IEEE STANDARDS ASSOCIATION

IEEE Standard for
Local and metropolitan area networks—

Media Access Control (MAC) Security

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 802.1AE™-2018
(Revision of IEEE Std 802.1AE-2006)

IEEE Std 802.1AE™-2018
(Revision of IEEE Std 802.1AE-2006)

IEEE Standard for
Local and metropolitan area networks—

Media Access Control (MAC) Security

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 27 September 2018

IEEE-SA Standards Board

Abstract: How all or part of a network can be secured transparently to peer protocol entities that use the MAC Service provided by IEEE 802® LANs to communicate is specified in this standard. MAC security (MACsec) provides connectionless user data confidentiality, frame data integrity, and data origin authenticity.

Keywords: authorized port, confidentiality, data origin authenticity, IEEE 802.1AE™, IEEE 802.1AEbn™, IEEE 802.1AEbw™, IEEE 802.1AEcg™, integrity, LANs, local area networks, MAC Bridges, MAC security, MAC Service, MANs, metropolitan area networks, port-based network access control, secure association, security, transparent bridging

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

PDF: ISBN 978-1-5044-5215-1 STD23339
Print: ISBN 978-1-5044-5216-8 STDPD23339

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

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

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute 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 Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. 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.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. 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: PROCUREMENT OF 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 should be considered 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, or 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 his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide 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 revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

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.

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 the IEEE-SA Website at http://ieeexplore.ieee.org or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

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.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the IEEE 802.1 Working Group had the following membership:

Glenn Parsons, *Chair*
John Messenger, *Vice Chair*
Mick Seaman, *Security Task Group Chair, Editor*

SeoYoung Baek
Shenghua Bao
Jens Bierschenk
Steinar Bjornstad
Christian Boiger
Paul Bottorff
David Chen
Feng Chen
Weiying Cheng
Rodney Cummings
János Farkas
Norman Finn
Geoffrey Garner
Eric W. Gray
Craig Gunther
Marina Gutierrez
Stephen Haddock
Mark Hantel
Patrick Heffernan
Marc Holness
Lu Huang
Tony Jeffree
Michael Johas Teener
Hal Keen
Stephan Kehrer
Philippe Klein
Jouni Korhonen
Yizhou Li
Christophe Mangin
Tom McBeath
James McIntosh
Tero Mustala
Hiroki Nakano
Bob Noseworthy
Donald R. Pannell
Walter Pienciak
Michael Potts
Karen Randall
Maximilian Riegel
Dan Romascanu
Jessy V. Rouyer
Eero Ryytty
Soheil Samii
Behcet Sarikaya
Frank Schewe
Johannes Specht
Wilfried Steiner
Patricia Thaler
Paul Unbehagen
Hao Wang
Karl Weber
Brian Weis
Jordon Woods
Nader Zein
Helge Zinner
Juan Carlos Zuniga

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

Thomas Alexander
Richard Alfvin
Amelia Andersdotter
Butch Anton
Harry Bims
Demetrio Bucaneg
Stephen Bush
William Byrd
Radhakrishna Canchi
Steven Carlson
Keith Chow
Charles Cook
Richard Doyle
János Farkas
Norman Finn
Michael Fischer
Yukihiro Fujimoto
Randall Groves
Qiang Guo
Stephen Haddock
Marco Hernandez
Werner Hoelzl
Russell Housley
Yasuhiro Hyakutake
Noriyuki Ikeuchi
Atsushi Ito
Raj Jain
Sangkwon Jeong
Piotr Karocki
Stuart Kerry
Yongbum Kim
Hyeong Ho Lee
Suzanne Leicht
Jon Lewis
Elvis Maculuba
Ignacio Marin Garcia
Brett McClellan
Richard Mellitz
John Messenger
Michael Montemurro
Rick Murphy
Nick S. A. Nikjoo
Satoshi Obara
Robert O'hara
Bansi Patel
Clinton Powell
Adee Ran
Karen Randall
R. K. Rannow
Alon Regev
Maximilian Riegel
Robert Robinson
Benjamin Rolfe
Jessy V. Rouyer
Richard Roy
Naotaka Sato
Mick Seaman
Thomas Starai
Walter Struppler
Jasja Tijink
Mark-Rene Uchida
Dmitri Varsanofiev
George Vlantis
Lisa Ward
Stephen Webb
Karl Weber
Chun Yu Charles Wong
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 27 September 2018, it had the following membership:

Jean-Philippe Faure, *Chair*
Gary Hoffman, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Ted Burse
Guido R. Hiertz
Christel Hunter
Joseph L. Koepfinger*
Thomas Koshy
Hung Ling
Dong Liu
Xiaohui Liu
Kevin Lu
Daleep Mohla
Andrew Myles
Paul Nikolich
Ronald C. Petersen
Annette D. Reilly
Robby Robson
Dorothy Stanley
Mehmet Ulema
Phil Wennblom
Philip Winston
Howard Wolfman
Jingyi Zhou

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.1AE-2018, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Security.

The first edition of IEEE Std 802.1AE was published in 2006. The first amendment, IEEE Std 802.1AEbn™-2011, added the option of using the GCM-AES-256 Cipher Suite. The 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. The 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™.

This revision, IEEE Std 802.1AE-2018, incorporates the text of IEEE Std 802.1AE-2006 and amendments IEEE Std 802.1AEbn-2011, IEEE Std 802.1AEbw-2013, and IEEE Std 802.1AEcg-2017.

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

IEEE Std 802.1X™-2010 specifies Port-based Network Access Control, provides a means of authenticating and authorizing devices attached to a Local Area Network (LAN), and includes the MACsec Key Agreement protocol (MKA) necessary to make use of IEEE Std 802.1AE.

IEEE Std 802.1AE is not intended for use with IEEE Std 802.11™. 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

1. Overview

1.1 Introduction

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.

MAC Security (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

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

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.

1.2 Scope

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 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 Management Information Base (MIB) module for managing the operation of MAC Security in TCP/IP networks.
n) Specifies requirements, criteria, and choices of Cipher Suites for use with this standard.

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

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used; therefore, 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.

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]

[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).

[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

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[8]

access priority: The priority associated with a transmit request made by a MAC Security Entity (SecY) at its Common Port.

Association Number (AN): A number that is concatenated with the Secure Channel Identifier (SCI) to identify a Secure Association (SA).

black-side: Identifies the Ethernet Data Encryption device (EDE) port that uses MACsec to protect transmitted frames and verify received frames.

bounded receive delay: A guarantee that a frame will not be delivered after a known bounded time.

NOTE—In the case of the MAC Service, this is typically assumed to be less than two seconds.

Bridged Local Area Network: A concatenation of individual IEEE 802 Local Area Networks (LANs) interconnected by MAC Bridges.

NOTE—Unless explicitly specified the use of the word *network* in this standard refers to a Bridged Local Area Network. The term *Bridged Local Area Network* is not otherwise abbreviated. The term *Local Area Network* and the abbreviation *LAN* are used exclusively to refer to an individual LAN specified by a MAC technology without the inclusion of Bridges. This precise use of terminology within this specification allows a Bridged Local Area Network to be distinguished from an individual LAN that has been bridged to other LANs in the network. In more general usage such precise terminology is not required, as it is an explicit goal of this standard that MAC Security is transparent to the users of the MAC Service.

Cipher Suite: A set of one or more algorithms, designed to provide any number of the following: data confidentiality, data authenticity, data integrity.

Common Port: An instance of the MAC Internal Sublayer Service (ISS) used by the SecY to provide transmission and reception of frames for both the controlled and uncontrolled ports.

Controlled Port: The access point used to provide the secure MAC Service to a client of a MAC Security Entity (SecY).

cryptographic key: A parameter that determines the operation of a cryptographic function such as

a) The transformation from plain text to cipher text and vice versa
b) Synchronized generation of keying material
c) Digital signature computation or validation.[9]

cryptographic mode of operation: An algorithm for the cryptographic transformation of data that features a symmetric key block cipher algorithm. *Syn:* **mode.**[10]

[8] *IEEE Standards Dictionary Online* is available at https://dictionary.ieee.org.

[9] This and some other definitions in this clause have been drawn from ASC TR1/X9, Technical Report for ABA ASC/X9 Standards Definitions, Acronyms, and Symbols, 2002.

[10] This and some other definitions in this clause have been drawn from NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, 2001.

Customer Edge Port: The red-side port of an Ethernet Data Encryption device (EDE-CS, EDE-CC, or EDE-SS).

NOTE—The terms *customer* and *provider* applied to the external and internal ports of an EDE-CS are those used by IEEE Std 802.1Q in its description of Provider Edge Bridges (PEBs) and reflect the role of those ports in the layered network architecture. They do not indicate control or ownership of the equipment. In this standard it is convenient to extend the use of those terms to ports that have the same relative relationship to the edge and network components of an EDE-CC or EDE-SS. This is not a suggestion that further variants of PEBs and Backbone Edge Bridges (BEBs) be specified, as the existence of additional variants would complicate interoperability, service provision, and the task of this standard.

Customer Network Port: 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.

data integrity: A property whereby data has not been altered in an unauthorized manner since it was created, transmitted or stored.[11]

edge component: The bridge component in an Ethernet Data Encryption device (EDE-CS, EDE-CC, or EDE-SS) that is attached to the red-side port.

Ethernet Data Encryption device (EDE): A two-port bridge that transmits and receives frames that are assumed to be unprotected to and from one red-side port, and conditionally relays those frames to and from its other black-side port, protecting and verifying frames transmitted and received on the black-side port using MACsec.

IEEE 802 Local Area Network (LAN): LAN technologies that provide a MAC Service equivalent to the MAC Service defined in IEEE Std 802.1AC. IEEE 802 LANs include IEEE Std 802.3 (Ethernet) and IEEE Std 802.11 [B2] (Wireless).

NOTE—IEEE 802 LANs are also referred to in the text of this standard simply as LANs.

initialization vector (IV): A vector used in defining the starting point of an encryption process within a cryptographic algorithm.[12]

integrity: *See:* **data integrity.**

integrity check value (ICV): A value that is derived by performing an algorithmic transformation on the data unit for which data integrity services are provided. The ICV is sent with the protected data unit and is recalculated and compared by the receiver to detect data modification.

key: *See:* **cryptographic key.**

key management: The generation, storage, distribution, deletion, archiving, and application of keys in accordance with a security policy.

Layer Management Interface (LMI): The interface between a protocol entity in a system and the system management, providing for the exchange of parameters with other system entities that are not attached to the service access points used and provided by the protocol entity.

Local Area Network (LAN): *See:* **IEEE 802 Local Area Network (LAN).**

[11] This and some other definitions in this clause have been drawn from NIST Special Publication 800-57, Recommendation for Key Management, 2005.

[12] This and some other definitions in this clause have been drawn from Federal Information Processing Standard (FIPS) 140-2, Security Requirements for Cryptographic Modules, 2001.

MAC Security Entity (SecY): The entity that operates the MAC Security protocol within a system.

MAC Security TAG (SecTAG): A protocol header, comprising a number of octets and beginning with an EtherType, that is prepended to the service data unit supplied by the client of the protocol and is used to provide security guarantees.

MAC service data unit (MSDU): A sequence of zero or more octets that compose the data to be communicated with a single MAC Service request or indication.

master key: A secret key that is used to derive one or more cryptographic keys that are used directly to protect data transfer.

message authentication: If the message arrives authenticated, the cryptographic guarantee is that the message was not modified in transit and that the message originated from an entity with the proper cryptographic credentials.

mode: *See:* **cryptographic mode of operation.**

multipoint: Involving or potentially involving more than one participant in the role of receiver, or in the role of transmitter, in a single data transfer or set of related data transfers.

network component: The bridge component in an Ethernet Data Encryption device (EDE-CS, EDE-CC, or EDE-SS) that is attached to the black-side port.

nonce: A non-repeating value, such as a counter, used in key management protocols to thwart replay and other types of attack.[13]

Packet Number (PN): A monotonically increasing value that is guaranteed unique for each MACsec frame transmitted using a given Secure Association Key (SAK).

Port Identifier: A 16-bit identifier that uniquely identifies each of a system's transmit Secure Channels (SCs) that uses the same MAC address as a component of its Secure Channel Identifier (SCI).

NOTE—The Port Identifier is not constrained to correspond to any other port number, identifier, or index. There can be more than one SC for a physical port, identifying frames transmitted by separate virtual ports, and more than one SC for a physical or virtual port if that port uses different SCs to transmit frames of different priorities.

Protocol Data Unit (PDU): A unit of data specified in a protocol and consisting of protocol information and, possibly, user data.

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

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

red-side: Identifies the Ethernet Data Encryption device (EDE) port that does not use MACsec to protect transmitted frames or verify received frames.

Reserved Address: A group address filtered by a bridge component to restrict the scope of the control protocols using that Destination Address (DA).

[13] This and some other definitions in this clause have been drawn from ASC TR1/X9, Technical Report for ABA ASC/X9 Standards Definitions, Acronyms, and Symbols, 2002.

secret key: A cryptographic key used with a secret key cryptographic algorithm that is uniquely associated with one or more entities and should not be made public.[14]

Secure Association (SA): A security relationship that provides security guarantees for frames transmitted from one member of a Connectivity Association (CA) to the others. Each SA is supported by a single secret key, or a single set of keys where the cryptographic operations used to protect one frame require more than one key.

Secure Association Identifier (SAI): An identifier for an Secure Association (SA), comprising the Secure Channel Identifier (SCI) concatenated with the Association Number (AN).

Secure Association Key (SAK): The secret key used by a Secure Association (SA).

Secure Channel (SC): A security relationship used to provide security guarantees for frames transmitted from one member of a Connectivity Association (CA) to the others. An SC is supported by a sequence of Secure Associations (SAs) thus allowing the periodic use of fresh keys without terminating the relationship.

Secure Channel Identifier (SCI): A unique identifier for a Secure Channel (SC), composed of a MAC Address and a Port Identifier.

NOTE—Key agreement protocols such as the MACsec Key Agreement protocol (MKA) (see IEEE Std 802.1X) are responsible for ensuring that each SCI used with a given Secure Association Key (SAK) is unique where a Cipher Suite requires that for nonce construction, as does the Default Cipher Suite (14.5). SCI uniqueness does not rely on MAC Address allocation procedures.

secure Connectivity Association (CA): A security relationship, established and maintained by key agreement protocols, that comprises a fully connected subset of the service access points in stations attached to a single Local Area Network (LAN) that are to be supported by MACsec.

Short Secure Channel Identifier (SSCI): A 32-bit value, managed by the key agreement protocol, that is unique for each SCI within the context of all MAC Security Entities (SecYs) using a given Secure Association Key (SAK).

spoofing: Claiming a fraudulent identity for purposes of mounting an attack.

Uncontrolled Port: The access point used to provide the insecure MAC Service to a client of a MAC Security Entity (SecY).

user priority: The priority associated with a transmit request received by the Controlled Port of a MAC Security Entity (SecY).

[14] From FIPS 140-2.

4. Abbreviations and acronyms

The following abbreviations and acronyms are used in this standard.

AES	Advanced Encryption Standard
AN	Association Number
BEB	Backbone Edge Bridge
BPDU	Bridge Protocol Data Unit
CA	secure Connectivity Association
CFM	Connectivity Fault Management
CRC	Cyclic Redundancy Check
DA	Destination Address
EDE	Ethernet Data Encryption device
EDE-CC	Ethernet Data Encryption device with red-side recognition of C-TAGs and black-side addition and removal of C-TAGs
EDE-CS	Ethernet Data Encryption device with red-side recognition of C-TAGs and black-side addition and removal of S-TAGs
EDE-M	VLAN-unaware Ethernet Data Encryption device operating as a Customer Bridge
EDE-SS	Ethernet Data Encryption device with red-side recognition of S-TAGs and black-side addition and removal of S-TAGs
EISS	Enhanced Internal Sublayer Service
EPON	Ethernet Passive Optical Network
ES	end station
FCS	frame check sequence
GCM	Galois Counter Mode
ICV	integrity check value
ISS	Internal Sublayer Service
IV	initialization vector
KaY	MAC Security Key Agreement Entity
LACP	Link Aggregation Control Protocol
LAN	IEEE 802 Local Area Network
LLC	Logical Link Control (ISO/IEC/IEEE 8802.2™ [B10])
LLDP	Link Layer Discovery Protocol
LMI	Layer Management Interface
MAC	Media Access Control
MACsec	Media Access Control Security
MIB	Management Information Base
MKA	MACsec Key Agreement protocol (IEEE Std 802.1X)

MKPDU	MACsec Key Agreement Protocol Data Unit
MPDU	MACsec Protocol Data Unit
MSDU	MAC Service Data Unit
MSTP	Multiple Spanning Tree Protocol
OLT	Optical Line Terminal
ONU	Optical Network Unit
PAE	Port Access Entity
PCP	Priority Code Point (IEEE Std 802.1Q)
PDU	Protocol Data Unit
PEB	Provider Edge Bridge
PEP	Provider Edge Port
PN	Packet Number
QoS	Quality of Service
RSTP	Rapid Spanning Tree Algorithm and Protocol
SA	Secure Association or Source Address, as applicable
SAI	Secure Association Identifier
SAK	Secure Association Key
SC	Secure Channel
SCB	Single Copy Broadcast
SCI	Secure Channel Identifier
SecTAG	MAC Security TAG
SecY	MAC Security Entity
SL	Short Length
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
SSCI	Short Secure Channel Identifier
STP	Spanning Tree Protocol
UNI	User Network Interface (IEEE Std 802.1Q)

5. Conformance

A claim of conformance to this standard is a claim that the behavior of an implementation of a MAC Security Entity (SecY) meets the requirements of this standard 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 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 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).

