counter64
       |     +--ro in-pkts-unchecked?   yang:counter64
       |     +--ro in-pkts-delayed?     yang:counter64
       |     +--ro in-pkts-late?        yang:counter64
       |     +--ro in-pkts-invalid?     yang:counter64
       |     +--ro in-pkts-not-valid?   yang:counter64
       |     +--ro receive-sa* [rxa]
       |        +--ro in-use?           boolean
       |        +--ro ssci?             uint32
       |        +--ro next-pn?          sec-pn-type
       |        +--ro created-time?     yang:date-and-time
       |        +--ro started-time?     yang:date-and-time
       |        +--ro stopped-time?     yang:date-and-time
       |        +--ro rxa               sec-an-type
       |        +--ro lowest-pn?        sec-pn-type
       |        +--ro key-identifier?   sec-key-identifier-type
       +--rw generation
```

```
      |  +--ro sci-base?                sec-sci-type
      |  +--rw max-transmit-channels?   uint8
      |  +--rw max-transmit-keys?       uint8
      |  +--rw protect-frames?          boolean
      |  +--rw always-include-sci?      boolean
      |  +--rw use-es?                  boolean
      |  +--rw use-scb?                 boolean
      |  +--ro including-sci?           boolean
      |  +--ro out-pkts-untagged?       yang:counter64
      |  +--ro out-pkts-too-long?       yang:counter64
      |  +--ro out-octets-protected?    yang:counter64
      |  +--ro out-octets-encrypted?    yang:counter64
      |  +--rw user-priority-tc* [user-priority]
      |  |  +--rw user-priority       dot1q-types:priority-type
      |  |  +--rw traffic-class?      dot1q-types:priority-type
      |  |  +--rw access-class-de0?   uint8
      |  |  +--rw access-class-de1?   uint8
      |  +--ro transmit-sc* [sci]
      |     +--ro sci                   sec-sci-type
      |     +--ro created-time?         yang:date-and-time
      |     +--ro started-time?         yang:date-and-time
      |     +--ro stopped-time?         yang:date-and-time
      |     +--ro transmitting?         boolean
      |     +--ro encoding-sa?          sec-an-type
      |     +--ro out-pkts-protected?   yang:counter64
      |     +--ro out-pkts-encrypted?   yang:counter64
      |     +--ro transmit-sa* [txa]
      |        +--ro in-use?            boolean
      |        +--ro ssci?              uint32
      |        +--ro next-pn?           sec-pn-type
      |        +--ro created-time?      yang:date-and-time
      |        +--ro started-time?      yang:date-and-time
      |        +--ro stopped-time?      yang:date-and-time
      |        +--ro txa                sec-an-type
      |        +--ro confidentiality?   boolean
      |        +--ro key-identifier?    sec-key-identifier-type
      +--rw current-cipher-suite
      |  +--rw cipher-suite-identifier?   sec-eui64-type
      |  +--rw data-key* [key-index]
      |     +--rw key-index         uint32
      |     +--ro key-identifier?   sec-key-identifier-type
      |     +--ro transmits?        boolean
      |     +--ro receives?         boolean
      +--rw cipher-suite-control* [implemented-cipher-suite]
         +--rw implemented-cipher-suite    sec-eui64-type
         +--rw enable-use?                 boolean
         +--rw require-confidentiality?    boolean
augment /sys:system/dot1x:pae-system:
  +--rw secy-system
     +--rw cipher-suites* [cipher-suite]
        +--rw cipher-suite                  sec-eui64-type
        +--ro name?                         string
        +--ro integrity-protection?         boolean
        +--ro confidentiality-protection?   boolean
        +--ro changes-data-length?          boolean
        +--ro icv-length?                   uint16
```

23.10.2 ieee802-dot1ae-pry schema

```
module: ieee802-dot1ae-pry

  augment /if:interfaces/if:interface:
    +--rw pry {macsec-priv}?
       +--ro secy-support?             boolean
       +--ro pry-address?              ieee:mac-address
       +--rw pry-mppdu-dest-address?   ieee:mac-address
       +--rw max-peers?                uint8
       +--ro num-peers?                uint8
       +--ro peer-entry*               ieee:mac-address
       +--rw reception
       |  +--rw privacy-protection?             boolean
       |  +--ro default-reassembly-algorithm?   boolean
       |  +--rw rx-statistics
       |     +--ro in-user-frames?                    yang:counter64
       |     +--ro in-user-octets?                    yang:counter64
       |     +--ro in-pad-octets?                     yang:counter64
       |     +--ro in-mppdus?                         yang:counter64
       |     +--ro in-encapsulated-frames?            yang:counter64
       |     +--ro in-user-express-fragments?         yang:counter64
       |     +--ro in-user-preemptable-fragments?     yang:counter64
       |     +--ro in-express-discard-fragments?      yang:counter64
       |     +--ro in-preemptable-discard-fragments?  yang:counter64
       |     +--ro in-unknown-mppcis?                 yang:counter64
       |     +--ro in-errored-mppdus?                 yang:counter64
       |     +--ro in-user-unprotected-frames?        yang:counter64
       |     +--ro in-user-unprotected-octets?        yang:counter64
       +--rw transmission
          +--rw privacy-protection?    boolean
          +--rw privacy-selection* [user-priority]
          |  +--rw user-priority            dot1q-types:priority-type
          |  +--rw privacy-type             identityref
          |  +--rw frame-access-priority?   dot1q-types:priority-type
          |  +--rw frame-reveal-de?         enumeration
          |  +--rw frame-padding?           enumeration
          +--rw channel* [channel-id]
          |  +--rw channel-id               identityref
          |  +--rw enable?                  boolean
          |  +--rw fragment-enable?         boolean
          |  +--rw access-priority?         dot1q-types:priority-type
          |  +--rw user-data-frame-size?    uint16
          |  +--rw mppdu-generation?        enumeration
          |  +--rw requested-kbit-rate      uint32
          |  +--ro mppdu-bits-on-wire?      uint32
          |  +--ro mppdu-interval?          uint32
          |  +--rw user-burst-octets?       uint32
          |  +--rw tx-statistics
          |     +--ro out-ch-user-frames?         yang:counter64
          |     +--ro out-ch-user-octets?         yang:counter64
          |     +--ro out-ch-pad-octets?          yang:counter64
          |     +--ro out-mppdus?                 yang:counter64
          |     +--ro out-encapsulated-frames?    yang:counter64
          |     +--ro out-express-fragments?      yang:counter64
          |     +--ro out-preempt-fragments?      yang:counter64
          +--rw frame-tx-statistics
             +--ro out-pf-user-frames?       yang:counter64
             +--ro out-pf-user-octets?       yang:counter64
             +--ro out-pf-pad-octets?        yang:counter64
             +--ro out-unprotected-frames?   yang:counter64
             +--ro out-unprotected-octets?   yang:counter64

  notifications:
    +---n pry-max-peers-exceeded
       +--ro pry-interface?   -> /if:interfaces/interface/name
```

23.10.3 ieee802-dot1x-eapol schema

```
module: ieee802-dot1x-eapol

  augment /if:interfaces/if:interface/dot1x:pae:
    +--rw eapol-group-address?   ieee:mac-address
```

23.11 YANG modules[15] [16] [17]

23.11.1 ieee802-dot1ae-secy YANG module

```
module ieee802-dot1ae-secy {
  yang-version 1.1;
  namespace "urn:ieee:std:802.1AE:yang:ieee802-dot1ae-secy";
  prefix secy;

  import ietf-interfaces {
    prefix if;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-system {
    prefix sys;
  }
  import ieee802-dot1q-types {
    prefix dot1q-types;
  }
  import ieee802-dot1x {
    prefix dot1x;
  }

  organization
    "Institute of Electrical and Electronics Engineers";
  contact
    "WG-URL: http://ieee802.org/1/
     WG-EMail: stds-802-1-l@ieee.org

     Contact: IEEE 802.1 Working Group Chair
     Postal: C/O IEEE 802.1 Working Group
           IEEE Standards Association
           445 Hoes Lane
           Piscataway, NJ 08855
           USA

     E-mail: stds-802-1-chairs@ieee.org";
  description
    "The MAC security entity (SecY) YANG 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.";
```

[15] Copyright release for YANG: Users of this standard may freely reproduce the YANG modules contained in this standard so that they can be used for their intended purpose.

[16] An ASCII version of each YANG module is attached to the PDF of this standard and can also be obtained from the IEEE 802.1 Website at https://1.ieee802.org/yang-modules/.

[17] References in this standard's YANG module definitions are not clickable, as each module has been incorporated unchanged after development and verification using YANG tools.

```
  revision 2022-06-14 {
    description
      "The following reference statement identifies each referenced IEEE
       Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1AE Media Access Control (MAC) Security:
       IEEE Stds 802.1AE-2018, 802.1AE-2018-Cor1-2020, 802.1AEdk-2022.
       IEEE Std 802.1X Port-Based Network Access Control:
       IEEE Std 802.1X-2020.
       IEEE Std 802.1AC Media Access Control (MAC) Service Definition:
       IEEE Stds 802.1AC-2016, 802.1AC-2016-Cor1-2018.";
  }

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

  typedef sec-an-type {
    type uint8 {
      range "0..3";
    }
    description
      "A 2-bit number that is concatenated with a MACsec Secure
       Channel Identifier to identify a Secure Association. Indicates
       an Association Number (AN) assigned by the Key Server for use
       with the key number for transmission.

       Each SC is comprised of a succession of SAs, each with a
       different SAK, identified by a Secure Association Identifier
       (SAI) comprising an SCI concatenated with a two-bit AN. The SAI
       is unique for SAs used by SecYs participating in a given CA at
       any instant.";
    reference
      "9.6 of IEEE Std 802.1AE";
  }

  typedef sec-pn-type {
    type uint64;
    description
      "The Packet Number (PN). A 32-bit or 64-bit unsigned value.
       A monotonically increasing value that is guaranteed unique
       for each MACsec frame transmitted using a given Secure
       Association Key (SAK).";
    reference
      "9.8 of IEEE Std 802.1AE";
  }

  typedef sec-sci-type {
    type string {
      pattern '[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){5}-[0-9a-fA-F]{4}';
    }
    description
      "The Secure Channel Identifier (SCI). An 8 octet binary
       number, where the first (most significant) 6 octets
       represent the MAC Address (in canonical format), and the
       next 2 octets represents the Port Identifier. Integers can
       be entered as hexadecimal.";
    reference
      "9.9 of IEEE Std 802.1AE, 10.7.14, 10.7.23 and
       9.8 of IEEE Std 802.1X";
  }

  typedef sec-eui64-type {
    type uint64;
    description
      "A 64 bit identifier.";
    reference
      "10.7.25 of IEEE Std 802.1AE";
  }

  typedef sec-key-identifier-type {
```

```
    type string {
      length "0..32";
    }
    description
      "The sec-key-identifier-type 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
      "10.7.14, 10.7.23 and
       9.8 of IEEE Std 802.1X";
  }

  /* ---------------------------------------------------
   * Group objects used by IEEE Std 802.1AE YANG module
   * ---------------------------------------------------
   */

  grouping mac-status {
    description
      "This holds statistics for the Provided interface ports both the
       controlled port and the uncontrolled port.";
    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
        "6.4 of IEEE Std 802.1AE,
        11.2 and 11.3 of IEEE  Std 802.1AC";
    }
    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
        "6.4 of IEEE Std 802.1AE";
    }
    leaf oper-point-to-point-mac {
      type boolean;
      config false;
      description
        "If the oper-point-to-point-mac 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
        "6.5 of IEEE Std 802.1AE";
    }
    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 entity
            providing the service.";
          reference
            "6.5, 10.7.4 of IEEE Std 802.1AE";
        }
        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
          "6.5, 10.7.4 of IEEE Std 802.1AE";
      }
      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
           the entity providing the service.";
        reference
          "6.5, 10.7.4 of IEEE Std 802.1AE";
      }
    }
    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 admin-point-to-point-mac parameter can
       take one of three values.";
    reference
      "6.5, 10.7.4 of IEEE Std 802.1AE";
  }
}