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 proforma (see Annex A) reflects 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***.

5.2 Protocol Implementation Conformance Statement (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 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 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 802.1Q PICS proforma as required for a VLAN-unaware MAC Bridge.
d) For an EDE-CS: the IEEE 802.1Q PICS proforma as required for a Provider Edge Bridge.
e) For an EDE-CC: the IEEE 802.1Q PICS proforma as required for each of the two C-VLAN components.
f) For an EDE-SS: the IEEE 802.1Q PICS proforma as required for each of the two S-VLAN components.

5.3 MAC Security Entity requirements

An implementation of a MAC Security Entity (SecY) for which conformance to this standard is claimed shall

a) Support the Controlled and Uncontrolled Ports, and use a Common Port as specified in Clause 10.
b) Support the MAC status and point-to-point parameters for the Controlled and Uncontrolled Ports as specified in 6.4, 6.5, and 10.7.
c) Process transmit requests from the Controlled Port as required by the specification of Secure Frame Generation (10.5).
d) Process receive indications from the Common Port as required by the specification of Secure Frame Verification (10.6), prior to causing receive indications at the Controlled Port.
e) Encode and decode MACsec Protocol Data Units (MPDUs) as specified in Clause 9.
f) Use a 48-bit MAC Address and a 16-bit Port Identifier unique within the scope of that address assignment to identify each transmit SCI, as specified in 8.2.1.
g) Satisfy the performance requirements specified in Table 10-3 and 8.2.2.
h) Support the Layer Management Interface (LMI) operations required by the Key Agreement Entity as specified in Clause 10.
i) Provide the management functionality specified in 10.7.
j) Protect and validate MPDUs by using Cipher Suites as specified in 14.1.
k) Support Integrity Protection using the Default Cipher Suite specified in Clause 14.
l) For each Cipher Suite implemented, support a minimum of
    1) One receive SC
    2) Two receive SAKs
    3) One transmit SC
    4) One of the two receive SAKs at a time for transmission, with the ability to change from one to the other within the time specified in Table 10-3.
m) Specify the following parameters for each Cipher Suite implemented
    1) The maximum number of receive SCs supported
    2) The maximum number of receive SAKs
    3) The maximum number of transmit SCs supported.

An implementation of a SecY for which conformance to this standard is claimed shall not

n) Introduce an undetected frame error rate greater than that achievable by preserving the original frame check sequence (FCS), as required by 10.4.

o) Implement any Cipher Suite that is additional to those specified in Clause 14 and does not meet all the criteria specified in 14.2, 14.3, and 14.4.1.

p) Support access to MACsec parameters by a management agent using any version of SNMP prior to v3.

An implementation of a SecY for which full conformance to this standard is claimed shall not

q) Implement Cipher Suites other than those specified in Clause 14.

NOTE—Conformance with Cipher Suite variance is allowed, as specified in 5.4 and in 14.4.1.

5.4 MAC Security Entity options

An implementation of a MAC Security Entity (SecY) for which conformance to this standard is claimed may

a) Support access to MACsec parameters by a management agent using SNMP version v3 and the MIB module specified in Clause 13.

b) Support more than one receive SC.

c) Support more than two receive SAKs.

d) Support more than one transmit SC.

e) Support Confidentiality Protection using the Default Cipher Suite without a confidentiality offset, as specified in Clause 14.

f) Support Confidentiality Protection using the Default Cipher Suite with a confidentiality offset, as specified in Clause 14.

g) Include Cipher Suites that are specified in Clause 14 in addition to the Default Cipher Suite.

An implementation of a SecY that supports more than one transmit SC shall

h) Support a Traffic Class Table and an Access Priority Table as specified in 10.7.17.

An implementation of a SecY for which conformance with Cipher Suite variance is claimed may

i) Use Cipher Suites not specified in Clause 14, but meeting the criteria specified in 14.2, 14.3, and 14.4.1.

5.5 EDE conformance

Ethernet Data Encryption devices (EDEs) of various types comprise bridging systems and bridge components used as specified in this standard. Clause 15 provides a taxonomy, specification, and a description of the intended use of each type (EDE-M, EDE-CC, EDE-CS, or EDE-SS).

The bridging system and/or bridge components that comprise an implementation of an EDE that is claimed to conform to this standard shall conform to the requirements of (and may use any of the options and recommendations permitted by) IEEE Std 802.1Q, IEEE Std 802.1X, and the provisions of this standard for each MAC Security Entity (SecY) that is part of the EDE implementation, with the restrictions, additions, exceptions, and clarifications specified in this standard for that type of EDE.

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).

5.6 EDE-M conformance

An implementation of an EDE-M (15.2, 15.4) that is claimed to conform to this standard shall

a) Comprise a VLAN-unaware MAC Bridge as specified by IEEE Std 802.1Q (5.14 of IEEE Std 802.1Q-2018) with the constraints and exceptions specified in this standard.
b) Incorporate a SecY in the black-side port interface stack (15.2, 15.4).
c) Be capable of being configured to secure connectivity within a customer or provider network (as specified in 15.2), using the Nearest non-TPMR group address for group-addressed EAPOL PDUs, and filtering frames whose destination MAC Address is a TPMR component Reserved Address or the Nearest non-TPMR Bridge group address.
d) Be capable of being configured to secure connectivity across a PBN to a peer EDE-M, MACsec-capable Customer Bridge, or an EDE-CS (as specified in 15.4 and 15.5), using the Nearest Customer Bridge group address for group addressed EAPOL PDUs and filtering frames whose destination MAC Address is a C-VLAN component Reserved Address with the exception of the Nearest Customer Bridge group address.

and may

e) Be capable of being configured to secure connectivity across a PBN to a peer EDE-CC (as specified in 15.6), using the EDE-CC PAE group address for group addressed EAPOL PDUs, and filtering frames whose destination MAC Address is a C-VLAN component Reserved Address or the EDE-CC PAE group address.
f) Be capable of being configured to recover signaled priority from a C-VLAN tag and to priority tag (or not) frames transmitted by the black-side port as specified in 15.4.

5.7 EDE-CS conformance

An implementation of an EDE-CS (15.5) that is claimed to conform to this standard shall

a) Comprise a Provider Edge Bridge as specified by IEEE Std 802.1Q (5.10.2 of IEEE Std 802.1Q-2018) including one and only one C-VLAN component (identified by this standard as the edge component of the EDE) and an S-VLAN component (the network component of the EDE).
b) Incorporate a SecY in each of the internal Provider Edge Port interface stacks (15.5).
c) Be capable of being configured to use the Nearest Customer Bridge group address for group addressed EAPOL PDUs (15.5).

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 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.

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).

5.9 EDE-SS conformance

An implementation of an EDE-SS (15.7) that is claimed to conform to this standard shall

a) Comprise two S-VLAN components, each as specified by IEEE Std 802.1Q (5.6 of IEEE Std 802.1Q-2018)—an edge component and a network component—internally connected as specified in 15.7.

b) Incorporate a SecY in each of the internal Provider Edge Port interface stacks (15.7).

c) Be capable of being configured to use the EDE-SS PAE group address as the destination MAC address of group addressed EAPOL PDUs for group addressed EAPOL PDUs (as specified in 15.7).

d) Filter and not forward all frames whose destination MAC address is either one of the addresses identified by IEEE Std 802.1Q as an S-VLAN component Reserved Address (Table 8-2 of IEEE Std 802.1Q-2018) or the EDE-SS PAE group address.

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 S-tagged with the same S-VID if they were S-tagged on receipt (15.7).

6. Secure provision of the MAC Service

MACsec provides secure communications between stations that are attached to the same LAN. An authenticated and authorized peer MAC Security Entity (SecY) within each station uses the insecure MAC Service provided by the LAN to provide the secure MAC Service to its client (see Figure 6-1). The requirements for MACsec discussed in this clause are informed by the goal of preserving the MAC Service.

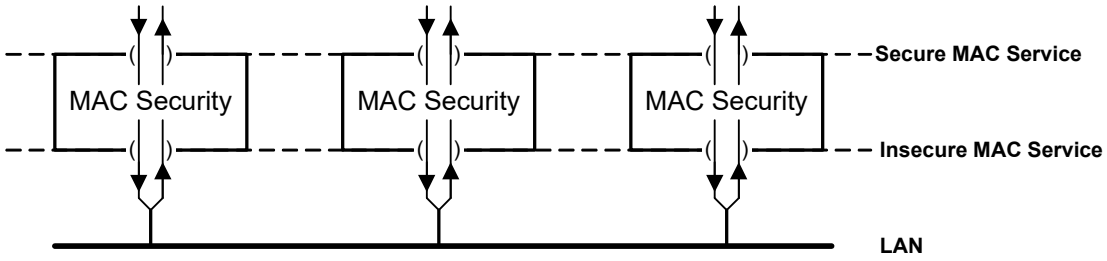


Figure 6-1—MACsec secured LAN with three stations

This clause discusses the

a) Primitives, parameters, connectivity, and status parameters provided by the MAC Service
b) Security threats posed by abuses of the MAC Service
c) Connectivity used and provided by the MACsec protocol
d) Service guarantees provided by MACsec and the security services they support
e) Quality of Service (QoS) issues addressed in the design, implementation, and use of MACsec.

NOTE 1—In order to introduce the concepts used in this standard, this clause can repeat or summarize the specification in other clauses; however, it contains no normative provisions that apply either to the subject matter of those other clauses or to the other standards referenced. For conformance to this standard see Clause 5.

NOTE 2—MACsec does not itself guarantee the security of a Bridged Local Area Network, as that security depends on the security of the individual LANs that comprise the network, on the policies adopted by clients of the secure MAC Service (7.3), and on the security of the MAC Bridges that interconnect those LANs.

NOTE 3—Authentication and authorization are outside the scope of this standard, which ensures secure communication between mutually authenticated and authorized service access points.

NOTE 4—The MAC Service and the secure MAC Service are provided at a service access point to a single client. The client is either a logical link control (LLC) Entity or an entity that in turn provides the MAC Service or a MAC Internal Sublayer Service (IEEE Std 802.1AC, IEEE Std 802.1Q).

6.1 MAC Service primitives and parameters

The MAC Service (IEEE Std 802.1AC) provides unconfirmed connectionless-mode data transfer between source and destination stations. The invocation of a request primitive at a service access point within a source station results, with a high probability, in a corresponding indication primitive at selected service access points in destination stations. A single service request at one service access point results in no more than one service indication at each of the other service access points.

Each request and indication primitive has the following four parameters:

— Destination Address
— Source Address
— Priority
— MAC Service Data Unit (MSDU)

The MAC Service can be provided by a single LAN or by a Bridged Local Area Network. The service provided to an LLC Client in an end station is specified in IEEE Std 802.1AC. The service provided by a LAN to a MAC Bridge is the MAC Internal Sublayer Service (ISS), which includes parameters necessary to the bridge relay function including the frame check sequence (FCS). Except as otherwise explicitly noted, the term *MAC Service* as used in the remainder of this clause refers both to the provision of the MAC Service to an LLC client and to provision of the ISS. Multiple instances of the MAC Service can be provided using a single instance of the ISS and supported in VLAN-aware Bridges using the Enhanced Internal Sublayer Service (EISS), (see 6.6 of IEEE Std 802.1Q-2018). When a VLAN TAG (IEEE Std 802.1Q) is used to distinguish the service instances supported, the additional parameters of the EISS are all encoded within the ISS MSDU.

NOTE 1—The MAC Service defined in IEEE Std 802.1AC is an abstraction of the features common to a number of specific media access control methods and is a guide to the development of client protocols.

NOTE 2—The priority parameter described in this clause is also referred to as the user_priority in some specifications. The functions that support the ISS can calculate an access_priority for use on a LAN in local support of the user_priority.

The MAC Service provided by a single LAN preserves the relative order of service requests and corresponding service indications with the same requested priority. Each instance of the MAC Service provided using an instance of the EISS preserves the relative order of requests and indications with the same destination address, source address, and priority if the destination address is an individual address, and the relative order of requests and indications with the same destination address and priority if the destination address is a group address.

NOTE 3—A Provider Bridged Network can use the EISS to provide instances of the MAC Service that appear, with the exception of ordering constraints, to be a single LAN and are used as such by MACsec.

The address and MSDU parameters delivered with a service indication have identical values to those supplied with the corresponding service request. The MAC Service does not validate the parameters supplied with the request; for example, it does not provide any assurance that the source address used by an LLC client is the individual address previously allocated to the station.

NOTE 4—For example, in the absence of policies that require authorization to use an address, or check that a MACsec participant does not change its address, the use of MACsec will not protect against ARP spoofing. MACsec can form part of a solution that prevents ARP spoofing provided that suitable client policies are used in conjunction with MACsec, as mentioned in 1.1, in Clause 6, and in Clause 7 (particularly 7.3). In the case of ARP spoofing, appropriate policies include a selection of not accepting frames from an end station with a different source address than that used in the authentication dialogue, not accepting frames from bridges that are not known to obey that rule, requiring use of device identification to ensure use of a legitimate MAC address, and filtering certain higher-layer protocol frames unless received (using MACsec) from a system allowed to send them.

The priority parameter delivered with a service indication has an identical value to that supplied with the corresponding service request, where the media access control method used supports communication of priority. The access to the LAN granted by the media access control method can take the requested priority value into account, but can also be based on other factors. The IEEE 802.3™ media access control method does not convey priority, and the priority value delivered with the service indication is determined by management of the receiving station. However, where the EISS is used to support an instance of the MAC Service, the priority parameter of the EISS is encoded within a VLAN TAG (IEEE Std 802.1Q) that forms the initial octets of the MSDU accompanying a service request to the instance of the ISS used to support the EISS. Figure 6-2 shows the priority field encapsulated in the VLAN TAG within the Secure Data portion of the MACsec frame. In this case, the value of the priority parameter delivered with the service indication will be identical to that provided with the corresponding request.

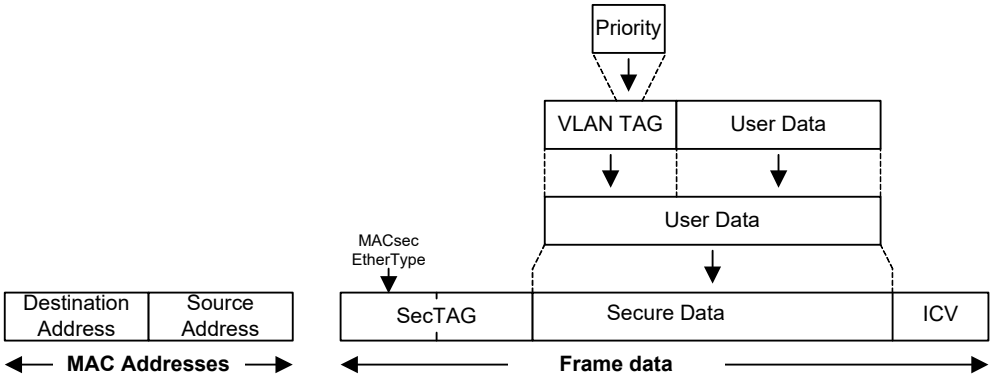


Figure 6-2—MACsec Frame, VLAN TAG, and QoS

6.2 MAC Service connectivity

The MAC Service provided by a single point-to-point or shared media LAN provides symmetric and transitive connectivity between all the stations connected to that LAN. Following a service request at one service access point, a corresponding service indication occurs, with high probability, at all the MAC ISS access points in other stations attached to the same LAN. Service indications at service access points that provide the MAC Service to an LLC Entity are filtered by media access functions, generally within each receiving station, to exclude frames that are not destined to an individual or group address used by the client.

NOTE 1—Symmetric connectivity means that if station A can communicate with station B, then B can also communicate with A. Transitive connectivity means that if station A can communicate with B, and B with C, then A can also communicate with C.

NOTE 2—Some media access control devices or methods are capable of not delivering some or all frames with unwanted destination addresses to some or all stations.