//end provided-interface-grouping
/* 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 created-time {
    type yang:date-and-time;
    config false;
    description
      "The system time when the SC was created.";
    reference
      "10.7.12 of IEEE Std 802.1AE";
  }
  leaf started-time {
    type yang:date-and-time;
    config false;
    description
      "The system time when receiving last became True for
       the SC.";
    reference
      "10.7.12 of IEEE Std 802.1AE";
  }
  leaf stopped-time {
    type yang:date-and-time;
    config false;
    description
      "The system time when receiving last became False for
       the SC.";
    reference
      "10.7.12 of IEEE Std 802.1AE";
  }
}

// end secy-secure-channel-grouping
/* 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 in-use is True, and MAC_Operational is True for the Common
         Port, the SA can receive and transmit frames.";
      reference
        "10.7.14, 10.7.23 of IEEE Std 802.1AE";
    }
    leaf ssci {
      type uint32;
      config false;
      description
        "Short Secure Channel Identifier for the Send and Transmit SA";
      reference
        "10.7.14, 10.7.23 of IEEE Std 802.1AE";
    }
    leaf next-pn {
      type 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
        "10.7.14, 10.7.23 of IEEE Std 802.1AE";
    }
    leaf created-time {
      type yang:date-and-time;
      config false;
      description
        "The system time when the SA was created.";
      reference
        "10.7.14, 10.7.23 of IEEE Std 802.1AE";
    }
    leaf started-time {
      type yang:date-and-time;
      config false;
      description
        "The system time when in-use last became True for the
         SA.";
      reference
        "10.7.14 of IEEE Std 802.1AE";
    }
    leaf stopped-time {
      type yang:date-and-time;
      config false;
      description
        "The system time when in-use last became False for the
         SA.";
      reference
        "10.7.14 of IEEE Std 802.1AE";
    }
  }

  // end secy-secure-association-grouping
  /* ------------------------------------------------
   * Configuration objects used by IEEE Std 802.1AE YANG module
   * ------------------------------------------------
   */

  augment "/if:interfaces/if:interface/dot1x:pae" {
    description
      "SecY augments a PAE under an ietf interface.";
    container secy {
      description
        "Augment interface with 802.1 SecY configuration nodes. The
         management information for each SecY is indexed by
         controlled-port-number 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
    "10.7 of IEEE Std 802.1AE";
  container controlled-port {
    description
      "Controlled port control and status.";
    uses mac-status;
    leaf controlled-port-enabled {
      type boolean;
      config false;
      description
        "By setting controlled-port-enabled False, the KaY can
         prohibit use of the Controlled Port until the secure
         connectivity required has been configured.";
      reference
        "10.7.6 of IEEE Std 802.1AE";
    }
  }
  container uncontrolled-port {
    description
      "Uncontrolled port control and status.";
    uses mac-status;
  }
  container verification {
    description
      "The Verification controls for validation and replay
       protect for a given secy.";
    reference
      "10.6 of IEEE Std 802.1AE";
    leaf max-receive-channels {
      type uint8;
      config false;
      description
        "Specifies maximum number of receive channels for a SecY.";
      reference
        "10.7.7 of IEEE Std 802.1AE";
    }
    leaf max-receive-keys {
      type uint8;
      config false;
      description
        "Specifies maximum number of receive keys for a SecY.";
      reference
        "10.7.7 of IEEE Std 802.1AE";
    }
    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
    "10.7.8, Figure 10-4 of IEEE Std 802.1AE";
}
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 replay-protect and
     replay-window controls allows replay protection to be
     disabled, to operate on a packet number window, or to
     enforce strict frame order. If replay-protect is set but
     the replay-window is not zero, frames within the window can
     be received out of order; however, they are not replay
     protected.";
  reference
    "10.6.2, 10.4 of IEEE Std 802.1AE";
}
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
    "10.7.8 of IEEE Std 802.1AE";
}
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
    "10.7.9 of IEEE Std 802.1AE";
}
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
    "10.7.9 of IEEE Std 802.1AE";
}
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
        "10.7.9 of IEEE Std 802.1AE";
    }
    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
        "10.7.9 of IEEE Std 802.1AE";
    }
    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
        "10.7.9 of IEEE Std 802.1AE";
    }
    leaf in-pkts-overrun {
      type yang:counter64;
      config false;
      description
        "The number of packets discarded because they exceeded
         cryptographic performance capabilities.";
      reference
        "10.7.9 of IEEE Std 802.1AE";
    }
    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
        "10.6, 10.6.3 of IEEE Std 802.1AE";
    }
    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
        "10.6, 10.6.3 of IEEE Std 802.1AE";
    }
    list receive-sc {
      key "sci";
      config false;
      description
        "The Receive Security Channel Status for a given
         secure channel identifier.";
      reference
        "10.7.9 of IEEE Std 802.1AE";
      leaf sci {
        type 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
          "7.1.2 and figure 7.7 of IEEE Std 802.1AE";
      }
      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
        "10.7.12 of IEEE Std 802.1AE";
    }
    leaf in-pkts-ok {
      type yang:counter64;
      config false;
      description
        "For this SC, the number of validated packets.";
      reference
        "10.6.5, 10.7.9 of IEEE Std 802.1AE";
    }
    leaf in-pkts-unchecked {
      type yang:counter64;
      config false;
      description
        "For this SC, the number of packets while
         validate-frames was disabled.";
      reference
        "10.6.5, 10.7.9 of IEEE Std 802.1AE";
    }
    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
        "10.6.5, 10.7.9 of IEEE Std 802.1AE";
    }
    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
        "10.7.9 of IEEE Std 802.1AE";
    }
    leaf in-pkts-invalid {
      type yang:counter64;
      config false;
      description
        "For this SC, the number 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
        "10.7.9 of IEEE Std 802.1AE";
    }
    leaf in-pkts-not-valid {
      type yang:counter64;
      config false;
      description
        "For this SC, the number of packets discarded because
         validation failed and validate-frames was 'strict' or
         the data was encrypted (so the original frame could not
         be recovered).";
      reference
        "10.7.9 of IEEE Std 802.1AE";
    }
    list receive-sa {
      key "rxa";
      description
        "The Receive Security Association (SA) Status for
         this association.";
      uses secy-secure-association-grouping;
```

```
        leaf rxa {
          type sec-an-type;
          description
            "The Association Number for this Receiving SA.";
          reference
            "10.7.13 of IEEE Std 802.1AE";
        }
        leaf lowest-pn {
          type 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
            "10.7.14 of IEEE Std 802.1AE";
        }
        leaf key-identifier {
          type 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
            "10.7.14, 10.7.24, of IEEE Std 802.1AE and
            9.8 of IEEE Std 802.1X";
        }
      }
    }
  }
  container generation {
    description
      "The Generation controls for given secy.";
    reference
      "10.5 of IEEE Std 802.1AE";
    leaf sci-base {
      type sec-sci-type;
      config false;
      description
        "The base for a set of secure channels Security
         Channel Identifier.";
      reference
        "7.1.2, 10.7.17 of IEEE Std 802.1AE";
    }
    leaf max-transmit-channels {
      type uint8;
      description
        "Number of transmit channels.";
      reference
        "10.7.16 of IEEE Std 802.1AE";
    }
    leaf max-transmit-keys {
      type uint8;
      description
        "Number of transmit keys.";
      reference
        "10.7.16 of IEEE Std 802.1AE";
    }
    leaf protect-frames {
      type boolean;
      default "true";
      description
        "The protect-frames control is provided to facilitate
         deployment.";
      reference
        "10.7.17 of IEEE Std 802.1AE";
```

```
  }
  leaf always-include-sci {
    type boolean;
    default "false";
    description
      "Mandates inclusion of an explicit SCI in the SecTAG when
       transmitting protected frames.";
    reference
      "10.5.3, 10.7.17 of IEEE Std 802.1AE";
  }
  leaf use-es {
    type boolean;
    default "false";
    description
      "Enables use of the ES bit in the SecTAG when transmitting
       protected frames.";
    reference
      "10.5.3, 10.7.17 of IEEE Std 802.1AE";
  }
  leaf use-scb {
    type boolean;
    default "false";
    description
      "Enables use of the SCB bit in the SecTAG when
       transmitting protected frames.";
    reference
      "10.5.3, 10.7.17 of IEEE Std 802.1AE";
  }
  leaf including-sci {
    type boolean;
    config false;
    description
      "True if an explicit SCI is included in the SecTAG when
       transmitting protected frames.";
    reference
      "10.5.3, 10.7.17 of IEEE Std 802.1AE";
  }
  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
      "10.7.18 of IEEE Std 802.1AE";
  }
  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
      "10.7.18 of IEEE Std 802.1AE";
  }
  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
      "10.7.9 of IEEE Std 802.1AE";
  }
  leaf out-octets-encrypted {
    type yang:counter64;
    config false;
    description
      "The number of plain text octets integrity protected and
       encrypted in transmitted frames.";
    reference
      "10.7.9 of IEEE Std 802.1AE";
```

```
  }
  list user-priority-tc {
    key "user-priority";
    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 default for this table is every row has a non-mapping
       priority with the first row having all zeros, the second row
       having all ones etc. up to the last row having all sevens.";
    reference
      "10.7.17 of IEEE Std 802.1AE";
    leaf user-priority {
      type dot1q-types:priority-type;
      description
        "The User Priority.";
      reference
        "10.7.17 of IEEE Std 802.1AE";
    }
    leaf traffic-class {
      type dot1q-types:priority-type;
      description
        "The traffic class that maps to the four most significant
         bits of the Port Identifier component of the SCI for the
         selected SC.";
      reference
        "10.7.17 of IEEE Std 802.1AE";
    }
    leaf access-class-de0 {
      type uint8 {
        range "0..15";
      }
      description
        "The access priority when not drop eligible.";
      reference
        "10.7.17 of IEEE Std 802.1AE";
    }
    leaf access-class-de1 {
      type uint8 {
        range "0..15";
      }
      description
        "The access priority when drop eligible.";
      reference
        "10.7.17 of IEEE Std 802.1AE";
    }
  }
  list transmit-sc {
    key "sci";
    config false;
    description
      "The transmit Security Channel, status for a given
       Security Channel Identifier.";
    reference
      "10.7.1 of IEEE Std 802.1AE";
    leaf sci {
      type 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
        "7.1.2 and figure 7.7 of IEEE Std 802.1AE";
    }
    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
        "10.7.21 of IEEE Std 802.1AE";
    }
    leaf encoding-sa {
      type sec-an-type;
      config false;
      description
        "The current value of the encoding-sa variable for the
         selected transmit SC.";
      reference
        "10.7.24 of IEEE Std 802.1AE";
    }
    leaf out-pkts-protected {
      type yang:counter64;
      config false;
      description
        "The number of integrity protected but not encrypted
         packets for this transmit SC.";
      reference
        "10.7.18, Figure 10-3 of IEEE Std 802.1AE";
    }
    leaf out-pkts-encrypted {
      type yang:counter64;
      config false;
      description
        "The number of integrity protected and encrypted packets
         for this transmit SC.";
      reference
        "10.7.18, Figure 10-3 of IEEE Std 802.1AE";
    }
    list transmit-sa {
      key "txa";
      config false;
      description
        "The transmit security association status for a
         given association number.";
      uses secy-secure-association-grouping;
      leaf txa {
        type sec-an-type;
        config false;
        description
          "The association number for the SA.";
        reference
          "10.7.23 of IEEE Std 802.1AE";
      }
      leaf confidentiality {
        type boolean;
        config false;
        description
          "True if the SA provides confidentiality as well as
           integrity for transmitted frames.";
        reference
          "10.7.23 of IEEE Std 802.1AE";
      }
      leaf key-identifier {
        type 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
          "10.7.14, 14.7, 14.8 of IEEE Std 802.1AE,
```

```
            9.8 of IEEE Std 802.1X";
        }
      }
    }
  }
  // end generation
  container current-cipher-suite {
    description
      "The current-cipher-suite is selected by the KaY.
       The Current Cipher Suite may also be selected and keys
       created by management, but a conformant implementation
       shall provide a mechanism to allow such selection
       and creation by network management to be disabled.";
    leaf cipher-suite-identifier {
      type sec-eui64-type;
      description
        "The Cipher Suite currently used by this SecY.";
      reference
        "10.7.27 of IEEE Std 802.1AE";
    }
    list data-key {
      key "key-index";
      description
        "An index of Keys Used.";
      leaf key-index {
        type uint32;
        description
          "Numeric key number used as index.";
        reference
          "10.7.27 of IEEE Std 802.1AE";
      }
      leaf key-identifier {
        type 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 MKPDUs, so network
           management equipment and personnel can observe and
           diagnose MKA operation (if necessary) without having
           access to any secret key.";
        reference
          "10.7.28 of IEEE Std 802.1AE";
      }
      leaf transmits {
        type boolean;
        config false;
        description
          "Transmits True means key is used for transmitting
           direction.";
        reference
          "10.5 of IEEE Std 802.1AE";
      }
      leaf receives {
        type boolean;
        config false;
        description
          "Receives True means key is used for receiving
           direction.";
        reference
          "10.5 of IEEE Std 802.1AE";
      }
    }
  }
  // end current-cipher-suite
  list cipher-suite-control {
    key "implemented-cipher-suite";
    description
      "The MKA Key Server selects the Cipher Suite to be used to
```

```
          protect communication within a CA. If enable-use is False
          for the selected Cipher Suite, the SecY does not participate
          in the CA and MAC_Operational for the Controlled Port
          remains False. If the MKA Key Server has selected integrity
          protection and enable-use and require-confidentiality are
          both True for the selected Cipher Suite, confidentiality
          protection is used.";
        leaf implemented-cipher-suite {
          type sec-eui64-type;
          description
            "cipher suite identifier (EUI-64)";
          reference
            "10.7.26 of IEEE Std 802.1AE";
        }
        leaf enable-use {
          type boolean;
          default "true";
          description
            "Enables use of the Cipher Suite by this SecY.";
          reference
            "10.7.26 of IEEE Std 802.1AE";
        }
        leaf require-confidentiality {
          type boolean;
          default "true";
          description
            "True if confidentiality protection is required if
             this Cipher Suite is used.";
          reference
            "10.7.26 of IEEE Std 802.1AE";
        }
      }
    }
  }

  // end secy augment interfaces
  /*
   Secy System
   */

  augment "/sys:system/dot1x:pae-system" {
    description
      "Augment system with 802.1AE MACSec System Cipher Suites nodes.";
    container secy-system {
      description
        "Augment system with 802.1AE SecY configuration nodes.";
      list cipher-suites {
        key "cipher-suite";
        description
          "A list of configuration parameters and operational state
           associated with a cipher suite.";
        leaf cipher-suite {
          type sec-eui64-type;
          description
            "A globally unique 64-bit (EUI-64) identifier for this
             cipher suite.";
          reference
            "10.7.25 of IEEE Std 802.1AE";
        }
        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
            "10.7.25 of IEEE Std 802.1AE";
        }
        leaf integrity-protection {
          type boolean;
```

```
          config false;
          description
            "True if integrity protection without confidentiality can
             be provided.";
          reference
            "10.7.25 of IEEE Std 802.1AE";
        }
        leaf confidentiality-protection {
          type boolean;
          config false;
          description
            "True if confidentiality with integrity protection can be
             provided.";
          reference
            "10.7.25 of IEEE Std 802.1AE";
        }
        leaf changes-data-length {
          type boolean;
          config false;
          description
            "Indicates that the cipher suite changes the data length.";
          reference
            "10.7.25 of IEEE Std 802.1AE";
        }
        leaf icv-length {
          type uint16;
          config false;
          description
            "The number of octets in the ICV.";
          reference
            "10.7.25 of IEEE Std 802.1AE";
        }
      }
    }
  } // end /sys:system

}
```

23.11.2 ieee802-dot1ae-pry YANG module

```
module ieee802-dot1ae-pry {
  yang-version 1.1;
  namespace "urn:ieee:std:802.1AE:yang:ieee802-dot1ae-pry";
  prefix pry;

  import ietf-interfaces {
    prefix if;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ieee802-dot1q-types {
    prefix dot1q-types;
  }
  import ieee802-types {
    prefix ieee;
  }
  import iana-if-type {
    prefix ianaift;
  }

  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http:ieee802.org/1/
     WG-EMail: stds-802-1-l@ieee.org

     Contact: IEEE 802.1 Working Group Chair
     Postal: C/O IEEE 802.1 Working Group
           IEEE Standards Association
           445 Hoes Lane
           Piscataway, NJ 08855
           USA

     E-mail: stds-802-1-chairs@ieee.org";
  description
    "This YANG module augments the configuration and operational state
     data for interfaces for the MAC Privacy project: Std 802.1AE; see
     that standard and its amendments for full legal notices.

     A MAC Privacy protection Entity (PrY) is a protocol shim in an
     interface stack that encapsulates user data frames in MAC Privacy
     protection Data Units (MPPDUs). Once those MPPDUs are
     confidentiality protected by MACsec, the ability of potential
     adversaries to draw conclusions from the source and destination
     MAC addresses, sizes, and transmission timing and frequency of
     user data frames is reduced or eliminated.

     Each PrY in a system and its managed objects augments its upper
     interface (Private Port), which provides a privacy protected
     service to its user, typically a Bridge Port (IEEE Std 802.1Q) or
     an end station protocol stack. Object names can be conveniently
     pronounced by rendering Pry as Privacy.";

  revision 2022-06-17 {
    description
      "The following reference statement identifies each referenced IEEE
       Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1AE Media Access Control (MAC) Security:
       IEEE Stds 802.1AE-2018, 802.1AE-2018-Cor1-2020, 802.1AEdk-2022.
       IEEE Std 802.1X Port-Based Network Access Control:
       IEEE Std 802.1X-2020.
       IEEE Std 802.1Q Bridges and Bridged Networks:
       IEEE Std 802.Q-2022";
  }

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

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

feature macsec-priv {
  description
    "Feature MAC Privacy.";
}

/*--------------------*/
/*   identities       */
/*--------------------*/

identity priority-map-identity {
  description
    "Base identity for assigning a priority to a Privacy type.";
}

identity channel-identity {
  description
    "Base identity for privacy channel.";
}

identity express-channel {
  base channel-identity;
  base priority-map-identity;
  description
    "This is the express privacy channel frame designation.";
  reference
    "20.13.4 of IEEE Std 802.1AE";
}

identity preemptable-channel {
  base channel-identity;
  base priority-map-identity;
  description
    "This is the preemptable privacy channel designation.";
  reference
    "20.13.4 of IEEE Std 802.1AE";
}

identity frame-identity {
  description
    "Base identity for privacy frame.";
}

identity privacy-frame {
  base frame-identity;
  base priority-map-identity;
  description
    "This is a privacy frame designation.";
  reference
    "3 of IEEE Std 802.1AE";
}

identity none-identity {
  description
    "Base identity for privacy frame.";
}

identity none {
  base none-identity;
  base priority-map-identity;
  description
    "This is no privacy encapsulation. Frames mapped to this
     identity are forwarded directly without MAC privacy
     encapsulation.";
  reference
    "17 of IEEE Std 802.1AE";
}

/*-------------------------*/
/* Notification statements */
/*-------------------------*/
```

```
notification pry-max-peers-exceeded {
  description
    "A max-peers-exceeded notification is sent when the value
     of if-num-peers exceeds if-max-peers. This is triggered
     only on the transition to the exceeded state and reset
     when the if-num peers is less than or equal to
     if-num-peers.";
  leaf pry-interface {
    type leafref {
      path "/if:interfaces/if:interface/if:name";
    }
    description
      "Contains the interface name containing the PrY that has
       exceeded the number of peers.";
  }
}

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

augment "/if:interfaces/if:interface" {
  when "if:type = 'ianaift:ethernetCsmacd' or if:type = "
     + "'ianaift:ilan' or if:type = 'ianaift:macSecControlledIF' or "
     + "if:type = 'ianaift:ptm' or if:type = 'ianaift:bridge'" {
    description
      "Augment interfaces with 802.1ae MACSec System specific
       configuration nodes.";
  }
  if-feature "macsec-priv";
  description
    "MACsec Privacy Mode.";
  container pry {
    description
      "Configure the MAC Privacy Options.";
    leaf secy-support {
      type boolean;
      config false;
      description
        "Set True by the system if the PrY is directly supported
         by a SecY and MKA, and False otherwise.  When True, the
         value of if-mppdu-dest-address and the entries in the
         PrYs peer address table (perr-entry list) are determined
         by the Key Agreement Entity (KaY) operating MKA, and are
         not writable by network management.";
      reference
        "23.4, of IEEE Std 802.1AE
         11.1.1 IEEE Std 802.1X";
    }
    leaf pry-address {
      type ieee:mac-address;
      config false;
      description
        "The individual MAC address associated with the PrY and
         other components of the PrYs interface stack.  Allocated
         by the system. Used by PrY as the source address of
         MPPDUs and by a supporting SecY (if present) for SCI
         assignment. The PrY will receive and process MPPDUs with
         this destination address.";
      reference
        "18.1, 23.4 of IEEE Std 802.1AE";
    }
    leaf pry-mppdu-dest-address {
      type ieee:mac-address;
      description
        "The destination MAC address used by the PrY to transmit
         MPPDUs. Also used to receive MPPDUs (if a Group address)
         when  reception privacy-protection  is True. Set by the
         KaY if if-secy-support is True, otherwise writable. If
         if-secy-support transitions from True to False, defaults
         to the Nearest non-TPMR Bridge Group address.";
```

```
    reference
      "18.1, 20.11 of IEEE Std 802.1AE
       11.1.1 IEEE Std 802.1X";
  }
  leaf max-peers {
    type uint8;
    description
      "The maximum number of peer PrYs supported by the
       configured reassembly algorithm.";
    reference
      "20.11, 20.13, 23.8 of IEEE Std 802.1AE";
  }
  leaf num-peers {
    type uint8;
    config false;
    description
      "The number of peer PrYs detected by the system. This
       value may be greater than if-max-peers a notification is
       raised when this value exceeds if-max-peers.";
    reference
      "20.13, of IEEE Std 802.1AE";
  }
  leaf-list peer-entry {
    type ieee:mac-address;
    config false;
    description
      "A list of peer PrYs. Frame Fragments received in MPPDUs
       with source MAC addresses not in this table are
       discarded.  When if-secy-support is True, table entries
       are created and deleted by the supporting Key Agreement
       Entity. When False the system automatically creates an
       entry for if-mppdu-dest-address if that is not a Group
       address, and other entries can be created by management.";
    reference
      "20.13, of IEEE Std 802.1AE";