MAC Service clients and client protocols can operate incorrectly if the connectivity provided is not as expected. While some protocols can detect the lack of symmetric connectivity, they can simply deny service as a result. Protocols can operate inefficiently if transitive connectivity is not provided. While MAC Bridges can filter frames to restrict provision of service, the use of Virtual LANs (VLANs), with each VLAN providing full connectivity, is preferred to lessen the administrative burden of ensuring correct connectivity.

NOTE 3—The original Spanning Tree Protocol (STP) could create loops in the network if symmetric connectivity was not provided. The Rapid Spanning Tree Algorithm and Protocol (RSTP), (see IEEE Std 802.1Q), detects non-symmetric connectivity between Bridges, but will deny service until the problem is resolved, and intermittent non-symmetric connectivity can result in data loops. For example, the operation of the OSPF routing protocol on a LAN is inefficient unless all participants can receive frames sent by each other. If a LAN that provides the ISS to attached MAC Bridges merely delivers frames to their intended destination instead of providing full connectivity, learning of source addresses can be inhibited and frames flooded throughout the bridged network for an indefinite period.

6.3 Point-to-multipoint LANs

A point-to-multipoint LAN provides connectivity from a single distinguished station to one or more other stations, i.e., from one point to multiple other points, and from each of the other stations to the distinguished station. The point-to-multipoint LAN does not provide direct connectivity between pairs of stations unless the distinguished station is one of the pair. Efficient multicast and broadcast from the distinguished station to all the others is provided: a single service request with a given destination address can result in service indications at each multipoint station wishing to receive frames with that destination address. Communication between the other stations occurs via the distinguished station, as specified by the relevant standard for the particular technology.

NOTE—Examples of point-to-multipoint LANs include IEEE 802.3 Ethernet Passive Optical Network (EPON) (see Clause 12), IEEE Std 802.11™ [B2], and certain provider network VLAN configurations. Depending on the particular LAN or network technology the distinguished stations are variously referred to as optical line terminals (OLTs), wireless access points (WAPs), head-ends, concentrators, star concentrators, or root nodes, and the other stations as optical network units (ONUs), stations, modems, or leaf nodes.

6.4 MAC status parameters

Each service access point can make available status parameters that reflect the operational state and administrative controls for the service instance provided at that access point.

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.

The **MAC_Operational** parameter is TRUE if, and only if, service requests can be made and service indications can occur.

The value of the MAC_Enabled and MAC_Operational parameters are determined by the specific entity providing the MAC Service. IEEE Std 802.1AC and IEEE Std 802.1Q specify how that determination is made for provision of the insecure ISS by specific media access control methods, and for provision of the insecure EISS. This standard specifies how these parameters are determined for the secure MAC Service.

NOTE—Correct provision and use of the MAC_Operational parameter is essential for high performance implementation of RSTP (IEEE Std 802.1Q), Multiple Spanning Tree Protocol (MSTP, IEEE Std 802.1Q), and Link Aggregation Control Protocol (LACP, IEEE Std 802.1AX™ [B4]). In the absence of this parameter, loss of connectivity is determined by repetitive loss of protocol frames that are normally transmitted at intervals of a few seconds, and it is assumed that frames transmitted immediately after a medium availability transition have a high probability of not being received by protocol peers.

6.5 MAC point-to-point parameters

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.

If the **operPointToPointMAC** parameter is TRUE, the service is used as if it provides connectivity to at most one other system; if FALSE, the service is used as if it can provide connectivity to a number of systems.

The **adminPointToPointMAC** parameter can take one of three values. If it is

a) **ForceTrue**, operPointToPointMAC shall be TRUE, regardless of any indications to the contrary generated by the service providing entity.
b) **ForceFalse**, operPointToPointMAC shall be FALSE.
c) **Auto**, operPointToPointMAC is as currently determined by the service providing entity.

IEEE Std 802.1AC and IEEE Std 802.1Q specify how the point-to-point status determination is made for provision of the insecure ISS by specific media access control methods, and for provision of the insecure EISS. This standard specifies how it is determined for the secure MAC Service.

NOTE—RSTP and MSTP (IEEE Std 802.1Q) require the use of operPointToPointMAC to facilitate rapid reconfiguration in some network failure scenarios. LACP (IEEE Std 802.1AX [B4]) does not aggregate links that are not point-to-point and can use operPointToPointMAC to make this determination.

6.6 Security threats

The expected features of the MAC Service described in 6.1 through 6.5—the relationships between service requests and indications, preservation of the parameters of these primitives, the connectivity provided, and the relationship of the MAC status parameters to the connectivity—can be accidentally and unintentionally distorted through misconfiguration or deliberately abused. Misconfiguration or abuse can result in

a) Inability to issue service requests
b) Indiscriminate loss of service indications
c) Specifically targeted loss of service indications
d) Repeated service indications at the intended destinations
e) Service indications with modified address or data parameters
f) Additional service indications with unmodified or selectively modified parameters
g) Service indications at unintended recipients
h) Delayed service indications that can disrupt network operation
i) Disclosure of the MSDU to unauthorized parties.

Deliberate abuse can serve as a basis for an attack upon the resources accessible from a LAN through attacks on the protocols that use the service and provide access to or control over those resources. The effort required by an attacker to abuse the service in any particular way depends in general on the media access control method used by the LAN, and the particular devices and components that support it.

The MAC Service does not guarantee the origin or authenticity of service requests and the accompanying parameters. Since a station's sole use of its source address and the restriction of service indications to intended recipients can depend on cooperative behavior from other stations, it is usually easy for an attacker that can attach a station to a LAN to receive any service indication and to issue additional service requests with parameters based on those indications. Other service abuses can require physical access to inconveniently located components.

It is beyond the scope of this standard to enumerate all the ways in which abuses of the service can be exploited. They include techniques commonly referred to as passive wiretapping, masquerading, and man-in-the-middle attacks. The latter is facilitated by source address spoofing, usually after another station with that source address has been observed to have been granted access to some resource. Attacks can include

j) Denial of service, to all or to selected stations
k) Theft of service
l) Access to confidential information
m) Modification of confidential information
n) Access to or control over restricted resources.

MACsec does not protect against brute force denial of service attacks that can be mounted by abusing the operation of particular media access control methods through degrading the communication channel or transmitting erroneous media access method specific control frames.

6.7 MACsec connectivity

The connectivity provided (6.2) between the MAC Internal Sublayer Service (ISS) access points of stations connected to a single LAN composes an insecure association between communicating stations. Key agreement protocols as defined in IEEE Std 802.1X establish and maintain a secure Connectivity Association (CA), which is a fully (i.e., symmetric and transitive) connected subset of the ISS service access points. Each instance of MACsec operates within a single CA.

NOTE 1—ISO/IEC 15802, the MAC Service definition, introduces the term *Connectivity Association* to discuss the relationship between service access points without referring to the details of particular media access control methods or to terms such as *physical connection* or *logical connection* that have other associated attributes and meanings.

MACsec itself does not provide comprehensive monitoring of the connectivity provided by a CA, although it can detect and will signal certain failures to the local MAC Security Key Agreement Entity (KaY). Together, operation of key agreement protocols and MACsec ensures that the status parameters provided by an instance of the secure MAC Service correctly reflect both the current connectivity and changes in the connectivity of the CA. Specifically

a) MAC_Operational is only True if the CA is complete (i.e., is symmetric and transitive), and the local MACsec Entity (SecY) can both receive and transmit.

   NOTE 2—EPON is discussed in Clause 12.

b) If MAC_Operational is True in stations wanting to join a new CA and in stations already in the target CA, and if stations are added to the CA, MAC_Operational transitions to False in either all the stations originally participating in the CA or in all those added, for sufficient duration such that clients of the service are aware of the transition.

   Determining which group transitions MAC_Operational to False is outside the scope of this specification and is determined by the KaY and signaled through the LMI.

c) If MAC_Operational is False in stations wanting to join a CA, and if these stations are added to a CA, there is no change in the MAC_Operational status of the stations already in the target CA, and MAC_Operational will transition to True in the joining stations after some period of time. This is the typical case for a single station joining a CA, in which its MAC_Operational is False until the join is accomplished when its state transitions to MAC_Operational True.

d) If adminPointToPointMAC is set to Auto and MAC_Operational is True, then operPointToPointMAC is True only if at most one other station is participating in the CA. If adminPointToPointMAC is set to forceFalse, then operPointToPointMAC must be False, regardless of the number of stations in the CA.

NOTE 3—Communication between KaYs in stations that compose a CA does not depend on the operation of MACsec.

6.8 MACsec guarantees

At each service access point that is a member of a CA, MACsec ensures that any service indication

a) Is the result of a service request at a service access point that is also a member of the same CA
b) Has the parameter values that are identical to those supplied with the service request

and can also ensure that

c) No more than one indication results from one service request
d) A service indication does not occur after a known bounded time has elapsed since the service request was made
e) The values of the octets that comprise the MAC Service Data Unit (MSDU) parameter cannot be ascertained except by members of the CA.

MACsec does not

f) Conceal the following from stations that are not members of the CA:
   1) Service requests
   2) Values of service request address parameters
   3) The number of octets that comprises the MSDU
g) Validate the parameters provided with a service request.

MACsec provides guarantees to within known bounds that are derived from the cryptographic methods and other mechanisms used.

The known bounded time in item d) is typically longer than required to enforce the maximum transit delay requirements of the MAC Service.

NOTE—The addition of explicit time indications to the MAC Security TAG (SecTAG) to provide tight bounds for transit delay was considered in the development of this standard, but the value delivered is small for the added complexity and the burden imposed on key agreement protocols. Higher-layer protocols that have tight timing requirements typically add their own timing markers. As these markers are carried within the MSDU, their integrity is protected by MACsec.

6.9 Security services

The guarantees provided by MACsec support the following security services for stations participating in MACsec:

a) Connectionless data integrity [6.8a), 6.8b)]
b) Data origin authenticity [6.8a)]. If the connectivity model is point-to-point, the originator is authenticated, but if the connectivity model is multipoint, then the authenticated originator is any member of the CA, rather than a particular individual station.
c) Confidentiality [6.8d)]
d) Replay protection [6.8c), 6.8d)]
e) Bounded receive delay

and can be used to limit the nature and extent of

f) Denial of service attacks.

MACsec does not support

g) Non-repudiation (ISO/IEC 7498-2)

or protect against

h) Traffic analysis (ISO/IEC 7498-2).

A MAC Bridge that forwards a frame with an erroneous source MAC address can unintentionally facilitate a denial of service or other attack on other LANs within a Bridged Local Area Network. The MAC Bridge should use MACsec in conjunction with an appropriate policy to verify the binding of the source MAC address to access to resources.

6.10 Quality of Service maintenance

The quality of the MAC Service can be lowered by direct attacks on the operation of particular media access control methods and indirect attacks on their resource allocation procedures facilitated by masquerading and unauthorized data modification. MACsec does not provide guarantees for frames, known as MAC control frames, that are internal to the operation of a particular media access method and cannot defend against abuses that use or affect such frames. MAC control frames are not forwarded by MAC Bridges, so attacks that exploit them can be localized to particular LANs.

NOTE—Where, within the operation of a particular media access control method, it is possible to establish secure Connectivity Associations (CAs) prior to performing certain control functions, those functions should be supported by frames transmitted using an instance of the ISS. The parameters of those frames can then be protected by MACsec, and the scope for abuse restricted. It is not a requirement of the Open Systems Interconnection (OSI) layer model (ISO 7498) that management and control of a particular layer be carried out purely within that layer by protocols whose identifiers and formats are specific to that layer. For example, SNMP can be used to manage MAC Bridges.

Operation of a security protocol has the potential to lower some aspects of QoS. The operation and design of MACsec is discussed below as it relates to

a) Service availability
b) Frame loss
c) Frame misordering
d) Frame duplication
e) Frame transit delay
f) Frame lifetime
g) Undetected frame error rate
h) Maximum service data unit size supported
i) Frame priority
j) Throughput.

Use of MACsec can lower service availability if delays occur in the creation of Connectivity Associations or in the distribution and maintenance of cryptographic keying material. Failures or attacks upon the system that supports authentication and authorization can result in denial of service.

The operation of MACsec introduces no additional frame loss on individual LAN segments other than that expected for a specific media access control method as a consequence of a small increase in frame size. The operation of MACsec between two customer systems across a provider bridged network can introduce additional frame loss caused by possible frame reordering from expedited forward or link aggregations within the provider bridged network. The reception of misordered frames can cause MACsec implementations to discard additional frames depending upon the configuration of replay protection parameters. MACsec can use separate secure channels to transit frames with different access priorities and thus reduce or eliminate undesirable frame discard resulting from the mutual reordering of those frames.

Conforming implementations of MACsec are capable of applying a new keying material starting with any frame in a sequence that is received with the minimum intervening spacing specified by the specific media access control method in use. Each frame protected by MACsec remains independent of its predecessors and successors, so loss of a single frame does not imply loss of additional frames.

MACsec does not introduce any additional potential for duplicating or misordering frames. No retransmission mechanisms are added to relax requirements for distribution and use of MACsec related information. MACsec implementations are required to preserve the sequence of requests and indications between the secure service access point supported and the insecure service access point used.

Since the MAC Service is provided at an abstract interface within an end station, it is not possible to specify the total frame transit delay precisely. It is, however, possible to measure the additional transit delay introduced by an additional component or intermediate system.

The minimum additional transit delay introduced by MACsec is due to the increase in the MSDU size required to convey security information and essential buffering requirements required to meet the processing requirements of particular Cipher Suites. Specific limits are placed on the additional delays allowed to MACsec implementations (Table 10-3). The permitted delay is short compared with the upper bound mandated by the MAC Service, so it does not threaten the correct operation of higher-layer protocols.

Frame lifetime can be increased by MACsec if additional delay is introduced by providing security. The typical bound on frame lifetime is approximately two seconds.

MACsec does not increase the undetected frame error rate for frames received and transmitted on a single LAN. The frame check sequence (FCS) method used by each specific media access control method protects the entire frame including information added by MACsec. The integrity check added by MACsec can increase the probability of detecting unintentional frame modifications, particularly where those do not correspond to the expected noise characteristics for which the FCS was originally designed, but equally is not a substitute for the FCS since it is designed to ensure that an attacker has an exceedingly low chance of predicting how to make an undetected modification to the frame's parameters rather than to efficiently detect burst noise characteristics.

Use of MACsec on each of a MAC Bridge's Ports will force a change in the data covered by an FCS, even if the frame is being relayed between LANs that use the same media access control method. Application of the techniques described in Annex O of IEEE Std 802.1Q-2018 allow an implementation to achieve an arbitrarily small increase in undetected frame error rate, even in cases where the data that is within the coverage of the FCS is changed.

The MSDU size that can be supported by an IEEE 802 LAN varies with the media access control method and its associated parameters (speed, electrical characteristics, etc.), and can be constrained by the owner of the LAN. MACsec adds security information to a transmitted MSDU, and thus the secure MAC service offers a smaller MSDU size than the insecure MAC service that it employs.

Where MACsec is used to support an instance of the ISS that in turn supports the EISS, encoding of the priority parameter of the EISS within the ISS MSDU ensures that priority can be communicated unchanged between service access points that are attached to a single LAN. Since MACsec is terminated at each of those service access points, a bridge that makes use of that ISS instance to support one of its ports can access or change the priority even if the two instances of MACsec encrypt the MSDU in order to provide confidentiality. An Access Priority Table (10.7.17) can be used to derive the access priority requested from the medium supporting the ISS from the priority requested by the user of the EISS.

Cryptography can be computationally intensive, and the operation of MACsec has the potential to lower throughput. The Cipher Suite(s) mandated and recommended by this standard have been chosen, in part, for their ability to support economic implementation across the range of LAN MAC data rates.

7. Principles of secure network operation

This clause establishes the principles and a model of secure network operation. It describes the security relationships used to support the secure MAC Service (Clause 6), and how that service is used to provide overall network security. It provides the context necessary to understand the operation of the MAC Security protocol (MACsec) (see Clause 8) and individual MAC Security Entities (SecYs) (see Clause 10).

Secure network operation comprises use of the secure MAC Service on each of the individual LANs that compose the network together with the application of appropriate security policies by the MAC Service clients in end stations and in intermediate systems that forward frames. This clause defines

a) The security relationships that support secure MAC Service

and describes how the secure MAC Service is

b) Supported on each of the individual LANs that compose the network (7.1)
c) Used by the protocol entities that are its Clients (7.3)

and delineates the responsibilities of the

d) Key Agreement Entities (KaYs, IEEE Std 802.1X)
e) MAC Security Entities
f) Clients of the secure MAC Service.

Security relationships and the terms that identify them have been defined, in various ways, by a number of publicly available documents. This standard has deliberately chosen new terms to minimize confusion whenever differences could exist between previously used terms and the requirements. For example, the attributes associated with an Secure Association Identifier (SAI) (see Figure 7-7) are similar to but not exactly the same as those associated with the Security Parameter Index (SPI) defined by IPsec (IETF RFC 4303 [B8]). The normative properties of all terms used in this standard are as defined by this standard.