  }
  container reception {
    description
      "Configure the MAC Privacy Reception.";
    leaf privacy-protection {
      type boolean;
      default "true";
      description
        "MACSec Privacy Reception Enable - True or False. When
         True the PrY processes received MPPDUs addressed to
         pry-address and if-mppdu-dest-address (if that is a Group
         address).  When False they are passed directly to the
         PrY's Private Port. All other MPPDUs are passed to the
         Private Port, unprocessed, irrespective of this control’s
         value.";
      reference
        "20.11 of IEEE Std 802.1AE";
    }
    leaf default-reassembly-algorithm {
      type boolean;
      config false;
      description
        "Set True by the system to indicate that the default
         reassembly algorithm is used. Set False, otherwise.  If
         the system supports additional reassembly algorithms it
         shall also support selection of the default algorithm.
         The maximum size of the user data frame (DA, SA, MSDU)
         that can be reassembled for delivery to the Private Port
         is the value of if-mtu (as provided by the IF-MIB plus 22
         octets).";
      reference
        "20.13, 20.13.1 of IEEE Std 802.1AE";
    }
    container rx-statistics {
      description
        "Configure the MAC Privacy Reception Statistics.";
```

```
leaf in-user-frames {
  type yang:counter64;
  config false;
  description
    "Total number of protected user data frames received in
     MPPDUs, encoded as Encapsulated Frames or reassembled
     from Frame Fragments.";
  reference
    "20.14.2 of IEEE Std 802.1AE";
}
leaf in-user-octets {
  type yang:counter64;
  config false;
  description
    "Total number of user data frame octets received.
     Excludes padding.";
  reference
    "20.14.2 of IEEE Std 802.1AE";
}
leaf in-pad-octets {
  type yang:counter64;
  config false;
  description
    "Number of pad octets received in MPPDUs. This
     includes MPPDU overhead bytes.";
  reference
    "20.14.2 of IEEE Std 802.1AE";
}
leaf in-mppdus {
  type yang:counter64;
  config false;
  description
    "Total number of MAC Privacy PDUs received.";
  reference
    "20.14.2 of IEEE Std 802.1AE";
}
leaf in-encapsulated-frames {
  type yang:counter64;
  config false;
  description
    "Total number of MAC Privacy user frames received
     that were not fragmented.";
  reference
    "20.14.2 of IEEE Std 802.1AE";
}
leaf in-user-express-fragments {
  type yang:counter64;
  config false;
  description
    "Total number of correctly encoded Express Frame
     Fragments received in MPPDUs. Includes fragments
     discarded by reassembly (unknown peer, too many peers,
     out of order, reassembled frame too large).";
  reference
    "20.14.2 of IEEE Std 802.1AE";
}
leaf in-user-preemptable-fragments {
  type yang:counter64;
  config false;
  description
    "Total number of correctly encoded Preemptable Frame
     Fragments received in MPPDUs. Includes fragments
     discarded by reassembly (unknown peer, too many peers,
     out of order, reassembled frame too large).";
  reference
    "20.14.2 of IEEE Std 802.1AE";
}
leaf in-express-discard-fragments {
  type yang:counter64;
  config false;
  description
    "Number of Express Frame Fragment discard events
```

```
            (discarding a fragment and/or a partially reassembled
            user data frame).";
        reference
          "20.14.2 of IEEE Std 802.1AE";
      }
      leaf in-preemptable-discard-fragments {
        type yang:counter64;
        config false;
        description
          "Number of Preemptable Frame Fragment discard events
           (discarding a fragment and/or a partially reassembled
           user data frame).";
        reference
          "20.14.2 of IEEE Std 802.1AE";
      }
      leaf in-unknown-mppcis {
        type yang:counter64;
        config false;
        description
          "Number of of unknown MPPDU components received.";
        reference
          "20.14.2 of IEEE Std 802.1AE";
      }
      leaf in-errored-mppdus {
        type yang:counter64;
        config false;
        description
          "Number of received MPPDUs containing an incorrectly
           encoded component.";
        reference
          "20.14.2 of IEEE Std 802.1AE";
      }
      leaf in-user-unprotected-frames {
        type yang:counter64;
        config false;
        description
          "Total number of frames with no privacy protection
           received.";
        reference
          "20.14.2 of IEEE Std 802.1AE";
      }
      leaf in-user-unprotected-octets {
        type yang:counter64;
        config false;
        description
          "Total number of octets with no privacy protection
           received.";
        reference
          "20.14.2 of IEEE Std 802.1AE";
      }
    }
  }
  container transmission {
    description
      "Configure the MAC Privacy Transmission.";
    leaf privacy-protection {
      type boolean;
      default "true";
      description
        "MACSec Privacy Enable - True or False. When True, the PrY
         protects transmitted user data frames as configured in
         the Privacy Selection list. When False, all user data
         frames are passed directly to the PrY's Controlled
         Port.";
      reference
        "20.5 of IEEE Std 802.1AE";
    }
    list privacy-selection {
      key "user-priority";
      description
        "User priority is mapped to privacy channels express or
         preemptable or to privacy frames.";
```

```
reference
    "17.4, 17.4.3, 20.5 of IEEE Std 802.1AE";
leaf user-priority {
  type dot1q-types:priority-type {
    range "0..7";
  }
  description
    "Transmit request user priority. There are eight values
     of User Priority that map to ether a priority channel,
     a priority frame or to none.";
  reference
    "17.4.3, 20.5 of IEEE Std 802.1AE";
}
leaf privacy-type {
  type identityref {
    base priority-map-identity;
  }
  mandatory true;
  description
    "An identity associated with the privacy channel or
     frame. Privacy protection type: none, privacy-frame,
     preemptable-channel, or express-channel.";
  reference
    "17.4.3, 20.5 of IEEE Std 802.1AE";
}
leaf frame-access-priority {
  type dot1q-types:priority-type;
  description
    "The Controlled Port priority (access priority) used to
     transmit Privacy Frames with the Private Port
     transmission priority (user priority) that selects this
     table entry.";
  reference
    "17.4.3, 20.7 of IEEE Std 802.1AE";
}
leaf frame-reveal-de {
  type enumeration {
    enum hidden {
      value 0;
      description
        "Set to zero to hide (clear) drop_eligible for Privacy Frames
         transmission.";
        }
    enum visible {
      value 1;
      description
        "Set to one to use (make visible) the drop_eligible value
         provided by the PrY's user for Privacy Frame transmission.";
        }
  }
  default "hidden";
  description
    "frame-reveal-de allows the drop_eligible parameter accompanying
     Privacy Frame transmission to be as supplied by the PrY's user or
     hidden.";
  reference
    "17.4.1, 20.7 of IEEE Std 802.1AE";
}
leaf frame-padding {
  type enumeration {
    enum none {
      value 1;
      description
        "Set to none when no extra pad octets are added.";
    }
    enum to-16 {
      value 16;
      description
        "Set to 16 when padding out to the nearest 16 octet
         boundary.";
    }
    enum to-32 {
```

```
          value 32;
          description
            "Set to 32 when padding out to the nearest 32 octet
             boundary.";
        }
        enum to-64 {
          value 64;
          description
            "Set to 64 when padding out to the nearest 64 octet
             boundary.";
        }
      }
      default "to-64";
      description
        "Specifies padding of the Privacy Frame MPPDU (excluding
         its source and destination MAC addresses) to four
         octets (to allow for the MAC Privacy protection
         EtherType and the MPPCI for an Encapsulated Frame) plus
         the nearest multiple of one(1) (for no padding),
         sixteen(16), thirty two(32), or sixty four (64) octets.
         The specified size excludes any octets to be added by
         supporting components lower in the interface stack
         (e.g. a MACsec SecTAG and ICV, and the Ethernet FCS) or
         other bridge components (e.g. an outer VLAN tag added
         by an EDE's network component).";
      reference
        "17.4.2, 20.7 of IEEE Std 802.1AE";
    }
  }
  list channel {
    key "channel-id";
    description
      "List of Channels supported with their corresponding per
       channel configuration Note both channels are forced to be
       configured.";
    reference
      "20.13.6 of IEEE Std 802.1AE";
    leaf channel-id {
      type identityref {
        base channel-identity;
      }
      description
        "The Channel may be express or preemptable. If only one is active
         then all traffic maps to the active channel and the express
         indication bit is set.";
    }
    leaf enable {
      type boolean;
      default "false";
      description
        "When True, user data frames assigned to this Privacy
         Channel a privacy-selection are transmitted using this
         channel's parameters. When False, they are transmitted
         using the other channel if enable is True
         for that channel and transmitted as Privacy Frames
         using the relevant frame privacy-type otherwise.";
      reference
        "20.8 of IEEE Std 802.1AE";
    }
    leaf fragment-enable {
      type boolean;
      default "true";
      description
        "When True permits user data frame fragmentation in this
         Privacy Channel. Should be True, for bandwidth
         efficiency and delay minimization. Provided to allow
         simple performance testing and fragmentation benefit
         analysis.";
      reference
        "20.10 of IEEE Std 802.1AE";
    }
    leaf access-priority {
```

```
      type dot1q-types:priority-type;
      description
        "The Controlled Port priority (access priority) used to
         transmit MPPDUs for this Privacy Channel.";
      reference
        "20.8 20.5.9.1 of IEEE Std 802.1AE";
    }
    leaf user-data-frame-size {
      type uint16 {
        range "128 .. 32768";
      }
      units "octets";
      default "1522";
      description
        "The largest user data frame, at the Private Port interface (i.e.
         prior to MAC Privacy protection) that can be transmitted as an
         MPPDU Encapsulated Frame without fragmentation.  Default allows
         for a standard Ethernet frame with a single VLAN tag. (The
         number of octets in an encapsulated frame component after the
         'following length' will be 1518 as the FCS is not encoded).  The
         user data frame size excludes octets subsequently added by
         MACsec, or other supporting interface stack components.
         Physical media, and the configuration of other system components
         can impose an upper bound lower than the configured value of
         this parameter.";
      reference
        "20.9.4 of IEEE Std 802.1AE";
    }
    leaf mppdu-generation {
      type enumeration {
        enum default {
          value 1;
          description
            "Default represents a regular timed delivery of
             a Privacy Channel based on requested-kbit-rate
             and mppdu-bits-on-wire.";
        }
        enum transmission-gate {
          value 2;
          description
           "Transmission-gate specifies transmission gate
            control of Privacy Channel MPPDU transmission.";
        }
        enum other {
          value 3;
          description
            "Optional other timing.";
        }
      }
      default "default";
      description
        "The MPPDU generation algorithm for this Privacy
         Channel. When default (fixed-rate), the (maximum)
         bandwidth is requested, with a catch up
         (burst)parameter to recover lost bandwidth if an MPPDU
         transmission has been delayed by another frame sent
         with higher access priority or by another component of
         the same interface stack. When transmission-gate, MPPDU
         transmission timing is gated.";
      reference
        "20.9, 20.9.4, 20.9.5 of IEEE Std 802.AE.
         IEEE Std 802.1Q";
    }
    leaf requested-kbit-rate {
      type uint32;
      units "kbit/s";
      mandatory true;
      description
        "The physical medium bit rate (kilobits per second) to
         be used by this Privacy Channel and the default MPPDU
         generation algorithm in the absence of higher priority
         traffic or other resource competition.";
```

```
      reference
        "23.5 of IEEE Std 802.1AE";
    }
    leaf mppdu-bits-on-wire {
      type uint32;
      units "octets";
      config false;
      description
        "The number of bit times required to transmit an MPPDU
         that conveys a single, Private Port transmitted, user
         data frame of user-data-frame-size encoded as an
         Encapsulated Frame(19.5.1). Calculated by the system,
         including all fields added by the interface stack.";
      reference
        "20.9.4 of IEEE Std 802.1AE";
    }
    leaf mppdu-interval {
      type uint32;
      units "nanoseconds";
      config false;
      description
        "The approximate interval (as calculated by the system)
         in nanoseconds between the transmission of MPPDUs for
         this Privacy Channel, in the absence of competing
         higher priority traffic or other resource competition.";
      reference
        "20.9.4 of IEEE Std 802.1AE";
    }
    leaf user-burst-octets {
      type uint32;
      description
        "The number of additional user data frame burst for use
         by the default MPPDU generation algorithm to recover
         channel bandwidth lost to competing higher priority
         traffic.";
      reference
        "20.9.4 of IEEE Std 802.1AE";
    }
    container tx-statistics {
      description
        "Transmission statistics for a Privacy Channel.";
      leaf out-ch-user-frames {
        type yang:counter64;
        config false;
        description
          "Number of user data frames sent in this Privacy
           Channel.";
        reference
          "20.14.1 of IEEE Std 802.1AE";
      }
      leaf out-ch-user-octets {
        type yang:counter64;
        config false;
        description
          "Number of user data octets sent in this Privacy
           Channel.  Not counting pad octets.";
        reference
          "20.14.1 of IEEE Std 802.1AE";
      }
      leaf out-ch-pad-octets {
        type yang:counter64;
        config false;
        description
          "Number of pad octets sent in this Privacy Channel.
           This includes MPPDU overhead.";
        reference
          "20.14.1 of IEEE Std 802.1AE";
      }
      leaf out-mppdus {
        type yang:counter64;
        config false;
        description
```