NOTE 1—The use of the term *secure network* in this clause refers to a network of end stations, LANs, bridges, routers and similar systems, and the servers and services that support these. The description and specification in this clause are limited to use of the secure MAC Service to contribute to overall system security (see Clause 1).

NOTE 2—In order to introduce the concepts used in this standard, this clause can repeat or summarize the specification in other clauses; however, it contains no normative provisions that apply either to the subject matter of those other clauses or to the other standards referenced. For conformance to this standard (see Clause 5).

NOTE 3—The term *individual LAN* is used in this standard to refer explicitly to an instance of media access method-specific technologies providing the MAC Service directly. The term excludes larger networks or subsets of a network that are created by aggregation or concatenation of individual LANs by Link Aggregation or bridges.

NOTE 4—The examples presented in this clause are intended to serve as a guide to best practice; however, the use of MAC Security is not limited to the examples given. Limits to the use of MAC Security that are required for the successful operation of network configuration and other protocols are made explicit.

7.1 Support of the secure MAC Service by an individual LAN

Each port that is capable of participating in an instance of the secure MAC Service comprises both a KaY and a SecY. Each KaY discovers or is made aware of the KaYs present in other stations attached to the same LAN, ensures that those stations are mutually authenticated and authorized, and creates and maintains the secure relationships between the stations that are used by the SecYs to transmit and receive frames. Specifically

a) A secure Connectivity Association (CA) is created to meet the requirements of the MAC Service (6.2) and MACsec (6.7) for connectivity between the stations attached to an individual LAN.

b) Each CA is supported by unidirectional Secure Channels (SCs), each SC supporting secure transmission of frames through the use of symmetric key cryptography, from one of the systems to all the others in the CA.

c) Each SC is supported by an overlapped sequence of Security Associations (SAs).

d) Each SA uses a fresh Secure Association Key (SAK) to provide the MACsec service guarantees (6.8) and security services (6.9) for a sequence of transmitted frames.

NOTE—An SC can be required to last for many years without interruption, since interrupting the MAC Service can cause client protocols to re-initialize and recalculate aggregations, spanning trees, and routes (for example). An SC lasts through a succession of SAs, each using a new SAK, to defend against a successful attack on a key while it is still in use. In contrast it is desirable to use a new SAK at periodic intervals to defend against a successful attack on a key while it is still in use. In addition, the MACsec protocol (Clause 8 and Clause 9) only allows 2^{32}–1 frames to be protected with a single key unless a Cipher Suite that supports extended packet numbering is used. Since 2^{32} minimum-sized IEEE 802.3 frames can be sent in approximately 5 min at 10 Gb/s, this can force the use of a new SA.

These security relationships (CAs, SCs, and SAs) and the information associated with each of them are further discussed in 7.1.1, 7.1.2, and 7.1.3. Their mutual relationship, and the insecure connectivity provided by the LAN that supports them, are illustrated in Figure 7-1 through Figure 7-3 for a point-to-point LAN and in Figure 7-4 through Figure 7-6 for stations attached to a shared media LAN.

Figure 7-1 shows two stations, A and B, connected to a point-to-point LAN that provides insecure bi-directional connectivity.

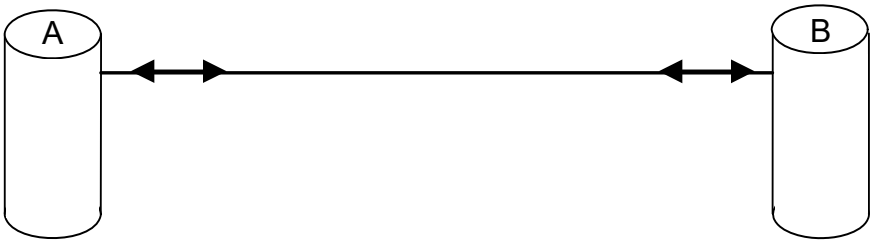


Figure 7-1—Two stations connected by a point-to-point LAN

Figure 7-2 depicts the CA created by MACsec Key Agreement following mutual authentication and authorization of A and B.

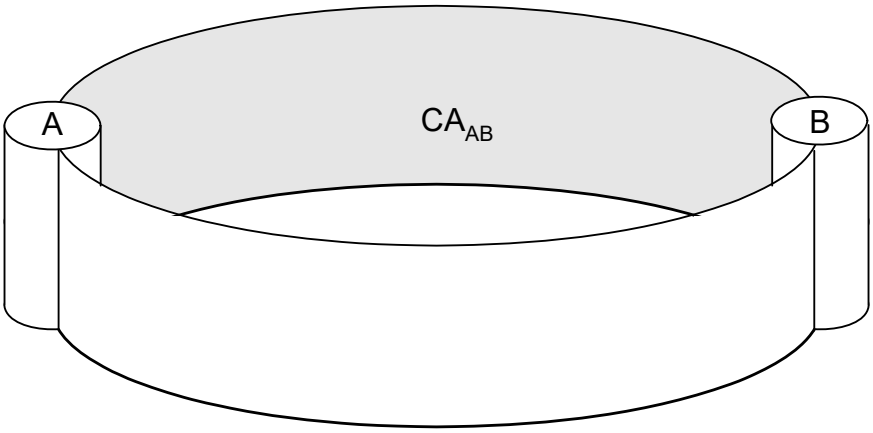


Figure 7-2—Two stations in a CA created by MACsec Key Agreement

Figure 7-3 shows the two SCs that support the CA.

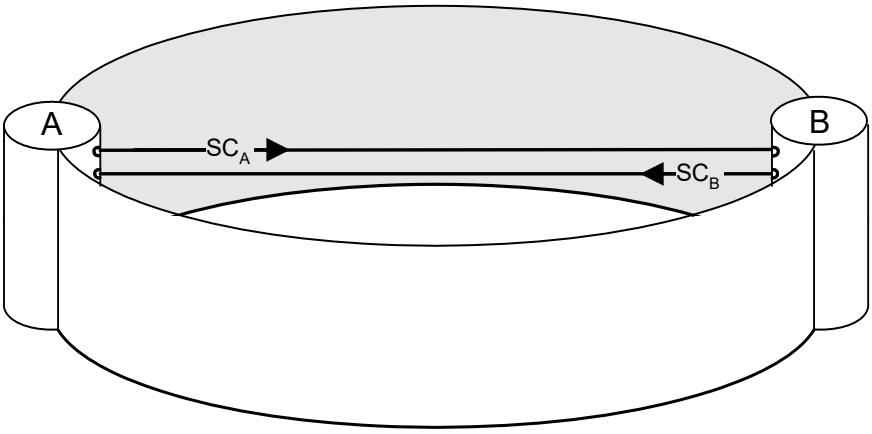


Figure 7-3—Secure communication between two stations

Figure 7-4 shows four stations, A, B, C, and D, attached to a shared media LAN that provides full but insecure connectivity between the stations.

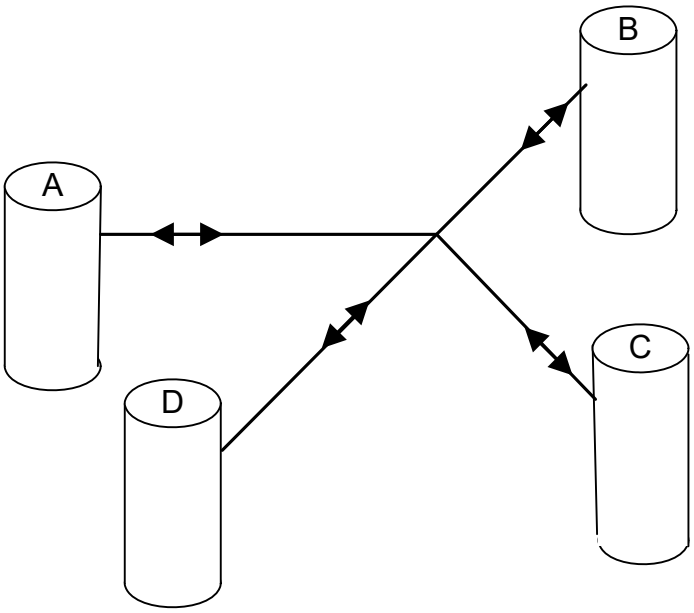


Figure 7-4—Four stations attached to a shared media LAN

Figure 7-5 depicts a CA created by MACsec Key Agreement following mutual authentication and authorization of A, B, and C. The CA excludes D.

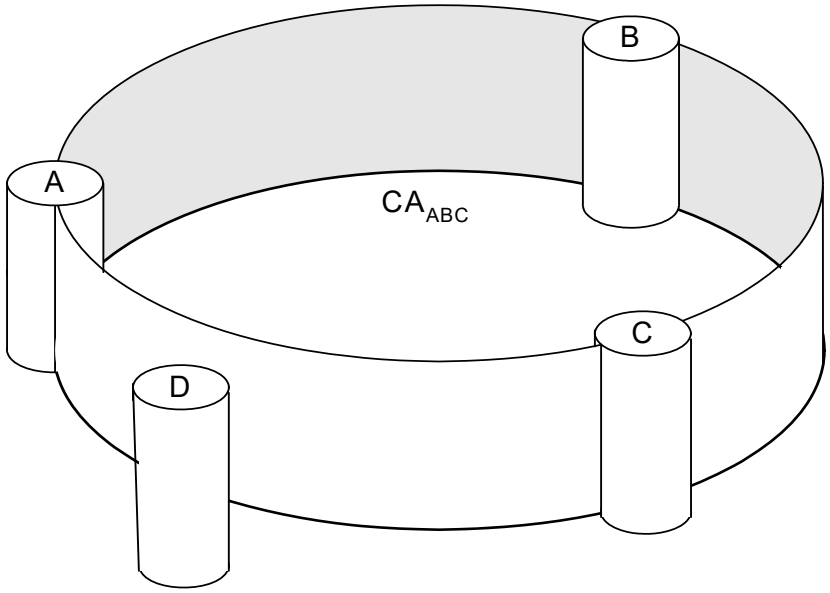


Figure 7-5—A CA including ports A, B, and C

Figure 7-6 shows the three SCs that support the CA, one for transmission by each of A, B, and C.

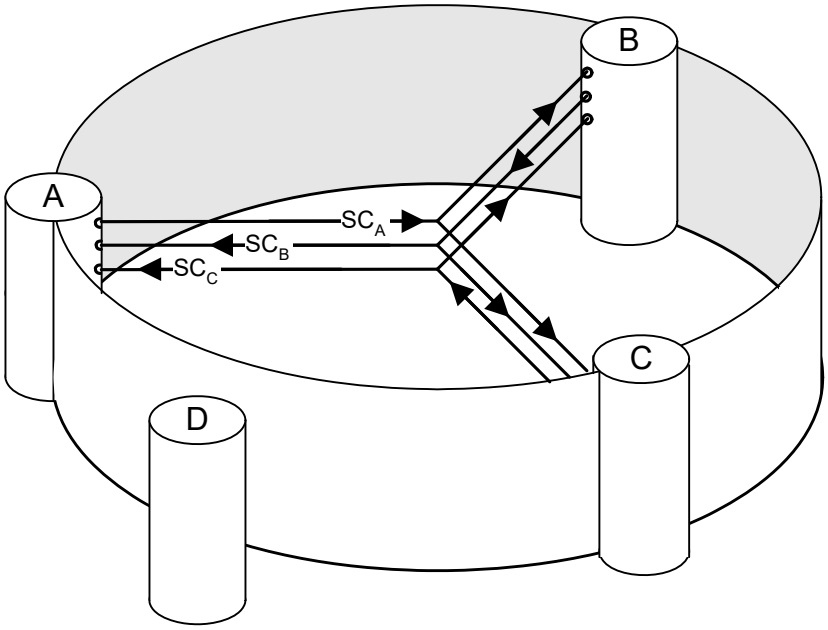


Figure 7-6—Secure communication between three stations

While D can send and receive frames using the insecure connectivity provided by the shared LAN, it does not have SAKs that would allow it to participate in any of the SAs that currently support SC_A, SC_B, or SC_C; therefore, D cannot compromise the integrity, confidentiality, or origin of any of the frames being exchanged by A, B, and C.

In the above examples (Figure 7-3, Figure 7-6), each station transmits frames using a single SC. A station can also use multiple transmit SCs, using each transmitted frame's priority to allocate to it one of the SCs. Each of these transmit SCs supports secure transmission for frames of one or more priorities from the transmitting station to all the others in the CA. This use of multiple transmit SCs allows MACsec to enforce in-order delivery (or the use of a smaller replay window than might otherwise be the case) for frames of any given priority.

7.1.1 Connectivity Association (CA)

MACsec Key Agreement is responsible for discovering, authenticating, and authorizing the potential participants in a CA. A SecY, as specified in this standard, does not need to be aware of the CA, except as a list of SCs in which it needs to participate. Since all the SCs in a CA use the same Cipher Suite at any one time, the Cipher Suite can be considered a property of the CA. A change in the Cipher Suite necessitates an interruption to the service provided by the CA.

Each SecY participates in only a single CA at any one time. There is a limit, readable by management and by the KaY, on the number of peer SecYs that can participate in a CA (10.7.7, 13.5).

NOTE—If this specification had allowed different SCs to use different Cipher Suites, a SecY implementing more than one Cipher Suite would have to be capable of simultaneous transmitting using one Cipher Suite and receiving using one or more other Cipher Suites.

7.1.2 Secure Channel (SC)

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.

NOTE 1—Including the Port Identifier component would appear to be unnecessary in the case of a simple system with a MAC address and a single SecY for each port. However, some systems require support for more SecYs than they have uniquely allocated addresses because they make use of technologies that support virtual MACs, or because their interface stacks include the possibility of including multiple SecYs at different sublayers (as do Provider Bridges [IEEE Std 802.1Q], for example), or because they transmit frames of different priorities using different SCs.

NOTE 2—An EPON Optical Line Terminal (OLT) can use a distinct SC to support the Single Copy Broadcast (SCB) capability (Clause 12). The formal identifier for this SC comprises a System Identifier for the OLT and a reserved Port Identifier. Both can be represented in the secured frame by a single SCB bit (Clause 9).

MACsec Key Agreement is responsible for informing each SecY of the identifier to be used for each transmitting SC and of the existence and identifier of each of the SCs for which the SecY is to receive frames. While the structure of the communication facilitated by each SC is point-to-multipoint (which encompasses point-to-point as a special case), the SecY does not have to be aware that its transmissions can reach multiple receivers, that the frames that it receives could be received by other SecYs, or of any relationship or lack of relationship between the inbound SCs (except in determining the value of the operPointToPointMAC status parameter, 6.5, 10.7.4). The operation of the MACsec Key Agreement protocol (MKA, specified in IEEE Std 802.1X) is defined in terms of the behavior of participants, each representing a single KaY and a single transmit SCI. When a SecY uses multiple transmit SCIs, each SCI is represented by a separate participant that sends and receives MACsec Key Agreement PDUs (MKPDUs) to and from all the other participants just as if it were representing a separate SecY in the same group CA, see Annex E.

NOTE 3—The point-to-multipoint nature of the SC does have technical consequences, in particular the decision to change from one SA to another is made by the transmitter using the SC, not by one or some number of the receivers.

7.1.3 Secure Association (SA)

Each SC comprises a succession of SAs, each with a different SAK. Each SA is identified by the SC identifier concatenated with a two-bit Association Number (AN, Figure 7-7). The Secure Association Identifier (SAI) thus created allows the receiving SecY to identify the SA and thus the SAK used to decrypt and authenticate the received frame. The SAI is only unique for the SAs used by SecYs participating in a given CA at any instant.

MACsec Key Agreement is responsible for creating and distributing SAKs to each of the SecYs in a CA. This key creation and distribution is independent of the cryptographic operation of each of the SecYs. The same SAK can be used for SAs that compose different SCs, provided that every initialization vector (IV) used with the SAK is unique. When the Default Cipher Suite (14.5) is used, the SCI is included in the IV to ensure uniqueness across SCs.

The decision to replace one SA with its successor is made by the SecY that transmits using the SC, after MACsec Key Agreement has informed it that all the other SecYs are prepared to receive using that SA. No notification, other than receipt of a secured frame with a different SAI is sent to the receiver. At any one instant a SecY has to be capable of storing SAKs for two SAs for each inbound SC, and of swapping from one SA to another without notice. Certain LAN technologies can reorder frames of different priority, so reception of frames on a single SC can use interleaved SAs. The time bound within which a receiver can accept interleaved SAs is 0.5 s.

The transmitting SecY does not interleave frames for different SAs on a given SC.

NOTE—When MKA (see IEEE Std 802.1X) is used to distribute the SAKs used by each of the SCs supporting a given CA, the same SAK is used for all SAs with a given AN (at any given time). The transmit SA for each SC is replaced with its successor at approximately the same time so that there is no need for any SecY participating in the CA to support more than two SAKs at a time.