```
          "Number of MPPDUs sent in this Privacy Channel.";
        reference
          "20.14.1 of IEEE Std 802.1AE";
      }
      leaf out-encapsulated-frames {
        type yang:counter64;
        config false;
        description
          "Number of Encapsulated Frames encoded for this
           Privacy Channel.";
        reference
          "20.14.1 of IEEE Std 802.1AE";
      }
      leaf out-express-fragments {
        type yang:counter64;
        config false;
        description
          "Number of Express Fragments encoded for this Privacy
           Channel.";
        reference
          "20.14.1 of IEEE Std 802.1AE";
      }
      leaf out-preempt-fragments {
        type yang:counter64;
        config false;
        description
          "Number of Preemptable Fragments encoded for this
           Privacy Channel.";
        reference
          "20.14.1 of IEEE Std 802.1AE";
      }
    }
  }
  container frame-tx-statistics {
    description
      "Frame Transmission stats.";
    leaf out-pf-user-frames {
      type yang:counter64;
      config false;
      description
        "Total number of user data frames sent as Privacy Frames
         (each in a separate MPPDU).";
      reference
        "20.14.1 of IEEE Std 802.1AE";
    }
    leaf out-pf-user-octets {
      type yang:counter64;
      config false;
      description
        "Total number of user data octets sent in Privacy Frames
         (each user data frame in a separate MPPDU). Not
         counting pad octets.";
      reference
        "20.14.1 of IEEE Std 802.1AE";
    }
    leaf out-pf-pad-octets {
      type yang:counter64;
      config false;
      description
        "Total number of pad octets sent in Privacy Frames (each
         conveying a single Private Port user data frame).  This
         includes MPPDU overhead.";
      reference
        "20.14.1 of IEEE Std 802.1AE";
    }
    leaf out-unprotected-frames {
      type yang:counter64;
      config false;
      description
        "Total number of user frames sent that are not privacy
         protected. These frames are mapped to none and these
         frames are not MPPDU encapsulated.";
```

```
            reference
              "20.14.1 of IEEE Std 802.1AE";
          }
          leaf out-unprotected-octets {
            type yang:counter64;
            config false;
            description
              "Total number of user octets sent that are not privacy
               protected. These octets are from the frames that are
               mapped to none and these frames are not MPPDU
               encapsulated.";
            reference
              "20.14.1 of IEEE Std 802.1AE";
          }
        }
      }
    }
  }
}
```

23.11.3 ieee802-dot1x-eapol YANG module

```
module ieee802-dot1x-eapol {
  yang-version 1.1;
  namespace "urn:ieee:std:802.1X:yang:ieee802-dot1x-eapol";
  prefix eapol;

  import ietf-interfaces {
    prefix if;
  }
  import ieee802-dot1x {
    prefix dot1x;
  }
  import ieee802-types {
    prefix ieee;
  }

  organization
    "Institute of Electrical and Electronics Engineers";
  contact
    "WG-URL: http://ieee802.org/1/
     WG-EMail: stds-802-1-l@ieee.org

     Contact: IEEE 802.1 Working Group Chair
     Postal: C/O IEEE 802.1 Working Group
           IEEE Standards Association
           445 Hoes Lane
           Piscataway, NJ 08855
           USA

     E-mail: stds-802-1-chairs@ieee.org";
  description
    "Augment to be added to 802.1X on next revision:
     Missing Eapol Address for 802.1X-2020";

  revision 2022-05-25 {
    description
      "The following reference statement identifies each referenced IEEE
       Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1X Port-Based Network Access Control:
       IEEE Std 802.1X-2020.";
  }