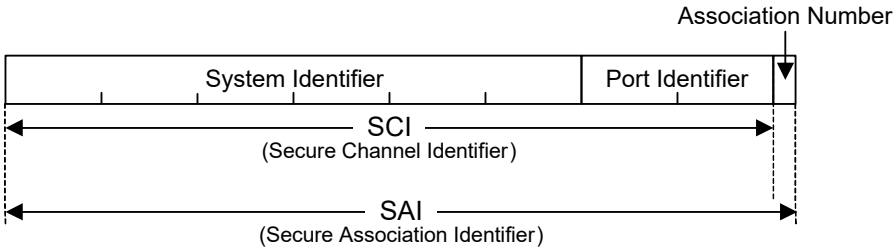


Figure 7-7—Secure Channel and Secure Association Identifiers

If a SecY does not have a usable SA for its outbound SC, i.e., an SA that can be used at no notice for frame transmission with a PN value that is not exhausted, or any of the current SAs for inbound SCs are not usable, then the MAC_Operational status parameter (6.4) will transition to FALSE.

7.2 Multiple instances of the secure MAC Service on a single LAN

Each service access point for an instance of the secure MAC Service is supported by a service access point for an instance of an insecure MAC Internal Sublayer Service (ISS). Multiple instances of the secure MAC Service can be provided by a single LAN, provided that each instance is uniquely identified by unencrypted fields contained in each received frame. These fields identify separate instances of the insecure MAC ISS, each capable of supporting a distinct service access point for MACsec.

Identification of each insecure service instance, and multiplexing and demultiplexing to and from the transmission capabilities provided by the LAN, can be performed wholly below the ISS by a media-specific or media-dependent function. Some media are defined to support such a multiplexing function, e.g., the

LLID used by IEEE 802.3 EPON (see Clause 12). Where such media-specific multiplexing functions are not available, the source address or SCI may be used (11.8). Provider Bridges are also capable of supporting multiple instances of the ISS over a network of individual LANs (see 11.6).

MACsec should not be used to support multiple instances of the secure MAC Service on a single physical LAN without the use of unencrypted frame fields to identify separate instances of insecure service, each supporting a single instance of secure service. While the use of security to provide multiplexing is impossible to prevent (since different cryptographic keys can be used to separate connectivity), relying solely on security to define the connectivity makes deployment and fault management difficult—the topology of an entire network could change as security was enabled or disabled on a single LAN. Key agreement protocols that use the insecure MAC service can require a matching instance of that service for each secure service instance.

NOTE 1—The service access point for the secure MAC Service is referred to as Controlled Port of the SecY (Clause 10) and the service access point for the insecure MAC Service as the SecY's Common Port. Access to the insecure service for protocol entities above MAC Security is provided at the Uncontrolled Port.

NOTE 2—Although the field or fields used to provide service instance multiplexing are not parameters of the ISS, and thus are not protected, the integrity of the secure MAC Service is not compromised. If the unprotected fields are modified, the frame can be delivered to the wrong SecY, but will subsequently fail integrity checks. Different SecYs use different security associations, keys, and cryptographic nonces. Additional management parameters are (cryptographically) bound to individual SecYs, not to the values of frame fields.

The secure MAC Service requirements for symmetric and transitive connectivity ensure that two or more service instances on the same LAN will appear as separate LANs to the clients of the SecYs. There is therefore no conflict between the use of bridges and the provision of multiple secure service instances.

When clients that are connected to a first service instance change and connect to a second service instance, the secure connectivity alters. MAC_Operational temporarily transitions to False for a minimum amount of time to allow the CA to re-establish its membership (as determined by the KaY). In particular, each time membership of a CA changes, MAC_Operational transitions False for at least one of each pair of SecYs whose connectivity has changed. For example, if members of CA_x leave CA_x and join CA_y and if CA_y has MAC_Operational True, then MAC_Operational transitions to False for either the members of CA_x who are joining CA_y or for the original members of CA_y. MAC_Operational transitions to True once all the new members have joined CA_y.

NOTE 3—Two SecYs that connect to the same LAN and participate in the same CA appear connected to the same LAN (as one would expect) and appear connected to different LANs as they participate in distinct CAs. The effect is similar to partitioning a LAN by switching a repeater on or off.

Distinct instances of the secure point-to-point MAC Service can be provided by a bridge to different end stations connected to the same shared media by using the source address of the frames transmitted by each end station to identify one of a number of SecYs in the receiving bridge (11.8). Where the source address is not sufficient to select the receiving SecY, the SCI can be used to provide service instance multiplexing for both secured and unsecured frames (11.8, 9.5, 10.6).

NOTE 4—IEEE Std 802.11 [B2] specifies its own mechanisms for identifying separate secure associations.

7.3 Use of the secure MAC Service

The secure MAC Service guarantees (6.8) the integrity of the parameters of each service indication, and that each indication is a result of a request made by a SecY that is a member of the same CA as the receiver, though not by any particular member. Management controls associated with each KaY can require certain authentication and key management methods to ensure these guarantees. However, the degree of trust placed in the security of the communication does not imply the degree of trust associated with the communicating peers. Accordingly, the MACsec Key Agreement framework facilitates authorization of each potential member and allows management of the acceptable authorization for inclusion in the CA.

The secure MAC service does not itself provide any means to label or distinguish different levels of authorization, and does not associate different levels of authorization with individual invocations of the service. A station either participates in a service instance or it does not.

To ensure correct operation of client protocols, secure service indications are not filtered or modified by a SecY except as specified in Clause 8 and Clause 9. Each protocol entity that is a client of the secure MAC Service should implement suitable policies (7.3.1) to support overall network security.

NOTE 1—For example, correct operation of Spanning Tree Protocol depends on the delivery of BPDUs to the Spanning Tree Protocol Entity of a given bridge from all the other bridges attached to the LAN that transmit frames that can be relayed by the given bridge. If a SecY were to require a higher level of authorization to pass received BPDUs through the Controlled Port, data loops in the network could result. However, the STP Entity can adopt a policy of discarding frames rather than permit another system that is not authorized as a bridge to be the Designated Bridge for the CA.

The client policies in use at any time should reflect the intersection of the capabilities permitted to the members of the CA. Policies can be

a) Selected by the client on the basis of the level of authorization, as provided by the PAE through a Layer Management Interface (LMI) (see 10.7) or
b) Selected by a central server that forms part of the management framework for the network, and
    1) Securely downloaded or
    2) Communicated to the client using a secure connection

NOTE 2—If one of the members of the CA is a bridge (strictly speaking the Bridge Port is the CA member), the other members should adopt policies that reflect their confidence in the policies applied by the bridge to forward frames. In this case the trust is partly transitive—the question to be answered by each member of the CA is the degree of trust to place in the bridge's trust of systems that originate frames that the bridge will forward.

7.3.1 Client policies

Client policies, which are not specified in this standard, can include but are not limited to

a) Limiting the set of protocol procedures that can be invoked by the peer
b) Segregating communications between different sets of peer users of the MAC Service
c) Filtering, i.e., discarding, received frames

Clients of the secure MAC Service can also record any exceptional policy actions taken, so as to initiate further administrative action, outside the scope of this standard, with the entities responsible for the operation of the authenticated peer systems.

NOTE—A VLAN-aware Bridge that assigns frames that have been received from a specific Bridge Port (the bridge's point of attachment to a service instance) to a VLAN on the basis of the authorization associated with the Port provides an example of policy of segregating communications, as described in item b) above.

7.3.2 Use of the secure MAC Service by bridges

Each Bridge Port uses the service provided by an individual LAN (see Clause 11), which is not dependent for its connectivity on the operation of other bridges. This ensures that the configuration protocols used by bridges, including the spanning tree protocol, operate over a physical topology (comprising a bipartite graph of bridges and individual LANs connected by Bridge Ports) that is not itself dependent on the active topology, or subsets of the active topology, calculated by those same configuration protocols.

NOTE 1—The apparent exception to this configuration restriction, which does not permit the creation of security associations to create "secure tunnels" through selected bridges in a Bridged Local Area Network, is the use of a Provider Bridged Network as specified in IEEE Std 802.1Q. However, a Provider Bridged Network appears to Customer Bridges as a single LAN providing full connectivity independent of the operation of Customer Bridge protocols.

PAEs and KaYs use group addressed frames to identify and communicate with peers whose SecYs are potential participants in the same CA. Frames with the group addresses used for this purpose are filtered by certain bridges and EDEs (Clause 15) to restrict each instance of the secure MAC Service to the appropriate LAN scope. By default PAEs and KaYs for MAC Bridges, VLAN Bridges, Provider Bridges, and Provider Backbone Bridges use the PAE group address specified by IEEE Std 802.1X (also identified as the Nearest Non-TMPR Bridge group address by IEEE Std 802.1Q). Use of this address restricts each instance of the secure MAC Service to an individual customer LAN.

NOTE 2—Use of Reserved Group MAC Addresses helps to ensure that the physical topology as perceived by spanning tree and other configuration protocols aligns with that provided by MAC Security.

The policies applied by the Bridge Forwarding Process that is a client of each MAC service instance can include but are not limited to

a) Use of static Filtering Database Entries
b) Use of the RSTP and MSTP restrictedRole parameters
c) The PVID for the port
d) Configuration of the VLAN Translation Table
e) Inclusion in the Member Set for any given VLAN and the setting of Enable Ingress Filtering
f) Identification of the Port as a Provider Edge Port
g) Port priority
h) Priority remapping tables.

NOTE 3—A Bridge Port is one of the bridge's points of attachment to an instance of the ISS and is used by the MAC Relay Entity and associated Higher-Layer Entities as specified in IEEE Std 802.1Q.

NOTE 4—The RSTP and MSTP restrictedRole parameters in IEEE Std 802.1Q ensure that the spanning tree active topology for other Bridge Ports is unaffected by BPDUs received on the Port, while continuing to protect against data loops and allowing the peer system to use the BPDUs it receives to select between redundant service instances. The restrictedRole parameter should be set if the authorization (see also 7.3) of the peer system(s) is not sufficient to allow full participation in determining the active topology of the network.

In response to a limited authorization on the Bridge Port, a bridge can be configured to discard frames other than from a specified number of MAC addresses and to use additional services provided by the network administrator to ensure that these permitted addresses are not used by other end stations in the network.

8. MAC Security protocol (MACsec)

MACsec provides the secure MAC Service (Clause 6) on a frame-by-frame basis, using cryptographic methods within the context of security relationships maintained by MACsec Key Agreement.

This clause

a) Sets out requirements for the design (8.1) and support (8.2) of MACsec
b) Provides an overview of its operation (8.3).

NOTE 1—The operation of MACsec Key Agreement Entity (KaY) and the protocols it uses are outside the scope of this standard. However, the security relationships (Clause 7) it establishes are essential to the operation of MACsec and form part of the support requirements.

Conformance to this standard is in terms of the observable protocol arising from the operation of a MAC Security Entity (SecY, Clause 10), including management of MACsec and the service provided to client protocols that use the secure MAC service.

Each of the possible sets of cryptographic algorithms used by MACsec to provide connectionless frame integrity and data confidentiality compose a Cipher Suite. This clause describes the result of Cipher Suite use by the SecY, illustrated in Figure 8-1. The normative specification of each Cipher Suite is provided in Clause 14. The Cipher Suite is selected as part of the establishment of the CA (7.1.1).

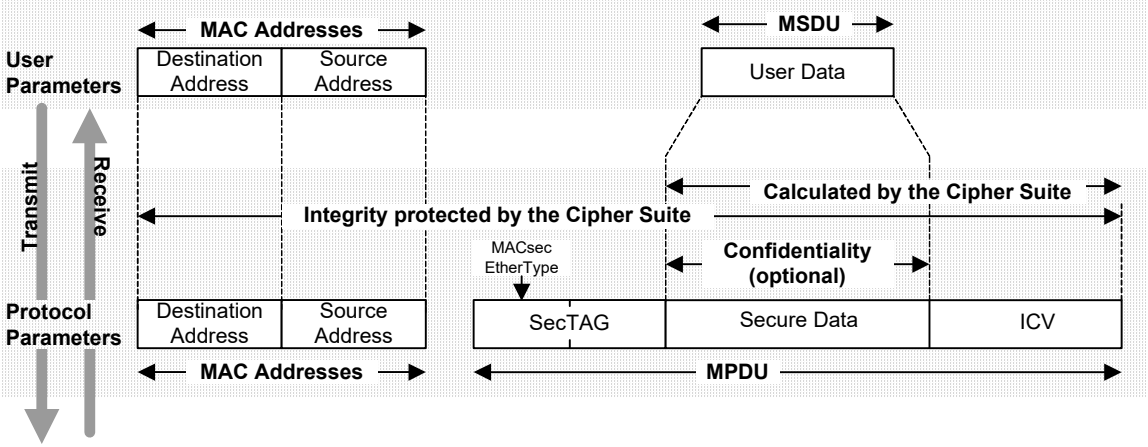


Figure 8-1—MACsec

NOTE 2—The Destination Address and Source Address parameters are shown as separate from the MPDU in Figure 8-1, as they are separate parameters of each service request. The encoding of these parameters into a transmitted frame on a medium is accomplished by the supporting service, which can interpose additional octets between those of the addresses and the MSDU. In the strict sense of externally visible transmission, this standard deals with parameters of service primitives, not with frames. However, it is often convenient to talk of these parameters as a frame.

8.1 Protocol design requirements

MACsec operates in networks comprising end stations and individual point-to-point or shared media LANs, arbitrarily interconnected by intermediate systems, such as MAC Bridges, VLAN-aware Bridges, and routers. MACsec supports, preserves, and maintains the quality of the secure MAC Service in all its aspects as specified by Clause 6, meeting requirements for

a) Connectivity (6.7)
b) Security (6.8, 6.9, 8.1.1)

c) Manageability (8.1.2)
d) Interoperability (8.1.3)
e) Deployment (8.1.4)
f) Coexistence (8.1.5)
g) Scalability (8.1.6)
h) Intrusion detection (8.1.7)
i) Localization and isolation of attacks (8.1.8)
j) Implementation (8.1.9).

These requirements are met by the operation of MACsec (8.3) together with requirements placed on

k) The architecture that specifies how MAC Security Entities (SecYs) are placed within LAN stations and communicate with selected peers (Clause 11)
l) The choice of cryptographic methods that compose each MACsec Cipher Suite (Clause 14)
m) Support of the protocol by each SecY, and on the system that contains it (8.2)
n) The operation of the protocols that support MACsec Key Agreement, including aspects of authentication, authorization, and distribution of keys.

8.1.1 Security requirements

In addition to providing the security guarantees (6.8) and services (6.9) required for support of the secure MAC service, the design of MACsec

a) Enables a succession of SAs, each with its own Secure Association Key (SAK) to be used to support the connectivity provided by each SC. The succession of SAKs (7.1), together with the use of Key Agreement protocols that provide Perfect Forward Secrecy, protects against the compromise of any single SAK, without disrupting service.
b) Ensures that a fresh SA, supporting an existing CA, can be used within a known bounded time (1 s, see 8.1.9) at intervals that are also bounded (keys can be changed as frequently as once every 10 s) after Key Agreement provides the associated SAK.
c) Allows operation of the Key Agreement protocol to be independent of MACsec. In particular allows fresh SAKs to be supplied at any time, without unnecessarily disrupting communication.

The security provided by each SAK rests on the security provided by the Cipher Suite (see Clause 14 for requirements), which in turn depends on the guarantees provide by the cryptographic mode of operation and its underlying block cipher, and on the protocols and procedures used to ensure that keys remain secret.

8.1.2 Manageability requirements

The design of MACsec ensures that the protocols that configure, and that run over media, individual LANs, and Bridged or Virtual Bridged Local Area Networks as a whole, can continue to operate with no diminution in the capabilities available to and customarily used by network administrators. Existing firewall and forwarding filters can still be applied to specific protocols.

When the Default Cipher Suite is used for integrity protection without confidentiality protection, protocol analyzers and other tools that support MACsec parsing can understand the User Data transmitted, but cannot modify that data without the receiving SecY being aware of the intrusion. This capability is also available whenever the Secure Data remains the same as the User Data and the integrity check value (ICV) length is the same as that of the Default Cipher Suite.

Where MACsec supports a shared media CA, or a point-to-point CA that uses shared transmission facilities, MACsec can convey the SCI (7.1.2, 8.2.1, 9.9), thus identifying the secure system that transmitted the MPDU both to the intended recipient and to network management systems.

8.1.3 Interoperability requirements

Interoperability between independent implementations of MACsec is facilitated by mandatory implementation of a Default Cipher Suite.

The use of Cipher Suites as a specification tool reduces the number of permutations of cryptographic algorithms and their parameters. Clause 14 mandates elements of Cipher Suite specification.

Where the underlying MAC Service used by MACsec is supported by a Provider Bridged Network (IEEE Std 802.1Q), communicating SecYs can be attached to different media operating (locally) at different transmission rates. Interoperability between, for example, 10 Gb/s and 1 Gb/s, and between 1 Gb/s and 100 Mb/s requires interoperability across the speed range. The design of MACsec facilitates interoperability from 1 Mb/s to 100 Gb/s without modification or negotiation of protocol formats and parameters. Operation at higher transmission rates depends on the capabilities of the Cipher Suite. The mandatory default Cipher Suite has been selected (Clause 14) in part because of its ability to perform across this range.

NOTE—Clearly additional ways of interconnecting different media access control methods could be standardized in the future. The above requirement mandates that interoperability be preserved between SecYs attached to a wide range of media operating over a wide speed range.

Communication between SecYs using different media access methods requires that MACsec not make use of any media-specific additions to the MAC Service, or rely on any deficiencies in support of the service being common to all communicating participants. MACsec includes an explicit indication of the length of the Secure Data to avoid imposing the minimum frame size and padding requirements of IEEE Std 802.3 on all other media access methods that make use of MACsec.

8.1.4 Deployment requirements

The design of MACsec allows security to be introduced into a network one LAN at a time. Additionally the controls provided by a SecY (Clause 10) allow the deployment of MACsec capable systems one by one on a LAN, prior to enabling security. Integrity checking of MPDUs using the Default Cipher Suite can be disabled to facilitate testing of Key Agreement protocols prior to enabling security. Management counters allow a network system administrator to confirm that the connectivity provided by a SecY is complete and that enabling security will not disrupt existing required connectivity.

8.1.5 Coexistence requirements

The design of MACsec allows coexistence with other protocols on the same insecure LAN. This

a) Supports incremental deployment (8.1.4).
b) Allows fresh keys to be derived, using Key Agreement protocols that can be independently specified and use different frame formats, while MACsec is operating.
c) Supports use of shared media providing independent services.

8.1.6 Scalability requirements

The resources required to support MACsec in any single LAN station (an end station or a Bridge Port) are a function of the number of the SecY peers on the same LAN, but are independent of other systems attached to the same network but not the same LAN.

8.1.7 Unauthorized access attempts

Detecting attempts at unauthorized access is facilitated by integrity and replay protection, and the management counters (10.7) that record the receipt of invalid (presumably modified) and repeated and misordered (likely to be replayed) frames. Management for client policies (7.3) that use the guaranteed connectivity provided by MACsec should also record attempted violations.

8.1.8 Localization and isolation of attacks

MACsec discards frames sent by systems that are not authenticated and authorized members of the CA, thus localizing the traffic sent by those stations to a single LAN. The authorization accorded by the policies enforced by clients of MACsec (7.3.1) can restrict unauthorized attempts to affect protocols that control the network infrastructure. Where communication that does result in unauthorized behavior elsewhere in the network has been permitted, the use of MACsec by the intervening systems allows tracing of the source of that communication.

8.1.9 Implementation

The design of MACsec allows the SecY to function asynchronously with respect to other processes in the system. Key Agreement protocols and changes of SAKs are not tightly synchronized to the service requests and indications processed by the SecY. Delays in communication and variations in scheduling between the SecY and KaY can be as much as one second, allowing autonomous processing of frames in real time by the SecY while the KaY can operate as a normally scheduled software process. Time is also allowed for the KaY to compute keys and for the SecY to compute key schedules, and perform other preprocessing.

8.2 Protocol support requirements

The support of MACsec places requirements on

a) The secure system of which the SecY forms a part for
   1) SC identification (8.2.1)
   2) Support of transmit and receive SAKs (8.2.2).
b) The functionality provided by Key Agreement protocols, and the operation of the KaY for
   1) Independence of KaY operation from MACsec operation and state (8.2.3)
   2) Discovering connectivity (8.2.4)
   3) Authentication (8.2.5)
   4) Authorization (8.2.6)
   5) Key exchange and maintenance (8.2.7).

8.2.1 SC identification requirements

Each SecY shall be capable of identifying each of its transmit SCs with an SCI that comprises a unique 48-bit MAC Address and a 16-bit Port Identifier that is unique within the scope of that address (7.1.2, 9.9).

NOTE—MKA (IEEE Std 802.1X) verifies that each participant in any given CA has a unique SCI, as part of satisfying Cipher Suite requirements prior to establishing secure communication.

8.2.2 SA Key requirements

On transmit the Cipher Suite implementation shall be able to

a) Prepare a new SAK for use within one second (8.1.9) of being given it by the KaY.
b) Change from the use of one installed SAK to the next within the time normally taken to transmit one minimum sized frame, and shall not discard any frames as a result of the change.

NOTE—Elsewhere in this standard, the requirement for switching between SAKs is modelled as a requirement to support two SAKs for transmission, allowing management counters to reflect the continued use of a key after its successor has been provided by the KaY. The behavior of an implementation capable of accepting the new key and using it within one frame time is fully conforming, and will not cause any apparent management anomalies.

On receive the Cipher Suite implementation shall be able to

c) Receive any frame and its immediate successor using any one of two SAKs, allowing the selection of different keys to switch without missing a frame.
d) Prepare a new SAK for use within one second (8.1.9) of being given it by the KaY.

The system does not need to be able to seamlessly switch between Cipher Suites.

8.2.3 KaY independence of MACsec

The KaY is aware of the required connectivity, identifying the SCs that compose the CA, independently of the design and state of MACsec.

The KaY operates resiliently in face of specifically identified denial of service attacks (as identified by the key agreement protocol specification).

These requirements are met in part by distinguishing key agreement frames from MACsec frames by using a different EtherType.

8.2.4 Discovering connectivity

The KaY discovers connections between peer stations and recognizes potential connections.

NOTE 1— The MAC status parameters (6.4) indicate when connectivity changes. The status parameters provided by the KaY can also temporarily transition False to indicate a change in the authentication or authorization of its peers, preventing attacks that secretly degrade the trust.

The KaY accepts indications of which Cipher Suites are supported by the SecY via the LMI.

NOTE 2—The negotiation of which Cipher Suite is to be used on a connection is based on what Cipher Suites are available locally and at the peer SecY.

The KaY accepts indications of which connectivity capabilities are supported by the SecY via the LMI. The KaY delivers the connectivity selection to the SecY via the LMI.

8.2.5 Authentication requirements

The PAE supports mutual authentication of peer stations, and the SecY assumes that such authentication has taken place.

8.2.6 Authorization requirements

The PAE provides authorization of services to be delivered to a peer station.

The PAE provides information to local services about the currently selected Cipher Suite.

8.2.7 Key exchange and maintenance

The KaY delivers SAKs via the LMI (10.7.28).

The KaY creates, manages, and maintains one CA that connects two or more KaYs and their corresponding SecYs. The KaY creates and maintains all of the point-to-multipoint SCs and SAs between itself and all the stations within the CA (10.2, 10.7.11–10.7.15, 10.7.21–10.7.24). An SAK delivered by a given KaY is not shared with any other KaY, is not used by the given KaY to support more than one CA, and once used to support an SA for a given SC is not re-used to support any other SA for that SC. A KaY can (and in the MACsec Key Agreement protocol (MKA) specified in IEEE Std 802.1X does) use a single SAK to support multiple SCs within a CA. It is recognized that two SAKs can have the same value with a probability of no less than 1 in $2^{keysize}$ when generated by an approved pseudorandom function.

The KaY monitors the use of PNs by the SecY via the LMI in order to identify impending exhaustion of the transmitting SA (10.7.23). IEEE Std 802.1X-2010 specifies the distribution of a fresh SAK when the value of the PN exceeds that of the constant PendingPNExhaustion (0xC000 0000 for 32-bit PNs). If extended packet numbering (a 64-bit PN) is used in conjunction with IEEE Std 802.1X, PendingPNExhaustion takes the value 0xC000 0000 0000 0000.

The KaY accepts indications that one SA is retired and a new one is started, in other words, when an overlapping pair of SAs is provisioned and the SecY switches from one to the next.

8.3 MACsec operation

MACsec comprises modification and additions to the MAC Service Data Unit (MSDU) conveyed by each frame transmitted by a user of the protocol, and illustrated in Figure 8-1. The MAC Security TAG (SecTAG) conveys parameters that identify the protocol, identify the key to be used to validate the received frame, and provide replay protection. The Secure Data field conveys the User Data, encrypted if confidentiality is provided. The ICV ensures the integrity of the MAC Destination Address, MAC Source Address, SecTAG, and User Data.

NOTE 1—The addition of the SecTAG and ICV to the MPDU, together with possible expansion of the User Data when conveyed in the Secure Data field can increase the size of a frame to the point that it no longer conforms to the maximum frame size specified by the media access method standard. If the implementation of the service used by MACsec cannot transmit the resulting MPDU, it is discarded.

MACsec does not transmit additional frames, such as keep alives or key exchanges. Each frame is delivered unmodified to peer users, subject to validation of the origin, destination and source address, and user data.

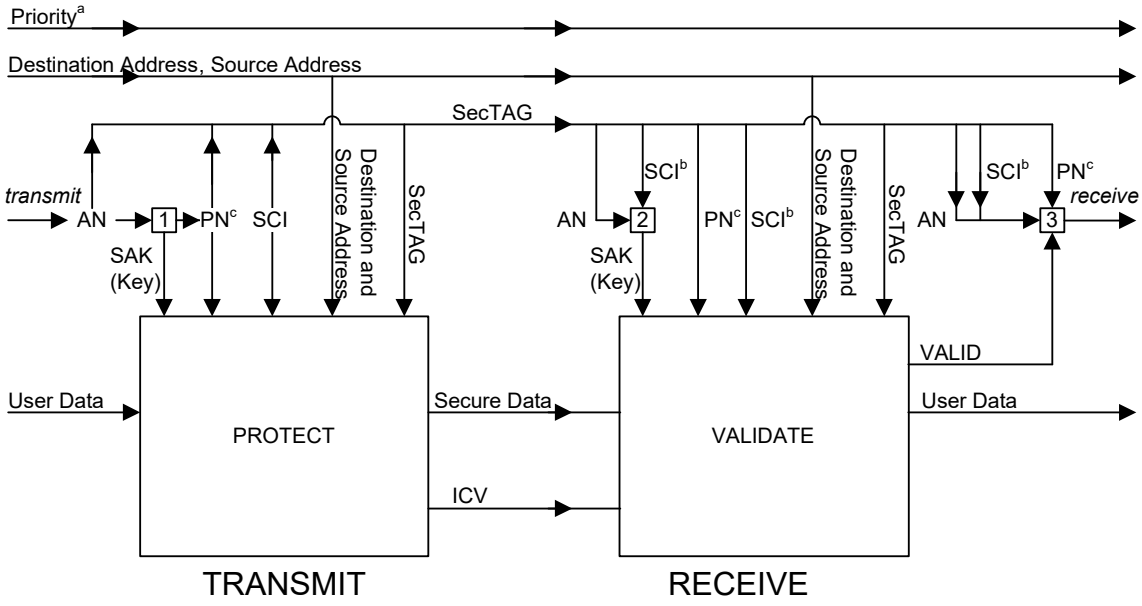
Figure 8-2 illustrates the transmission and reception of a frame by MACsec.

On transmission, the frame is first assigned to a transmit SC and to the SA (7.1.3) identified by its Association Number (AN) (7.1.3, 9.6) that will be used by that SC to protect the transmitted frame. The AN is used to identify the SAK (7.1.3) and the next PN (9.8) for that SA. The AN, the SCI (7.1.2), and the 32 least significant bits of the PN are encoded in the SecTAG (the SCI can be omitted for point-to-point CAs if only one transmit SC is in use) along with the MACsec EtherType (9.8) and the number of octets in the frame following the SecTAG if less than 48 [8.1.3; Short Length (SL) field in 9.7].

The protection function (14.1) of the Current Cipher Suite is presented with the SAK, the PN and SCI, the destination and source addresses of the frame together with the octets of the SecTAG, and the User Data. It returns the Secure Data and the ICV.

On receipt of a MACsec frame, the AN, SCI, PN, and SL field (if present) are extracted from the SecTAG. If the CA is point-to-point and the SCI is not present, the value previously communicated by the KaY will be used. The AN and SCI are used to assign the frame to an SA, and hence to identify the SAK. If the Current Cipher Suite uses extended packet numbering (a 64-bit PN), the full PN is recovered (as specified in 10.6) using the 32 least significant bits conveyed in the SecTAG and the 32 most significant bits used in a prior successful frame validation.

The validation function of the Current Cipher Suite is presented with the SAK, the PN and SCI, the destination and source addresses of the frame together with the octets of the SecTAG, and the Secure Data and ICV. If the integrity of the frame has been preserved and the User Data can be successfully decoded from the Secure Data, a VALID indication and the octets of the User Data are returned.



[a] Priority can be changed by media access method or receiving system and is not protected
[b] The SCI is extracted from the SCI field of the SecTAG if present. A value conveyed by key agreement (point-to-point only) is used otherwise.
[c] The SecTAG carries only the least significant 32 bits of the PN. When a 64 bit PN (extended packet numbering) is used, the most significant 32 bits are recovered on receipt, and the complete 64 bit PN is presented to PROTECT, VALIDATE, and the replay check.

Functions
1 Lookup Key and next PN for transmit SA identified by AN
2 Lookup Key PN for receive SA identified by SCI, AN
3 Discard if received frame not VALID. Discard if replay check of PN for receive SA identified by SCI, AN fails. Updated replay check.

Figure 8-2—MACsec operation

NOTE 2—If the Current Cipher Suite supports extended packet numbering, the PN comprises 64 bits. The validation functions of the GCM-AES-XPN Cipher Suites (14.7, 14.8) use the SCI to identify a 32-bit Short Secure Channel Identifier (SSCI) supplied by the KaY and construct a 96-bit IV using that SSCI and the PN.

If the receive frame is valid, replay protection (if enabled) is applied, by checking that the received PN is not less than the lowest acceptable PN for the SA. If the check succeeds the parameters of the frame, unchanged from those transmitted, are presented to the MACsec client, and the lowest acceptable PN updated. The lowest acceptable PN can lag behind the received PN values, providing a window in which replay is tolerated, to allow receipt of frames that have been misordered by the network.

The format and encoding of each of the fields that comprise the SecTAG, including the support of different MACsec protocol versions is specified in Clause 9. The operation of the SecY that operates the MACsec protocol, the service that it provides, and the management control variables, error handling, and diagnostic information recorded is described in Clause 10.

9. Encoding of MACsec Protocol Data Units

This clause specifies the structure and encoding of the MACsec Protocol Data Units (MPDUs) exchanged between MAC Security Entities (SecYs). It

a) Specifies rules for the representation and encoding of protocol fields
b) Specifies the major components of each MPDU and the fields they comprise
c) Reviews the purpose of each field and the functionality provided
d) Specifies validation of the MPDU on reception
e) Documents the allocation of an EtherType value, the MACsec EtherType, to identify MPDUs.

NOTE—The MPDU validation checks specified in this clause are deliberately limited to ensuring successful decoding, and do not overlap with the specification of SecY operation (Clause 10).

9.1 Structure, representation, and encoding

All MPDUs shall contain an integral number of octets.

The octets in a MPDU are numbered starting from 1 and increasing in the order they are put into the MAC Service Data Unit (MSDU) that accompanies a request to or indication from the instance of the MAC Internal Sublayer Service (ISS) used by a SecY.

The bits in an octet are numbered from 1 to 8 in order of increasing bit significance, where 1 is the least significant bit in the octet.

Where octets and bits within a MPDU are represented using a diagram, octets shown higher on the page than subsequent octets and octets shown to the left of subsequent octets at the same height on the page are lower numbered, bits shown to the left of other bits within the same octet are higher numbered.

Where two or more consecutive octets are represented as hexadecimal values, lower numbered octet(s) are shown to the left and each octet following the first is preceded by a hyphen, e.g., 01-80-C2-00-00-00.

When consecutive octets are used to encode a binary number, the lower octet number has the more significant value.When consecutive bits within an octet are used to encode a binary number, the higher bit number has the most significant value. When bits within consecutive octets are used to encode a binary number, the lower octet number composes the more significant bits of the number. A flag is encoded as a single bit, and is set (True) if the bit takes the value 1, and clear (False) otherwise. The remaining bits within the octet can be used to encode other protocol fields.

9.2 Major components

Each MPDU comprises

a) A MAC Security TAG (SecTAG) (9.3)
b) Secure Data (9.10)
c) An integrity check value (ICV) (9.11).

Each of these components comprises an integral number of octets and is encoded in successive octets of the MPDU as illustrated in Figure 9-1.

NOTE—The MPDU does not include the source and destination MAC addresses, as these are separate parameters of the service requests and indications to and from the insecure service that supports MACsec.