  augment "/if:interfaces/if:interface/dot1x:pae" {
    leaf eapol-group-address {
      type ieee:mac-address;
      description
        "The destination Group MAC Address used by this PAE
         when transmitting EAPOL frames.";
      reference
        "12.9, and Figure 12-3 of IEEE Std 802.1X";
    }
    description
      "The destination Group MAC Address augmentation for
       transmitting EAPOL frames.";
  }
}
```

Annex B

(informative)

Bibliography

Change Annex B, Bibliography as follows:

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed., Boston: Pearson Education Inc., 2004, ISBN 0-321-19368-7.

[B2] IEEE Std 802.11™, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.[18,19]

[B3] IEEE Std 802.1AS™, IEEE Standard for Local and metropolitan area networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.

[B4] IEEE Std 802.1AX™, IEEE Standard for Local and metropolitan area networks—Link Aggregation.

[B5] IEEE Std 802E™-2020, IEEE Recommended Practice for Privacy Considerations for IEEE 802® Technologies.

[B6] IETF RFC 2279, UTF-8, a Transformation format of ISO 10646, Yergeau, F., Jan. 1998.[20]

[B7] IETF RFC 3410, Introduction and Applicability Statements for Internet-Standard Management Framework, Case, J., Mundy, R., Partain, D., and Stewart, B., Dec. 2002.

[B8] IETF RFC 3748, Extensible Authentication Protocol (EAP), Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H., June 2004.

[B9] IETF RFC 4303, IP Encapsulating Security Payload (ESP), Kent, S., Dec. 2005.

[B10] IETF RFC 5116, An Interface and Algorithms for Authenticated Encryption, McGrew, D., Jan. 2008.

[B11] IETF RFC 6087, Guidelines for Authors and Reviewers of YANG Data Model Documents, Bierman, A., Jan. 2011.

[B12] IETF RFC 6241, Network Configuration Protocol (NETCONF), Enns, R., Bjorklund, M., Schoenwaelder, J., Bierman, A., June 2011.

[B13] IETF RFC 6242, Using the NETCONF Protocol over Secure Shell (SSH), Wasserman, M., June 2011.

[18] IEEE publications are available from The Institute of Electrical and Electronics Engineers (https://standards.ieee.org/).
[19] The IEEE standards or products referred to in this annex are trademarks of The Institute of Electrical and Electronics Engineers, Inc.
[20] IETF RFCs are available from the Internet Engineering Task Force (https://www.ietf.org/rfc.html).

[B14] IETF RFC 6536, Network Configuration Protocol (NETCONF) Access Control Model, Bierman, A., Bjorklund, M., March 2012.

[B15] ISO/IEC/IEEE 8802.2, ISO/IEC/IEEE International Standard — Information Technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical Link Control.[21]

[B16] McGrew, D., "Generation of Deterministic Initialization Vectors (IVs) and Nonces," Oct. 2013.[22]

[B17] McGrew, D. A., and J. Viega, "The Galois/Counter Mode of Operation (GCM)," 31 May 2005.[23]

[B18] MEF 16, Ethernet Local Management Interface (E-LMI).[24]

[B19] Seaman, M., "The XPN recovery algorithm," June 2012.[25]

[21] ISO/IEC documents are available from the International Organization of Standardization (https://www.iso.org/) and from the International Electrotechnical Commission (http://www.iec.ch). These documents are also available from the American National Standards Institute (https://www.ansi.org/).

[22] Available at https://tools.ietf.org/html/draft-mcgrew-iv-gen-03.

[23] A prior revision of this document was the normative reference for GCM in IEEE Std 802.1AE-2006, but has been superseded by NIST SP 800-38D for that purpose. It does contain additional background information and can be downloaded from https://pdfs.semanticscholar.org/114a/4222c53f1a6879f1a77f1bae2fc0f8f55348.pdf.

[24] MEF standards are available from the MEF Forum (https://www.mef.net).

[25] Available at https://www.ieee802.org/1/files/public/docs2012/aebw-seaman-xpn-recovery-0612-v02.pdf.

Annex D

(normative)

PICS Proforma for an Ethernet Data Encryption device[26]

Change D.5 as follows:

D.5 EDE type and common requirements

Item	Feature	Status	References	Support
EDEM	Does the implementation conform to the specification for an EDE-M?	O.1	5.2(a)(b)(c), 5.5, 5.6, 15.6, D.7	Yes [ ]
EDECS	Does the implementation conform to the specification for an EDE-CS?	O.1	5.2(a)(b)(d), 5.5, 5.7, 11.2, D.7	Yes [ ]
EDECC	Does the implementation conform to the specification for an EDE-CC?	O.1	5.2(a)(b)(e), 5.5, 5.8, 15.6, D.7	Yes[ ]
EDESS	Does the implementation conform to the specification for an EDE-SS?	O.1	5.2(a)(b)(f), 5.5, 5.9, 15.7, D.7	Yes [ ]
TWOP	Does the EDE have two and only two externally accessible ports identified as red-side and black-side?	M	5.5(a)	Yes [ ]
SECY	Is an MKA-capable PAE associated with each SecY?	M	5.5(b)	Yes [ ]
PRY	Does the EDE provide MAC Privacy protection?	O	5.5(c)	Yes [ ] | No [ ]
SECB	Does the EDE incorporate a SecY in the black-side port interface stack?	EDEM:M	5.6(b)	Yes [ ]
SECP	Does the EDE incorporate a SecY in each Provider Edge Port interface stack?	EDECS:M EDECC:M EDESS:M	5.7(b), 5.8(b), 5.9(b)	Yes [ ]

[26]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

Change D.8 as follows:

D.8 EDE-CC Configuration

Item	Feature	Status	References	Support	
CC	Does the EDE-CC comprise two C-VLAN components internally connected as specified in 15.6?	EDECC:M	5.8(a), 15.6	Yes [ ]	
CCadd	Can the PAE supporting each PEP's SecY use the EDE-CC PAE group address to transmit and receive EAPOL frames		5.8(c), 15.6	Yes [ ]	
CCftr	Does the EDE's edge component relay entity filter frames whose destination MAC address is a C-VLAN component Reserved Address or the EDE-CC PAE group address?		5.8(d), 15.6	Yes [ ]	
CCrlyu	Are frames received untagged on the red-side port and relayed through the black-side port transmitted untagged?	EDECC AND ¬PRY:M	5.8(e), 15.6	Yes [ ]	
CCrlypu	Can the EDE be configured to relay frames received untagged on the red-side port through the black-side port untagged?	EDECC AND PRY:M	5.8(f), 15.6	Yes [ ]	
CCrlyt	Are frames received C-tagged on the red-side port and relayed through the black-side port transmitted C-tagged with the received C-VID?	EDECC AND ¬PRY:M	5.8(e), 15.6	Yes [ ]	
CCrlypt	Can the EDE be configured to relay frames received C-tagged on the red-side port through the black-side port C-tagged with the received C-VID?	EDECC AND PRY:M	5.8(f), 15.6	Yes [ ]	
CCrlypm	Can the EDE be configured to use the C-VID of a frame received on the red-side port to determine the C-VID when relaying that frame through the black-side port?	EDECC AND PRY:O	5.8(g), 15.6	Yes [ ]	No [ ]

Annex G

(informative)

Change the title of Annex G as follows:

SecY Management and MIB revisions

Insert the following new annex.

Annex H

(normative)

PICS proforma for MAC Privacy protection[27]

H.1 Introduction

The supplier of a protocol implementation which is claimed to conform to this standard's provisions for MAC Privacy protection shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

a) By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight.
b) By the supplier and acquirer—or potential acquirer—of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma.
c) By the user—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSs).
d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

H.2 Abbreviations and special symbols

H.2.1 Status symbols

M	mandatory
O	optional
O.n	optional, but support of at least one of the group of options labeled by the same numeral is required
X	prohibited
pred:	conditional-item symbol, including predicate identification: see H.3.4
¬	logical negation, applied to a conditional item's predicate

H.2.2 General abbreviations

N/A	not applicable
PICS	Protocol Implementation Conformance Statement

[27] *Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

H.3 Instructions for completing the PICS proforma

H.3.1 General structure of the PICS proforma

The first part of the PICS proforma, implementation identification and protocol summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire, divided into several subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No), or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked.)

Each item is identified by an item reference in the first column. The second column contains the question to be answered; the third column records the status of the item—whether support is mandatory, optional, or conditional; see also H.3.4. The fourth column contains the reference or references to the material that specifies the item in the main body of this standard, and the fifth column provides the space for the answers.

A supplier may also provide (or be required to provide) further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled *Ai* or *Xi,* respectively, for cross-referencing purposes, where *i* is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format and presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformation Statement for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation’s configuration capabilities, in case that makes for easier and clearer presentation of the information.

H.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but that have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

H.3.3 Exception information

It may occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer will be found in the Support column for this: instead, the supplier shall write the missing answer into the Support column, together with an *Xi* reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this standard.

NOTE—A possible reason for the situation described above is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

H.3.4 Conditional status

H.3.4.1 Conditional items

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply—mandatory or optional—are dependent upon whether or not certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the "Not Applicable" answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form "**pred:** S" where **pred** is a predicate as described in H.3.4.2 below, and S is a status symbol, M or 0.

If the value of the predicate is True (see H.3.4.2), the conditional item is applicable, and its status is indicated by the status symbol following the predicate: the answer column is to be marked in the usual way. If the value of the predicate is False, the "Not Applicable" (N/A) answer is to be marked.

H.3.4.2 Predicates

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: the value of the predicate is True if the item is marked as supported, and is False otherwise;

b) A predicate-name, for a predicate defined as a Boolean expression constructed by combining item-references using the Boolean operator OR: the value of the predicate is True if one or more of the items is marked as supported;

c) A predicate-name, for a predicate defined as a Boolean expression constructed by combining item-references using the boolean operator AND: the value of the predicate is True if all of the items are marked as supported;

d) The logical negation symbol "¬" prefixed to an item-reference or predicate-name: the value of the predicate is True if the value of the predicate formed by omitting the "¬" symbol is False, and vice versa.

Each item whose reference is used in a predicate or predicate definition, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

H.4 PICS proforma for IEEE Std 802.1AE MAC Privacy protection

H.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) of machines and/or operating system names	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirement for full identification. NOTE 2—The terms *Name* and *Version* should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

H.4.2 Protocol summary, IEEE Std 802.1AE MAC Privacy protection

Identification of protocol specification	IEEE Std 802.1AEdk-2023, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security : MAC Privacy protection
Identification of amendments and corrigenda to the PICS proforma which have been completed as part of the PICS	Amd. : Corr. : Amd. : Corr. :
Have any Exception items been required? (See H.3.3: the answer Yes means that the implementation does not conform to IEEE Std 802.1AE.)	No [ ] Yes [ ]

Date of Statement	

H.5 Mandatory capabilities

Item	Feature	Status	References	Support
Does the implementation of each MAC Privacy protection entity support:				
PrySAPs	The PrivatePort and Controlled Ports as specified in Clause 20?	M	5.10 a), Clause 20	Yes [ ]
PrySTAT	The MAC status and point-to-point parameters for the Private Port as specified in 6.4, 6.5, and 20.4?	M	5.10 b), 20.4	Yes [ ]
PryENC	MPPDU encoding constraints as specified in 19.6?	M	19.6	Yes [ ]
PryTX	Transmission from the Private Port as specified in 20.5 through 20.10?	M	5.10 d), 20.5, 20.6, 20.7, 20.8, 20.9, 20.10	Yes [ ]
PryPP	Per priority transmitted user data frame privacy selection (unprotected, as an individual Privacy Frame, or in a Privacy Channel)?	M	5.10 e), 20.5	Yes[ ]
PryUNP	Unprotected user data frame transmission, with the ability to manage transmission access priority?	M	5.10 f), 20.6	Yes [ ]
PryPF	Individual Privacy Frame transmission, with the ability to manage transmission access priority?	M	5.10 g), 20.7	Yes [ ]
PryCH	Configuration of up to two Privacy Channels?	M	5.10 h), 20.8	Yes [ ]
PryGEN	Generation of Privacy Channel MPPDUs as specified?	M	5.10 i), 20.9	Yes [ ]
PryGEND	Use of the default Privacy Channel MPPDU generation algorithm?	M	5.10 j), 20.9.4	Yes [ ]
PryCHA	Accepting user data frames for Privacy Channel encapsulationas specified in 20.10?	M	5.10 k), 20.10	Yes [ ]
PryCHE	Use of the default Privacy Channel MPPDU encapsulation algorithm?	M	5.10 l), 20.10.1	Yes [ ]
PryDA	Transmit all MPPDUs using the same destination MAC Address?	M	5.10 m), 18.1	Yes [ ]
PryPAE	Use of the PAE Group Address used by MKA as the destination MAC Address of MPPDUs?	M	5.10 n), 17.4, 18.1, 21.1.1	Yes [ ]
PrySRC	Use of the MAC Address used to generate MACsec SCI(s) as the source address of transmitted MPPDU when directly supported by a SecY?	M	5.10 o), 18.1	Yes [ ]
PryDAC	Administrative configuration of the destination MAC Address used to transmit MPPDUs?	M	5.10 p), 20.14	Yes [ ]
PryRX	Recognize received MPPDUs as specified?	M	5.10 r), 20.11	Yes [ ]
PryRXV	Validate and extract received MPPDU components as specified?	M	5.10 s), 20.12, 19.7	Yes [ ]
PryRE	Support simultaneous reassembly of Express and Preemptable frames?	M	5.10 t), 20.13	Yes [ ]

H.5 Mandatory capabilities *(continued)*

Item	Feature	Status	References	Support
PryRDV	Deliver user data frames received unfragmented and fragmented in the order specified?	M	5.10 u), 20.13	Yes [ ]
PryRED	Use of the default algorithm to reassemble frames from a single peer using one or two Privacy Channels?	M	5.10 v), 20.10.1	Yes [ ]

H.6 Optional capabilities

Item	Feature	Status	References	Support
Does the implementation of each MAC Privacy protection entity support:				
PryREP	Simultaneous frame reassembly from both Express and Preemptable Frame Fragments from more than one peer PrY?	O	5.11 a), Clause 20	Max peers: _____
PrySPS	Stream-based Privacy Selection?	O	5.11 b), 20.5.2	Yes [ ] No [ ]

Insert the following new annex.

Annex I

(informative)

Privacy considerations in bridged networks

This informative annex describes privacy considerations related to the use, design, and deployment of bridged networks based on IEEE Std 802.1Q and related standards (IEEE Std 802.1X, IEEE Std 802.1AB, IEEE Std 802.1AE, IEEE Std 802.1AR, IEEE Std 802.1AS, IEEE Std 802.1AX, IEEE Std 802.1BA, IEEE Std 802.1BR, IEEE Std 802.1CB, and IEEE Std 802.1CM).

The unintentional or unauthorized disclosure of personal information arises from a combination of the following factors:

a) The use of personal devices that are attached to, or form part of, the network (I.1)
b) The type of information that adversaries might wish to acquire (I.2)
c) The efficient operation and management of the network (I.3)
d) The use of security protocols for authentication, authorization, integrity, and confidentiality (I.5)
e) The frame fields that contain information useful to an adversary, the sophistication of, and the network access afforded to, that adversary (I.5)

Privacy considerations particular to a given referenced standard are discussed in I.6.

This annex is informative. It does not modify the mandatory or optional provisions or the recommendations contained in any referenced standard.

I.1 Personal devices

Privacy, in the context of bridged networks, relates to the use of personal devices i.e., devices used by one person or a small group of people. Information that identifies a personal device or is associated with that device identification can thus yield information about the location and activities of a person.

Shared service devices, in contrast, support applications for a large enough group of people such that correlation between any given person and the observable behavior of the device is weak. Other devices, e.g., sensors in industrial networks, have no direct correlation with a person.