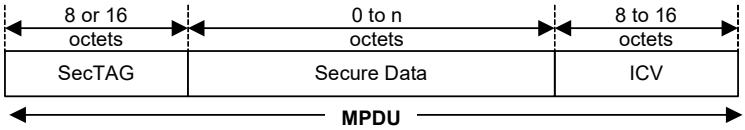


Figure 9-1—MPDU components

9.3 MAC Security TAG

The SecTAG is identified by the MACsec EtherType (9.4), and conveys the

a) TAG Control Information (TCI, 9.5)
b) Association Number (AN, 9.6)
c) Short Length (SL, 9.7)
d) Packet Number (PN, 9.8)
e) Optionally encoded Secure Channel Identifier (SCI, 9.9).

The format of the SecTAG is illustrated in Figure 9-2.

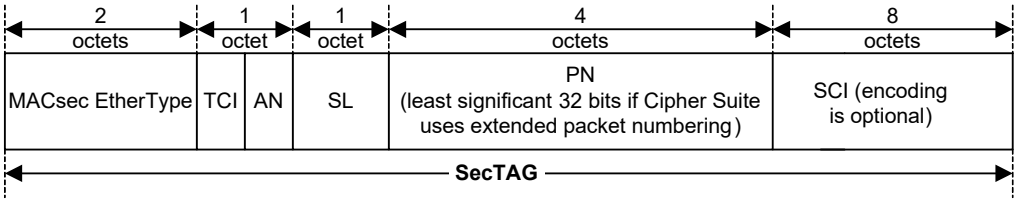


Figure 9-2—SecTAG format

9.4 MACsec EtherType

The MACsec EtherType (Table 9-1) comprises octet 1 and octet 2 of the SecTAG. It is included to allow

a) Coexistence of MACsec capable systems in the same environment as other systems
b) Incremental deployment of MACsec capable systems
c) Peer SecYs to communicate using the same media as other communicating entities
d) Concurrent operation of Key Agreement protocols that are independent of the MACsec protocol and the Current Cipher Suite
e) Operation of other protocols and entities that make use of the service provided by the SecY's Uncontrolled Port to communicate independently of the Key Agreement state.

Table 9-1—MACsec EtherType allocation

Tag Type	Name	Value
IEEE 802.1AE Security TAG	MACsec EtherType	88-E5

The encoding of the MACsec EtherType in the MPDU is illustrated in Figure 9-3.

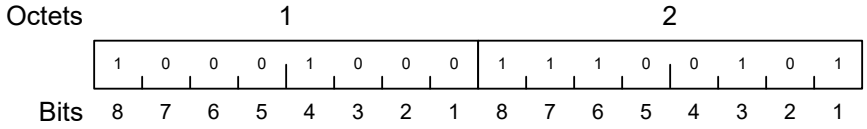


Figure 9-3—MACsec EtherType encoding

9.5 TAG Control Information (TCI)

The TCI field comprises bits 8 through 3 of octet 3 (Figure 9-4) of the SecTAG. These bits facilitate

a) Version numbering of the MACsec protocol without changing the MACsec EtherType
b) Optional use of the MAC Source Address parameter to convey the SCI
c) Optional inclusion of an explicitly encoded SCI (7.1.2, Figure 7-7)
d) Use of the EPON (Clause 12) Single Copy Broadcast (SCB) capability, without requiring an explicit SCI to distinguish the SCB Secure Channel
e) Extraction of the User Data from MPDUs by systems that do not possess the SAK (8.1.2, 8.1.4) when confidentiality is not being provided
f) Determination of whether confidentiality or integrity alone are in use.

The encoding of the MACsec TCI in the MPDU is illustrated in Figure 9-4.

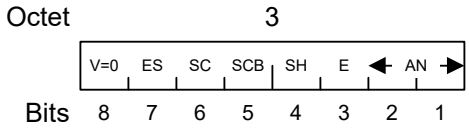


Figure 9-4—MACsec TCI and AN Encoding

The version number shall be 0 and is encoded in bit 8.

NOTE—Future versions of the MACsec protocol may use additional bits of the TCI to encode the version number. The fields and format of the remainder of the MPDU may change if the version number changes.

If the MPDU is transmitted by an end station and the first 6 octets of the SCI are equal to the value of the octets of MAC Source Address parameter of the ISS request in canonical format order, bit 7 [the End Station (ES) bit] of the TCI may be set. If the ES bit is set, bit 6 (the SC bit) shall not be set and an SCI shall not be explicitly encoded in the SecTAG. The ES bit is clear if the Source Address is not used to determine the SCI.

If an SCI (9.9, 7.1.2) is explicitly encoded in the SecTAG, bit 6 (the SC bit) of the TCI shall be set. The SC bit shall be clear if an SCI is not present in the SecTAG.

If and only if the MPDU is associated with the Secure Channel that supports the EPON Single Copy Broadcast capability, bit 5 (the SCB bit) of the TCI may be set. If the SCB bit is set, bit 6 (the SC bit) shall not be set and an SCI shall not be explicitly included in the SecTAG.

If the ES bit is set and the SCB is not set, the SCI comprises a Port Identifier (7.1.2) component of 00-01. If the SCB bit is set, the Port Identifier (7.1.2) component has the reserved SCB value of 00-00.

If the Encryption (E) bit is set and the Changed Text (C) bit is clear, the frame is not processed by the SecY (10.6) but is reserved for use by the KaY. Otherwise, the E bit is set if and only if confidentiality is being provided and is clear if integrity only is being provided and the C bit is clear if and only if the Secure Data is exactly the same as the User Data and the ICV is 16 octets long.

When the Default Cipher Suite (14.5) is used for integrity protection only, the Secure Data is the unmodified User Data, and a 16 octet ICV is used. Both the E bit and the C bit are therefore clear, and the data conveyed by MACsec is available to applications, such as network management, that need to see the data but are not trusted with the SAK that would permit its modification. Other Cipher Suites may also integrity protect data without modifying it, and use a 16 octet ICV, enabling read access to the data by other applications. The E and C bits are also clear for such Cipher Suites when integrity only is provided.

Some cryptographic algorithms modify or add to the data even when integrity only is being provided, or use an ICV that is not 16 octets long. The C bit is never clear for such an algorithm, even if the E bit is clear to indicate that confidentiality is not provided. Recovery of the data from a MACsec frame with the E bit clear and the C bit set requires knowledge of the Cipher Suite at a minimum. That information is not provided in the MACsec frame.

If both the C bit and E bit are set, confidentiality of the original User Data is being provided.

9.6 Association Number (AN)

The AN is encoded as an integer in bits 1 and 2 of octet 3 of the SecTAG (Figure 9-4) and identifies up to four different SAs within the context of an SC.

NOTE—Although each receiving SecY only needs to maintain two SAs per SC, the use of a 2-bit AN simplifies the design of protocols that update values associated with each of the SAs.

9.7 Short Length (SL)

SL is an integer encoded in bits 1 through 6 of octet 4 of the SecTAG and is set to the number of octets in the Secure Data (9.10) field, i.e., the number of octets between the last octet of the SecTAG and the first octet of the ICV, if that number is less than 48. Otherwise, SL is set to zero. If the number is zero then the frame is deemed not to have been short. The Secure Data field always comprises at least one octet.

Bits 7 and 8 of octet 4 shall be zero.

9.8 Packet Number (PN)

The 32 least significant bits of the PN are encoded in octets 5 through 8 of the SecTAG to

a) Provide a unique IV PDU for all MPDUs transmitted using the same SA
b) Support replay protection.

NOTE 1—The IV used by the Default Cipher Suite GCM-AES-128 (14.5) and the GCM-AES-256 Cipher Suite (14.6) comprises the SCI (even if the SCI is not transmitted in the SecTAG) and a 32-bit PN. Subject to proper unique MAC Address allocation procedures, the SCI is a globally unique identifier for a SecY. To satisfy the IV uniqueness requirements of Counter mode of operation, a fresh key is used before PN values are reused.

NOTE 2—If the Current Cipher Suite provides extended packet numbering, i.e., uses a 64-bit PN, the 32 least significant bits of the PN are conveyed in this SecTAG field and the 32 most significant bits are recovered on receipt as specified in 10.6. The IV used by the GCM-AES-XPN Cipher Suites (14.7, 14.8) is constructed from a 32-bit SSCI distributed by key agreement protocol and unique for each SCI within the scope of the CA (and hence within potential users of the same SAK) and the 64-bit non-repeating PN.

9.9 Secure Channel Identifier (SCI)

If the SC bit in the TCI is set, the SCI (7.1.2, 8.2.1) is encoded in octets 9 through 16 of the SecTAG and facilitates

a) Identification of the SA where the CA comprises three or more SCs
b) Network management identification of the SecY that has transmitted the frame.

Octets 9 through 14 of the SecTAG encode the System Identifier component of the SCI. This comprises the six octets of a MAC address uniquely associated with the transmitting SecY. The octet values and their sequence conform to the Canonical Format specified by IEEE Std 802.

Octets 15 and 16 of the SecTAG encode the Port Identifier component of the SCI, as an integer.

The 64-bit value FF-FF-FF-FF-FF-FF-FF-FF is never used as an SCI and is reserved for use by implementations to indicate the absence of an SC or an SCI in contexts where an SC can be present.

An explicitly encoded SCI field in the SecTAG is not required on point-to-point links, which are identified by the operPointToPointMAC status parameter of the service provider, if the transmitting SecY uses only one transmit SC. In that case, the secure association created by the SecY for the peer SecYs, together with the direction of transmission of the secured MPDU, can be used to identify the transmitting SecY. Therefore, an explicitly encoded SCI is unnecessary. Although the SCI does not have to be repeated in each frame when only two SecYs participate in a CA (see Clause 8, Clause 9, and Clause 10), the SCI (for Cipher Suites using a 32-bit PN) or the SSCI (for Cipher Suites using a 64-bit PN) still forms part of the cryptographic computation.

9.10 Secure Data

The Secure Data comprises all the octets that follow the SecTAG and precede the ICV. The Secure Data field is never of zero length, since the primitives of the MAC Service require a non-null MSDU (User Data) parameter.

NOTE 1—In practice, if the MSDU composed by the operation of the current Cipher Suite following MPDU reception contains less than two octets, it will be discarded by the user of the SecY's controlled port, since it is too short to contain an EtherType or an LLC length field. Such discard is, however, determined by the user of the Controlled Port and not by the SecY itself.

NOTE 2—Ethernet transports frames of a minimum size, and provides no explicit indication of PDU length if the PDU is composed of fewer octets. The SL field allows the originator of the frame, which is not necessarily aware of the need of an intervening Ethernet component to pad the frame, to specify the number of octets in the MPDU, thus allowing the receiver to unambiguously locate the ICV.

9.11 Integrity check value (ICV)

The length of the ICV is Cipher Suite dependent, but is not less than 8 octets and not more than 16 octets, depending on the Cipher Suite.

NOTE—The ICV protects the destination and source MAC address parameters, as well as all the fields of the MPDU.

9.12 PDU validation

A received MPDU is valid if and only if it comprises a valid SecTAG, one or more octets of Secure Data, and an ICV, i.e.,

a) It comprises at least 17 octets.
b) Octets 1 and 2 compose the MACsec EtherType.
c) The V bit in the TCI is clear.
d) If the ES or the SCB bit in the TCI is set, then the SC bit is clear.
e) Bits 7 and 8 of octet 4 of the SecTAG are clear.
f) If the C and SC bits in the TCI are clear, the MPDU comprises 24 octets plus the number of octets indicated by the SL field if that is non-zero and at least 72 octets otherwise.
g) If the C bit is clear and the SC bit set, then the MPDU comprises 32 octets plus the number of octets indicated by the SL field if that is non-zero and at least 80 octets otherwise.
h) If the C bit is set and the SC bit clear, then the MPDU comprises 8 octets plus the minimum length of the ICV as determined by the Cipher Suite in use at the receiving SecY, plus the number of octets indicated by the SL field if that is non-zero and at least 48 additional octets otherwise.
i) If the C and SC bits are both set, the frame comprises at least 16 octets plus the minimum length of the ICV as determined by the Cipher Suite in use at the receiving SecY, plus the number of octets indicated by the SL field if that is non-zero and at least 48 additional octets otherwise.

10. Principles of MAC Security Entity (SecY) operation

This clause

a) Provides an overview of the SecY (10.1), the service that it provides, and its relationship to other entities in a secure system including its associated MACsec Key Agreement Entity (KaY).
b) Describes the functionality of the SecY (10.2).
c) Provides a model of operation (10.3) comprising an architecture (10.4) and its constituent processes (10.5 through 10.7) that supports the detailed functionality including management controls.
d) Details the addressing requirements and specifies the addressing of SecYs (10.8).

NOTE—Clause 6 defines the properties of the secure MAC Service, Clause 7 describes the security relationships used to support the service and how the service is used, providing the context within which each SecY operates, Clause 8 sets out requirements for the MACsec protocol and introduces the operation of the protocol, and Clause 9 specifies the encoding of parameters in MPDUs. This clause does not repeat all the information provided in those prior clauses, but includes sufficient reference to facilitate an understanding of SecY operation. Clause 7 of IEEE Std 802.1AC-2016 describes the basic architectural concepts and terms used in this clause, including service, service access point, service primitive, and ports.

10.1 SecY overview

Each SecY uses the MAC Service provided by a Common Port (10.4) to provide one instance of the secure MAC Service (Clause 6) to the user of its Controlled Port and one instance of insecure service to the user of its Uncontrolled Port (Figure 10-1).

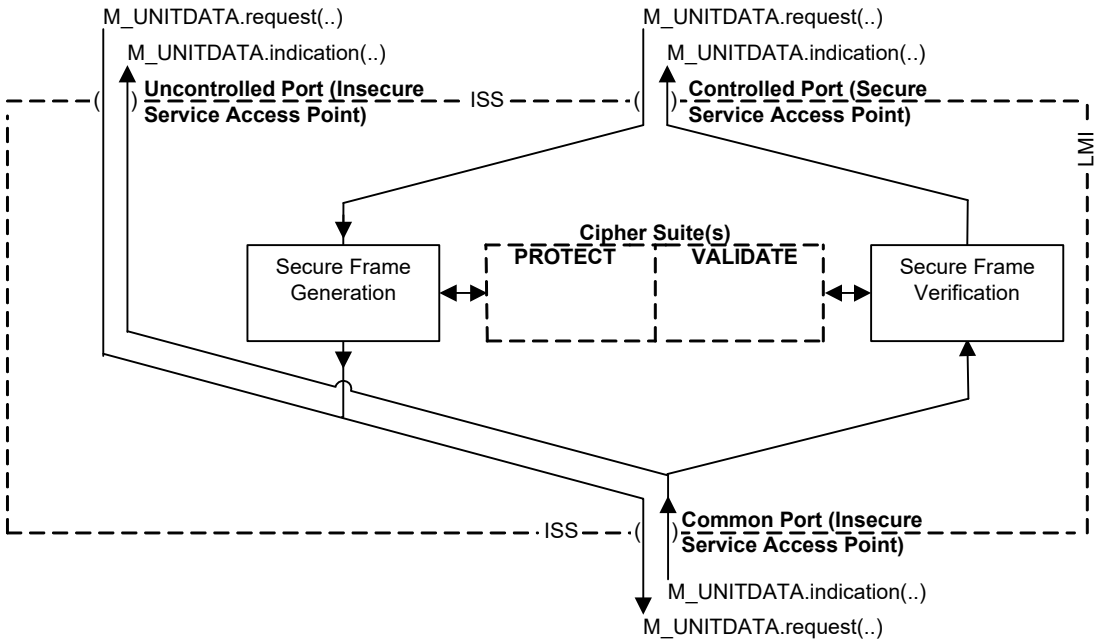


Figure 10-1—SecY

The integrity and origin of the parameters of each service request and indication accepted from and delivered to the Controlled Port are protected and validated by the SecY. The SecY may also encrypt to provide user data confidentiality. If the parameters that accompany a service indication at the Common Port are not successfully validated as required by management controls, no service indication will occur at the Controlled Port and the received parameters will be discarded.

Each service request made by the user of a SecY's Uncontrolled Port results in an identical request at the Common Port, and each service indication received from the Common Port results in an identical indication to the user of its Uncontrolled Port in addition to any indication at the Controlled Port.

NOTE 1—Some frames received at the Uncontrolled Port will be discarded because they can only be useful to a SecY supporting the associated Controlled Port.

The relative order of Common Port indications and the corresponding indications at the Uncontrolled Port and the Controlled Port is not defined, save that the order of indications from one Port to another Port is preserved. Similarly the relative order of user requests at the Uncontrolled and Controlled Ports does not define the order of requests to the Common Port. The interval between any request or indication and the SecY making a corresponding request or indication shall not exceed the bounds specified in Table 10-3.

The specification of the cryptographic algorithms used at any time to provide integrity and confidentiality, together with the values of parameters (for example, key size) used by those algorithms, compose a Cipher Suite (Clause 14). This standard mandates a default Cipher Suite that can provide integrity protection only or both integrity and confidentiality. A SecY may implement additional Cipher Suites. This standard only permits the use of Cipher Suites that meet well defined criteria (14.2, 14.3).