In general IEEE 802.1 standards are applicable to both personal devices and shared service and other devices. However some protocol roles, e.g., Grandmaster in IEEE Std 802.1AS Timing and Synchronization for Time-Sensitive Applications, are unlikely to be associated with personal devices in other than the smallest bridged networks, and are even more unlikely to be associated with mobile personal devices.

I.2 Goals of adversaries

An adversary can be interested in the following personal information:

a) Who is using a personal device (identification)
b) Where are they (location, and location tracking)
c) What are they doing (activity, application use)
d) With whom are they associated (communicating, shared interest)

The information on all, or indeed on any of these, need not be complete to be useful to an adversary. The adversary can, for example, be interested in facts such as:

— An identified person appears to be engaging in the same, unknown, activity as a group of unknown persons at another identified location.
— There appears to be no one at an identified location.

The information obtained need not be particularly accurate to be useful to an adversary. It is sufficient that the cost of acquiring the information is less than the benefit expected from its use, allowing for the probability that it is incorrect and any costs associated with the use of incorrect information. Use of incorrect information can negatively affect a targeted person.

I.3 Network operation

Bridged networks support frame based transmission, with variable length frames and without requiring attached stations (except for certain time-sensitive network applications) to adhere rigidly to a clocked transmission schedule. Stations are not obliged to transmit when there is nothing to transmit and frames are not all padded to the same length, so the use of network resources benefits from statistical multiplexing. At the same time some network applications have requirements for timely delivery that cannot be met simply by relying on that multiplexing and increasing transmission speeds but require signaling, to bridges in the network, of the differential service requirements of individual frames. Time-sensitive network applications with more stringent delivery requirements require bandwidth allocation, supported by end station protocols or management configuration, and sufficient information in individual frames for bridges to associate each frame with an allocation (and thus with an individual end station and a particular type of end station application). Bridged networks provide more bandwidth than is available from each of their constituent individual LANs by restricting data frames to paths to their intended destinations. One of a number of alternate paths to a given destination end station or set of end stations can be used to further increase the available bandwidth, but common network application frame ordering requirements constrain the distribution of frames amongst such paths to those that bridges can distinguish as belonging to separate application flows.

Bridges in the network can distinguish between application flows using each frame’s destination MAC address (DA), source MAC address (SA), the VLAN identifier (VID) and priority code point (PCP encoded in the VLAN tag (if present), the EtherType (or LSAP) identifying the higher layer protocol conveyed by the frame, and the initial fields of that protocol. Protocols that operate over the bridged network and are used by personal devices to support network applications and to communicate with application servers and other devices (as opposed to reserving network resources for that communication) typically use the Internet Protocol (IP). It is rare for two personal devices to communicate without transmitting frames via one or more intervening routers. Any given IP subnet is often supported by a single VLAN, so bridges that support parallel paths for routed application flows from individual end stations typically use the source and destination IP addresses, the conveyed protocol type (IP, UDP, or SCTP), and source and destination ports for that protocol (see 9.1.5 of IEEE Std 802.1CB-2017).

Some protocols, e.g., IEEE Std 802.1Q Stream Reservation Protocol (SRP), transmit frames with a group destination MAC address that identifies the type of the intended recipient protocol entity and allows bridges to use address filtering to restrict those frames to an appropriate scope, reaching only the nearest bridge, for example. Some group addresses support a particular type of application, and thus associate the source MAC address (and the station that is using it) with that application.

The deployment and operational costs of bridged networks have been considerably reduced by the use of protocols that volunteer device information (e.g., IEEE Std 802.1AB) even when those protocols are not used to support full ‘plug-and-play’ operation. Management and troubleshooting of faulty devices or apparently incorrect network behavior depends on the recording of device location and gathering statistics on network use. Stations implementing protocols whose operation depends on the presence of a reachable

collaborating peer (e.g., IEEE Std 802.1AS gPTP time synchronization) typically advertise their capabilities, either by using IEEE Std 802.1AB or by sending their own messages.

I.4 Network security and privacy

As described above (I.3) efficient use of network resources, particularly for data frames that require other than best effort delivery, depends on the bridges in the network being able to identify end stations and (for some applications) service characteristics (priority, bandwidth and delay) required by their network applications. Where physical access and attachment to the whole or part of a bridged network is restricted to authorized personnel, confidentiality protection can be limited to that provided by higher layer protocols, notably TLS or IPsec. This leaves all the identifying information specified in IEEE 802.1 standards exposed to an adversary that does gain access to the network media.

The end stations and bridges in bridged networks are typically connected by IEEE Std 802.3 Ethernet links. MACsec (IEEE Std 802.1AE) can be used to provide both confidentiality and integrity protection hop by hop, leaving (in the most common configuration) just the MAC source and destination addresses, frame length, and frame transmission timing visible to an adversary with access to the network media. MACsec adds fields to each frame, but an adversary can recover the original frame length. MACsec operation can affect frame timing, but implementations suitable for use in time-sensitive networks impose a small fixed delay so as not to degrade the operation of IEEE Std 802.1AS time synchronization or IEEE Std 802.1Q timing gates supporting traffic shaping and bandwidth allocation. The frame to frame timing relationships that an adversary might observe remain unaltered. Where MACsec is used with Ethernet frame preemption and in-order delivery of preemptable and (separately) of preempting frames is enforced, an observer can distinguish these two classes of frames. Privacy considerations particular to IEEE Std 802.1X (Port-Based Network Access Control) support of MACsec are described below (I.6.2).

Unlike IEEE Std 802.11 operation in which a mobile end station participates in observable protocol to discover and select a suitable service it is rare for an end station to be connected to an Ethernet link that does not provide the expected service. Authentication, authorization, and confidentiality protection of subsequent data frames, if required, typically occur before additional end station information is disclosed. For exceptions see I.6.2, I.6.3.

I.5 Privacy exposures

A personal device can be identified explicitly by a single frame field, notably by using a universal MAC address as the source address of transmitted frames.

A station can use a locally assigned MAC address, as described in IEEE Std 802 and IEEE Std 802c. These addresses can be chosen randomly, or explicitly assigned by a higher layer protocol. A local assignment can be drawn from the entire local address space, or from one of the subsets described in IEEE Std 802c. However once a local MAC address has been assigned to a station and is being used to support higher layer protocols (such as IP), to restrict data frames to the path to that station, and to reserve resources in bridges along the path, any further MAC address change can be expected to interrupt or degrade the MAC Service. Moreover the disappearance of one address coupled with rapid appearance of another facilitates correlation of the two addresses and cannot be expected to reduce an adversary’s ability to infer information from the frame fields and other characteristics of persistent flows.

Where an individual frame field does not directly identify a personal device, either persistently as in the case of a universal MAC address or temporarily while the device is continuously active, an adversary can correlate those frame fields and other frame characteristics to identify (to an acceptable probability) the frames and frame flows associated with a single device and even to ascribe a permanent identity to that device or the particular network applications and activities supported by the device. Such a correlation is called a ‘fingerprint’, and the process of obtaining it ‘fingerprinting’. Fingerprinting does not necessarily

require a detailed understanding of the protocols used by a device, but can use general correlation and machine learning techniques to find any persistent pattern in the behavior of a device. Indeed a fingerprint can use device characteristics, such as the persistent scheduling of a transmission by one activity immediately after transmission for another activity, that do not appear in protocol specifications. In the absence of information that all the personal devices of a given type in widespread use consistently use the same network applications in the same way (and consequently exhibit indistinguishable network behavior) it has to be assumed that devices and activities can be distinguished by a sufficiently interested adversary.

The pattern of frame sizes transmitted and received by a personal device can fingerprint application activity and reveal details of that activity. The Ethernet MAC does, however, impose a minimum frame size, and higher layer protocols include fields that allow them to determine the applicable data length. To support Ethernet bridging of frames to and from media without the minimum size requirement, MACsec can encode the short length of those frames, but short frames that have been padded prior to being protected with MACsec appear to be of uniform length, thus depriving an adversary of the opportunity of fingerprinting application types using the small frame sizes that can be used in initial capability advertisement.

NOTE 1—Frame size patterns have been used to identify banking applications for specific financial institutions, approximate account balances, and whether money is being added to or removed from the account.

Static personal devices, e.g., desktop computers and home routers, typically connect to a bridged network using an individual wired IEEE Std 802.3 Ethernet connection. An adversary that can gain access to that wired connection has usually already identified (knowledge of home occupancy, etc.) the person or people associated with such a device and there is no question of tracking device movement. However the pattern of device activity (e.g., turning on security cameras when there is nobody at home) can reveal important personal location information.

NOTE 2—This annex does not detail privacy exposures resulting from media access control method operation, but notes that they can exist. For example, PoE (Power over Ethernet) use can reveal the identity and software version of some consumer electronics devices even when the adversary is restricted to observing the neighboring electromagnetic field.

Bridged networks are typically intraconnected with Ethernet links. Where these are wholly on private premises, access by an adversary can be prevented or at least made so difficult and expensive as to limit the targets to previously identified persons. Where personal device traffic to and from those private premises passes through an IP router, the privacy considerations are those applicable to the use of IP.

NOTE 3—At the time of preparation of this annex, discussion of the extension of TSN capabilities beyond the scope of bridged networks to the use of IP under the heading of ‘DetNet’ (deterministic networks) was still at an early stage. The privacy impacts of explicit flow identification and resource allocation described in this annex can be expected to apply.

IEEE Std 802.11, non-standard wireless connectivity, and in-home electrical power wiring can also be used to connect devices to personal bridged networks and to connect bridges within those networks. Where IEEE Std 802.11 is used to connect to an access point (AP) operating as an IP router, the security considerations applicable to IEEE Std 802.11 and IP apply. Where non-standard wireless connectivity and electrical power wiring are used, an adversary located sufficiently close as to be able to intercept the wireless signal or access power wiring outside possibly secured premises can be assumed to have access to MAC address, frame size, and frame timing information at a minimum with the further possibility of access to all the resource allocation and flow identification information conveyed. Frames with specific group and individual MAC addresses can be filtered by bridges in the network and do not necessarily traverse those links.

An adversary with management access to bridges in the network has access to resource allocation and flow identification information, but not (at least with standardized objects) the sizes of specific frames and their transmission timing. Such adversaries can include organizations that have a business relationship with the targeted person and are considered trustworthy by that person.

I.6 Standard specific considerations

This clause (I.6) summarizes particular ways in which each of the bridged network related standards can, when supporting personal devices, expose information that can be used to fingerprint the device's identity or use of network applications. Unless otherwise stated the general considerations described above (I.3, I.4, I.5) also apply to the use of each standard. The brief summary of each standard's capabilities is intended to provide the context for privacy considerations, and is not a substitute for the text of each referenced standard.

I.6.1 IEEE Std 802.1Q Bridges and bridged networks

The general considerations described above (I.3, I.4, I.5) all apply to the use of IEEE Std 802.1Q.

I.6.2 IEEE Std 802.1X Port-Based Network Access Control

IEEE Std 802.1X specifies a general method controlling access to a network, both by systems that are the source and destination of frames carried by the network and by relay systems that are to be connected to multiple other systems in the network and that forward frames between those connections. In both cases each of the system's ports either participates in a mutual authentication exchange with the neighboring system or proves the success of past authentication and authorization to access the network. This clause discusses potential privacy exposures arising from the use of the media-independent capabilities of IEEE Std 802.1X with Ethernet, for privacy considerations related to the use of IEEE 802.11 connections to or within bridged networks see IEEE Std 802.11.

Extensible Authentication Protocol (EAP, IETF RFC 3748 [B8]) messages are encapsulated in EAP over LANs (EAPOL) PDUs so they can be sent between a Supplicant port (also referred to as a Peer in IETF EAP RFCs), that wishes to gain access to the network, and an Authenticator port, on a system that provides network access. EAP is an authentication framework, not a specific authentication mechanism, and more than 40 specific authentication methods have been defined. An Authenticator is typically supported by an Authentication Server (AS) that executes the particular method or sequence of methods selected. The authentication credentials supported by different methods can differ, as can the degree to which they expose the identity claimed by a Supplicant. EAP messages between the Authenticator and the Authentication Server are typically encapsulated in the RADIUS (IETF RFC 3579) or Diameter (IETF RFC 4072) protocols. IEEE Std 802.1X mandates the use of mutual authentication methods, and requires support for EAP-TLS (IETF RFC 5216) if integration with the use of IEEE Std 802.1AR is claimed.

Following EAP authentication, RADIUS or Diameter server can provide the Authenticator with attributes that include access controls appropriate to the authorization accorded to the Supplicant and information that supports subsequent reattachment of a device to the network without repetition of the full authentication exchange and authorization process. These attributes can include persistent identifiers, e.g., the EAP-Key-Name (the IEEE 802.1X secure Connectivity Association Key Name, CKN) and Network-Id-Name (2.2 and 2.7 of IETF RFC 7268). Privacy considerations relating to communication between the Authenticator, the Authentication Server, a RADIUS or Diameter Server, and any Online Certificate Status Protocol (OCSP) Server are described in the relevant IETF RFCs.

If data transmission, following successful authentication and authorization, between the Supplicant and Authenticator ports is protected by MACsec, the MACsec Key Agreement protocol (MKA) is used to distribute the succession of Secure Association Keys (SAKs) used to provide confidentiality and integrity protection. MKA uses keys derived from a secure Connectivity Association Key (CAK) and the CKN to integrity protect MKPDUs and to confidentiality and integrity protect (using AES Key Wrap) distributed SAKs. The contents of MKPDUs (other than distributed keys) are not confidentiality protected to support network monitoring and debugging without needing to share the CAK or derived keys. The CAK and CKN can be derived from an EAP authentication or can be pre-shared by other means, including local device

management. The initial octets of each MKPDU contain the CKN, so a peer MACsec capable system knows which (if any) of its keys to use to verify that the MKPDU has been transmitted by a previously authenticated system. A device can be configured to attempt, or require, EAP authentication each time it is connected to the network, thus obtaining a fresh CAK and CKN. Shared service infrastructure devices typically need to be capable of restoring connectivity to their neighbors without re-authentication, since neither they nor their neighbors are guaranteed to have connectivity to an Authentication Server or other supporting services.

EAPOL frames, and integrity protected MKPDUs which are carried in EAPOL frames, can convey network announcements (IEEE Std 802.1X). These can be used by personal devices, but are expected to be transmitted by shared service devices.

I.6.3 IEEE Std 802.1AB Station and Media Access Control Connectivity Discovery

The Link Layer Discovery Protocol (LLDP) allows a station to advertise, to others attached to the same LAN, the station's management address and major capabilities. The receiving stations allow management access to received LLDP information to support network topology discovery and configuration checking. The point of LLDP would be lost if the advertised attributes were to be temporary or unavailable to intended recipients. Standard attributes include a system name and description. The range of attributes has been extended by other standards and organizations such as equipment suppliers.

LLDP is a one way protocol: it does not contain mechanisms for soliciting or confirming receipt of information. The destination address of each LLDPDU is usually one of the reserved group addresses specified in IEEE Std 802.1Q and filtered by bridges to limit the scope of its propagation through the network. This filtering allows a management application to use the information received by end stations and bridges in the network to build a map of the network topology. The filtering also restricts exposure of any station's advertised attributes to adversaries that have access to the individual LANs traversed by the LLDPDUs that station transmits, or that have management access to their recipients or to the management application. IEEE Std 802.1AB-2016 mandates support for the Nearest Bridge group address (01-80-C2-00-00-0E, also referred to as the Individual LAN Scope group address). This address is filtered by all bridges.

Where port access is controlled by IEEE Std 802.1X, IEEE Std 802.1AB mandates Controlled Port support for LLDP exchanges, thus providing confidentiality (on the LAN) if MACsec is used. Unprotected transmission using the Uncontrolled Port is permitted.

I.6.4 IEEE Std 802.1AE MAC Security

The exposure of personal information, including information that can contribute to fingerprinting a device or activity, conveyed in frames that are confidentiality protected by MAC Security (MACsec) can be reduced as described above (I.4). The potential exposure of personal device information by the supporting IEEE Std 802.1X MACsec Key Agreement protocol (MKA) is discussed in I.6.2.

MACsec protects communication between neighboring systems, but the scope of that protection depends on what each system considers to be a potential neighbor. By default frames conveyed by the IEEE Std 802.1X Port Access Control Protocol (PACP) that encapsulates the Extensible Authentication Protocol (EAP, IETF RFC 3748) are transmitted to the Nearest non-TPMR Bridge group address (also referred to as the IEEE Std 802.1X PAE address), so any intervening TPMR cannot access confidentiality protected frame fields (see I.4). However MACsec can also be used to secure a point-to-point connection across a Provider Bridge Network exposing any priority information required by PBN systems to provide the desired class of service, and to secure connectivity where the PBN uses VLAN tag information to select a provider service instance (15.4 and 15.5 of IEEE Std 802.1AE). Where a Provider Backbone Bridge (PBB) is used, the source MAC address of the originator of the frame is encapsulated and confidentiality protected. A PBB is not, itself, likely to be a personal device.

MAC Privacy protection (Clause 17) can be used in conjunction with MACsec to further address Privacy aspects of Ethernet frames discussed in I.2. Specifically MAC Privacy protection addresses these issues as outlined in (17.2) by hiding user data frame attributes such as:

a) Source and destination MAC addresses.

b) Frame sizes.

c) Frame transmission timing.

I.6.5 IEEE Std 802.1AR Secure Device Identity

IEEE Std 802.1AR specifies Secure Device Identifiers (DevIDs) for use with IEEE Std 802.1X and other industry authentication, provisioning, and authorization protocols. Privacy consideration for use of DevIDs are discussed in 6.5 of IEEE Std 802.1AR-2018.

I.6.6 IEEE Std 802.1AS Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks

The generalized precision time protocol (gPTP), state machines, and algorithms specified in IEEE Std 802.1AS support time-sensitive applications such as audio, video, and time-sensitive control, by maintaining synchronized time across packet networks, including bridged networks, comprising interconnected time-aware systems. Each time-aware system exchanges messages with its immediate neighbor to measure the link propagation delay experienced by packets forwarded by that neighbor. Time-aware end stations receive time information, either directly or indirectly via one or more time-aware relay systems, from a grandmaster that is the source of time information in a network domain. Each system adjusts the time information received to account for the link propagation delay, and in the case of time-aware relays for the residence time of the information in the relay prior to forwarding. The current grandmaster, and the port used to receive information from that grandmaster, is selected by a best master clock algorithm (BMCA) that constructs a time-synchronization spanning tree throughout the network domain with a spanning tree priority vector that allows each time-aware system to select its best port for receiving (and in the case of a time-aware relay, as the basis for forwarding) timing information.

Each time-aware system port that supports gPTP is identified by a sourcePortIdentity, comprising a clockIdentity and a portNumber. The clockIdentity identifies the clock being used by a specific time-aware bridge or end station for a particular instance of distributed time and is constructed using an NUI-48 or NUI-64 (see IEEE Std 802c): i.e., while it is an identifier and not a protocol address it is constructed in the same way as a MAC address, is intended to be unique within a network, and it is possible to tell by examining one of the bits derived from the NUI in the construction (the bit corresponding to the U/L bit when the an NUI is used as a MAC Address) whether the clockIdentity is intended to be locally or globally unique.

The media-independent specification of gPTP is supported by media-dependent procedures. Neighboring time-aware systems connected by full-duplex point-to-point links, such as those specified by IEEE Std 802.3, use gPTP messages to measure the propagation delay and convey timing information. Each message includes the transmitter’s sourcePortIdentity. If the connection is confidentiality protected by MACsec, this message field is only visible to the communicating systems.

Neighboring IEEE Std 802.11 stations, whether AP capable or not, do not use gPTP messages to measure propagation delay and convey timing. They use the IEEE 802.11 MAC Layer Management Entity (MLME) which generates, timestamps, and consumes measurement frames to provide timing information.

NOTE 1—For privacy considerations related to the IEEE 802.11 MLME see IEEE Std 802.11.

The BMCA spanning tree conveys a trace of each port’s sourcePortIdentity on the best path (for timing distribution) from each potential grandmaster. A personal device attached to the network is thus aware of,

and receives a permanent identifier for each system that is part of, that path. The BMCA protocol does not propagate path information in the reverse direction (i.e., towards the grandmaster root of a timing tree): the sourcePortIdentity of a personal device that has a single port attached to the network is only conveyed to its immediate neighbor.

NOTE 2—While use of the redundant grandmasters and the BMCA allows the precision timing service provided by IEEE Std 802.1AS to be resilient in the face of system and link failures, it is highly desirable that the network remain stable and the standard provides priority values for grandmaster selection and timing path selection that discriminate against devices that are not permanently part of the network and powered on. A personal device, and particularly a mobile personal device, is therefore unlikely to find itself in the position of propagating BMCA path trace information including the sourcePortIdentity of one of its ports, even if it has more than one port.

Time-aware stations connected by media for which gPTP is supported by media-independent procedures send Signaling messages (10.4 of IEEE Std 802.1AS-2011) that signal each station's ability to participate in the protocol together with station dependent parameters that control aspects of protocol operation (e.g., message interval request). For stations connected by IEEE Std 802.11 media, this capability is provided by the IEEE 802.11 MLME.

I.6.7 IEEE Std 802.1AX Link Aggregation

Link Aggregation allows parallel point-to-point links to be aggregated to form a Link Aggregation Group (LAG) that is treated as a single link. A bridge or end station port generally distributes frames amongst the links so as to preserve frame ordering within flows (see I.3). The distribution algorithm and parameters for its use can be specified by using Conversation-sensitive Collection and Distribution (CSCD). A further capability Distributed Resilient Network Interface (DRNI), that provides system level redundancy by allowing two cooperating systems to mimic the behavior of a single system terminating a LAG, is unlikely to be used by personal devices.

The addition and removal of links to and from a LAG is facilitated by the operation of the Link Aggregation Control Protocol (LACP) in each of the systems they connect. To ensure that the candidate links for a given LAG do connect the same pair of systems, LACP exchanges a System Identifier that is a combination of System Priority and System MAC Address. This System MAC Address needs to be unique amongst any set system capable of aggregating links with each other, but does not have to be globally unique and is not necessarily (except for any conditions imposed to avoid the profligate assignment of unique identifiers) the address used as a source MAC address by transmitted frames originating from the system. Other LACP parameters, because of potential system to system differences, can contribute to system fingerprinting though not in such a clear way. LACPDUs are transmitted to a group address selected to limit their propagation within the network, typically the Nearest non-TPMR Bridge group address (01-80-C2-00-00-03).

When MACsec is used to protect communication between neighboring systems, the MAC Security Entity is instantiated in the interface stack associated with each of the individual aggregatable links (see 11.5) and thus can confidentiality protect both the conversations carried over those links and operation of LACP.

I.6.8 IEEE Std 802.1BA Audio Video Bridging (AVB) Systems

IEEE Std 802.1BA specifies the selection of specific features and options from IEEE Std 802.1Q, IEEE Std 802.1AS, and LAN MAC/PHY standards that facilitate manufacture of AVB-capable components. A person not skilled in networking can use those components to build networks that provide working audio and video services. This standard does not introduce additional privacy considerations beyond those inherent in the referenced standards.

I.6.9 IEEE Std 802.1BR Virtual Bridged Local Area Networks—Bridge Port Extension

IEEE Std 802.1BR specifies a method for increasing the effective geographical extent of the control parameters of a single bridge by supporting multiple instances of the Enhanced Internal Sublayer Service, each associated with a single bridge port, over a single LAN connected to an External Bridge Port Extender that can support one or more ports attached to LANs, each serving a single end station, and zero or more ports connected to further External Bridge Port Extenders. Bridge Port Extenders also support frame replication for multicast. While Bridge Port Extenders extend the effective extent of a single bridge they do require port extender specific configuration to support time-sensitive network flows.

Bridge Port Extenders were standardized to meet data center bridging requirements and are not expected to be personal devices or to provide services directly to personal devices.

I.6.10 IEEE Std 802.1CB Frame Replication and Elimination for Reliability

Frame Replication and Elimination for Reliability (FRER) increases the probability that any given packet will be delivered by replicating each of an identifiable sequence of packets, transmitting the replicates on disjoint network paths, and eliminating duplicates where those paths meet. The sequence of replication, replicate transmission, and duplicate elimination, can be repeated between transmission by the original source of the packets and reception by the eventual destination(s). Resources can be reserved on each of the paths that support duplicate transmission so that TSN delivery objectives (timeliness and extremely low loss) can be met even if LANs or relay systems fail (on all but one of the potential paths between the original source and a destination).

FRER requires, at a minimum, the addition of a sequence number to each packet. IEEE Std 802.1CB specifies a redundancy tag (R-TAG) that adds just that sequence number, and also allows use of the High-availability Seamless Redundancy (HSR) sequence tag or the Parallel Redundancy Protocol (PRP) sequence trailer both specified by IEC 62439-3:2016 (7.8, 7.9, and 7.10 of IEEE Std 802.1CB-2016). None of these contribute significantly to an adversary’s ability to assign each packet to a stream or flow as is necessary, using the contents of other frame fields, by bridges and end stations supporting resource allocation and FRER. IEEE Std 802.1CB does not specify positioning of its processing relative to that carried out by the IEEE Std 802.1AE MAC Security Entity in interface stacks, but the usual considerations place the latter closer to the PHY. MACsec confidentiality protection, where used, applies to the FRER tags and trailer just as it would to the stream and flow identifying frame fields.

While IEEE Std 802.1CB addresses the requirements addressing from industrial networks it can be used to support personal devices.

I.6.11 IEEE Std 802.1CM Time-Sensitive Networking for Fronthaul

IEEE Std 802.1CM specifies the selection of specific features and options from IEEE Std 802.1Q, IEEE Std 802.1AC, IEEE Std 802.3, IEEE Std 1588, ITU-T G.8275.1, ITU-T G.8261, ITU-T G.8262, ITU-T G.8262.1, and ITU-T G.8264, to enable the transport of time-sensitive fronthaul streams in Ethernet bridged networks. This standard does not introduce additional privacy considerations beyond those inherent in the referenced standards.

IEEE SA
STANDARDS ASSOCIATION

RAISING THE WORLD'S STANDARDS

Connect with us on:

Twitter: twitter.com/ieeesa
Facebook: facebook.com/ieeesa
LinkedIn: linkedin.com/groups/1791118
Beyond Standards blog: beyondstandards.ieee.org
YouTube: youtube.com/ieeesa

standards.ieee.org
Phone: +1 732 981 0060

IEEE