The KaY is part of the Port Access Entity (PAE, IEEE Std 802.1X) associated with the SecY and uses the service provided by the Uncontrolled Port to transmit and receive frames that support key agreement protocols. These frames are distinguished by EtherType, so other selected protocol entities can also communicate using insecure frames by making use of the Uncontrolled Port.

The KaY determines the value of the MAC_Operational parameter (IEEE Std 802.1AC) associated with Controlled Port (10.7.4, 10.7.5) consistent with the provisions of this standard (6.4, 6.5, 6.7, 7.1.3, 7.2, 10.5.1, 10.5.2, 10.7.14, 10.7.2, 10.7.25).

The KaY communicates transmit and receive keys and other information (10.2) to the SecY through its Layer Management Interface (LMI). The LMI is also used to exchange information with local protocol entities responsible for network management, such as an SNMP Agent.

NOTE 2—The term *local* refers to any other entity residing within the same system. Information exchange with a local entity can be modelled as occurring through its LMI (10.1, 10.3, 10.4, Figure 10-1), thus facilitating information exchange between entities not necessarily adjacent in a protocol layer reference model. No constraints are placed on the information exchanged, but there is no synchronization with any particular invocation of service at a service access point, so LMI exchanges do not effectively add to the parameters of a service such as the MAC service.

10.2 SecY functions

Each SecY supports

a) Secure transmission of the parameters of service requests made by the user of its Controlled Port
b) Insecure transparent transmission from the Uncontrolled Port
c) Reception, verification, and delivery of secure service indications to the Controlled Port
d) Reception and transparent delivery of service indications to the Uncontrolled Port
e) MAC Status (6.4) and point-to-point parameters (6.5) for the Uncontrolled and Controlled Ports.

Management controls that support deployment (8.1.4) of MACsec include

f) Transmission and reception by the user of the Controlled Port without frame modifications
g) Reception without integrity checking
h) Use of multiple transmit SCs and a configurable replayWindow to support media access control methods and provider networks that can misorder frames with different priorities and/or addresses.

Selection of a Cipher Suite, CA establishment, and SA support, is supported by allowing the KaY to

i) Discover which Cipher Suites are implemented and how many receive SCs each can support
j) Select the Current Cipher Suite
k) Identify the SCs to be used to support reception for the CA
l) Provide transmit and receive SAKs for identified SAs
m) Confirm that SAKs have been installed, i.e., are ready for use
n) Monitor the PN used for transmission, in order to provide new SAKs prior to PN exhaustion.

Operational and diagnostic controls and statistics provide

o) Administrative control over the optional security tagging capabilities of the SecY
p) A count of frames intended for transmission but discarded as too long for the Common Port
q) Counts of received frames without the MACsec EtherType, discarded by validation checks, without SCIs when the LAN connectivity is not restricted to point-to-point communication, identified as belonging to unknown SCs, identified as belonging to an SA that is not in use, failing the replay check, failing the integrity check, and delivered to the user.

NOTE—Except where explicitly specified otherwise, throughout this standard the term *user* refers to the user of the MAC service instance provided by the Controlled Port, and the term *provider* refers to the instance of protocol and procedures that provides the MAC service instance to the SecY at the Common Port.

10.3 Model of operation

The model of operation in this clause is simply a basis for describing the functionality of a SecY. It is in no way intended to constrain 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.

10.4 SecY architecture

A SecY uses an instance of the MAC Internal Sublayer Service (ISS, 6.1), referred to as the Common Port, to provide a secured instance of the ISS, the Controlled Port, and an insecure instance of the ISS, the Uncontrolled Port, that provides transparent transmission and reception through the Common Port.

The architecture of a SecY is illustrated in Figure 10-2 and comprises

a) The Controlled, Uncontrolled, and Common Ports together with their MAC Status parameters
b) The Secure Frame Generation process (10.5)
c) The Secure Frame Verification process (10.6)
d) Cipher Suite protection of transmitted frames and validation of received frames (8.2, Clause 14)
e) A Transmit Multiplexer and a Receive Demultiplexer
f) Optional transmit and receive frame check sequence (FCS) regenerators
g) A SecY Management process (10.7).

The Transmit Multiplexer accepts transmit requests from the Uncontrolled Port and the Secure Frame Generation process for the Controlled Port and submits corresponding requests to the Common Port. The Receive Demultiplexer submits each indication from the Common Port to the Uncontrolled Port and to the Secure Frame Verification process for the Controlled Port.

NOTE 1—This specification most clearly sets out the resulting behavior of a conforming implementation. Real implementations can implement the behavior in any way that yields the same externally visible behavior (including the values of management counters). For example, examination of the specification in this clause shows that there need be no implementation burden corresponding to duplication of the received frame if validateFrames is Strict and none of the users of the Uncontrolled Port make use of the MACsec EtherType. .

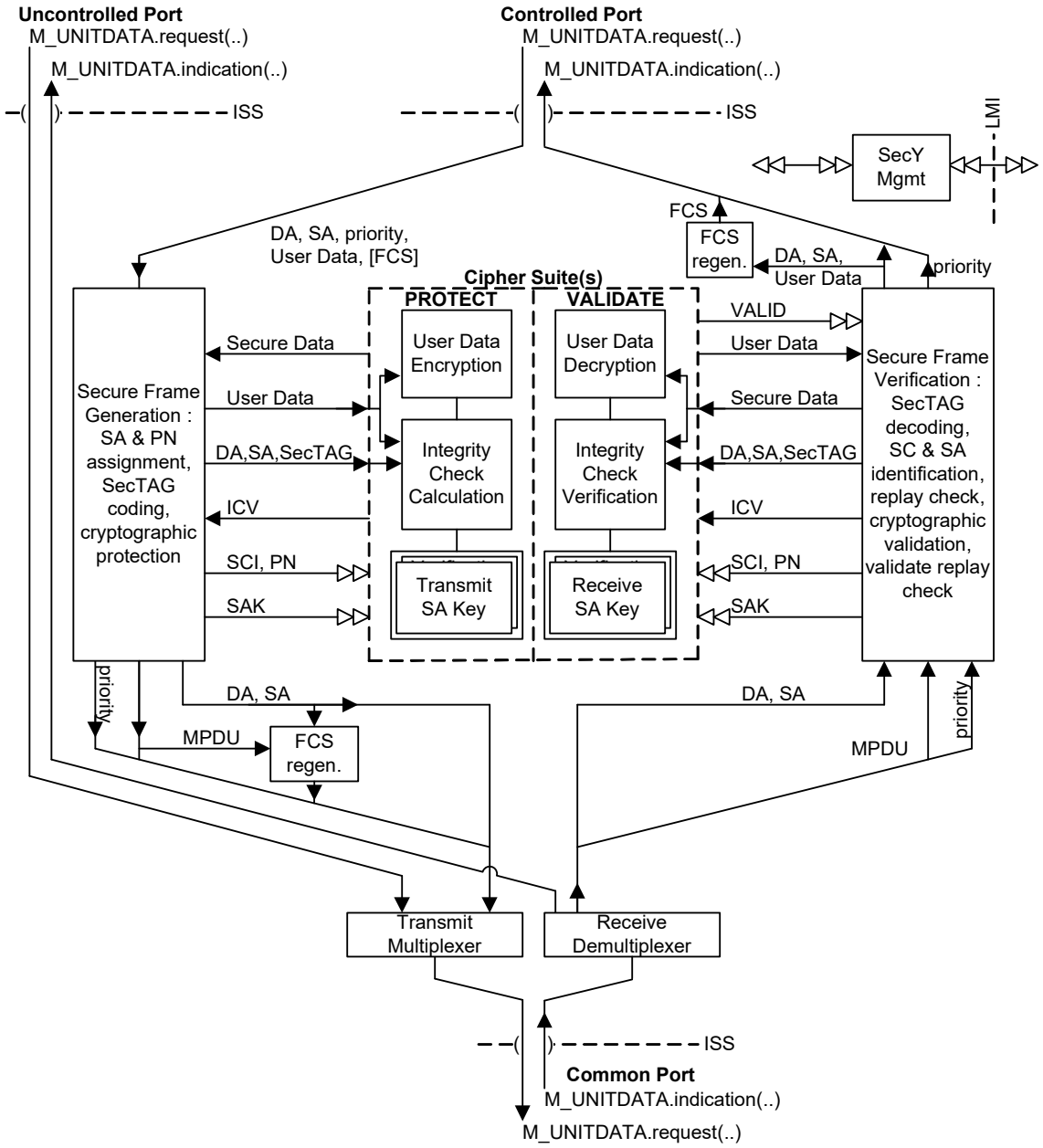


Figure 10-2—SecY architecture and operation

A Layer Management Interface (LMI) is used by the SecY Management process to communicate the capabilities of the SecY, its controls, status, protocol, management events, and counters to and from other entities that compose the secure system containing the SecY.

Management controls are provided to allow a SecY to be incorporated in a network system before MACsec is deployed, and to facilitate staged deployment. If protectFrames is not set, frames submitted to the Controlled Port are transmitted without modification. The validateFrames control allows untagged frames to be received, and Cipher Suite validation of tagged frames to be disabled or its result simply counted without frame discard. The replayProtect and replayWindow controls allows replay protection to be disabled, to

operate on a packet number window, or to enforce strict frame order. If replayProtect is set but the replayWindow is not zero, frames within the window can be received out of order; however, they are not replay protected. Management counters allow configuration and operational errors to be identified and rectified before enabling secure operation. The effect of the controls, and the counters maintained, are summarized in Figure 10-3 and Figure 10-4.

The FCS can be included as a parameter of an M_UNITDATA.request or M_UNITDATA.indication primitive. When the data that is within the FCS coverage is modified by the addition of an ICV or encryption of the user data, the FCS changes. The SecY shall not introduce an undetected frame error rate greater than that which would have been achieved by preserving the original FCS (6.10).

NOTE 2—There are number of possibilities for changing FCS without diminishing the coverage provided. One is to generate a new FCS by algorithmically modifying the received FCS, based on knowledge of the FCS algorithm and the transformations that the frame has undergone between reception and transmission.

10.5 Secure frame generation

For each transmit request at the Controlled Port, the Secure Frame Generation process

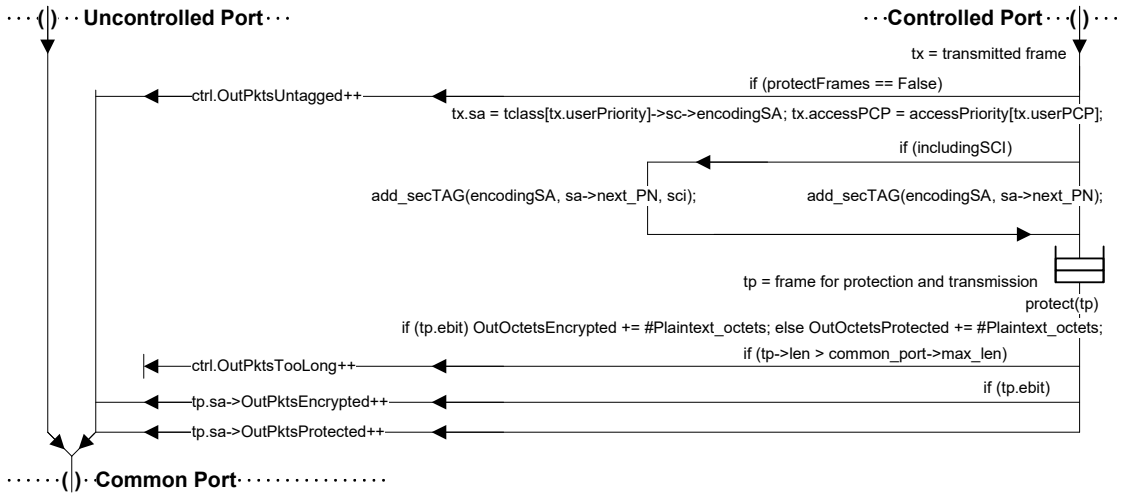
a) Assigns the frame to an SA (10.5.1).
b) Assigns the nextPN variable for that SA to be used as the value of the PN for that protected frame (10.5.2).
c) Encodes the octets of the SecTAG including the least significant 32 bits of the PN in the PN field (10.5.3).
d) Provides the protection function (14.1, 10.5.4) of the Current Cipher Suite with
    1) The SAK
    2) The SCI for the SC used by the SecY to transmit
    3) The PN
    4) The SecTAG
    5) The sequence of octets that compose the User Data.
e) Receives the following parameters from the Cipher Suite protection operation:
    6) The sequence of octets that compose the Secure Data
    7) The ICV
f) Issues a request to the Transmit Multiplexer with the destination and source MAC addresses and an MPDU comprising the octets of the SecTAG, Secure Data, and the ICV concatenated in that order (10.5.5). If the SecY does not implement an Access Priority Table (10.7.17), the priority of the request is the same as that received from the Controlled Port; otherwise, it is the access priority given by the table for the received priority.

If the management control protectFrames is False, the preceding steps are omitted, an identical transmit request is made to the Transmit Multiplexer, and the OutPktsUntagged counter incremented.

NOTE—This model of operation supports the externally observable behavior that can result when the Cipher Suite implementation calculates the Secure Data and ICV parameters for a number of frames in parallel, and the responses to protection and validation requests are delayed. Transmitted frames are not misordered.

10.5.1 Transmit SA assignment

Each frame is assigned to the SA identified by the current value of the encodingSA variable for the selected transmit SC. If the SecY does not implement a Traffic Class Table it uses a single transmit SC. If implemented, the Traffic Class Table specifies the value of the most significant four bits of the SCI's Port Identifier component for each possible transmit request user priority, allowing selection of one of up to eight distinct SCs (see 10.7.17).



Tests and their consequences are annotated in this diagram using the computer language 'C ++' (ISO/IEC 14882), with variable names corresponding to abbreviations of the text of this clause (10), which takes precedence.

NOTE—Secure generation frame counters are identified as reported by management. Confidentiality or integrity only protection is selected for an SA when it is created, so either but not both of the OutOctetsEncrypted or OutOctetsProtected counts and either OutPktsEncrypted or the OutPktsProtected will be incremented while that SA is in use, and the current value of the packet counter can be derived from nextPN for the SA less any change in the value of OutPktsTooLong since that SA has been used for protection, allowing an implementation to optimize counter resources.

Figure 10-3—Management controls and counters for secure frame generation

The encodingSA is updated following an LMI request from the KaY to start transmitting using the SA and can be read but not written by network management. Frames will be protected using the encodingSA immediately after the last frame assigned to the previous SA has been protected. If the SA is not available for use, and the management control protectFrames is set, MAC_Operational transitions to False for the Controlled Port, and frames are neither accepted or delivered using the port.

10.5.2 Transmit PN assignment

The frame's PN is set to the value of nextPN for the SA, and nextPN is incremented. If the nextPN variable for the encodingSA is zero (or 2^{32} if the Current Cipher Suite does not support extended packet numbering, 2^{64} if it does) and the protectFrames control is set, MAC_Operational transitions to False for the Controlled Port and frames are neither accepted or delivered. The initial value of nextPN is set by the KaY via the LMI prior to use of the SA, and its current value can be read both while and after the SA is used to transmit frames. The value of nextPN can be read, but not written, by network management.

10.5.3 SecTAG encoding

The SecTAG is encoded as specified in Clause 9.

The SC bit in the SecTAG shall be set and the SCI explicitly encoded in the SecTAG, and the management status parameter includingSCI set to True, if and only if

a) The management control alwaysIncludeSCI is True,
   or
b) The number of transmit SCs is greater than one,
   or
c) The number of receive SCs enabled for reception is greater than one, and
   1) The management control useES is False,
      and
   2) The management control useSCB is False.

If the management control useES is True and includingSCI is False, the ES bit in the SecTAG shall be set. Otherwise, if useES is False or includingSCI is True, the ES bit shall be clear.

If the management control useSCB is True and includingSCI is False, the SCB bit in the SecTAG shall be set. Otherwise, if useSCB is False or includingSCI is True, the SCB bit shall be clear.

NOTE—These rules cover the case where useSCB is True and the number of active receive channels is greater than one. However, SCB bit use is currently restricted to supporting a transmit only EPON interface (see Clause 12).

Table 10-1 summarizes the rules [a) through c) above], with each of the columns to the right representing a valid combination of controls, number of SCs, and SecTAG encoding.

Table 10-1—Management controls and SecTAG encoding

Mgmt controls	alwaysIncludeSCI	T[a]	F	F	F	F	F
	useES	—	—	F	T	T	F
	useSCB	—	—	F	T	F	T
#SCs	#transmitSCs > 1	—	T	—	F	F	F
	#receiveSCs enabled for reception > 1	—	—	T	—	—	—
Mgmt status	includingSCI	T	T	T	F	F	F
SecTAG encoding	SC bit set? (SCI explicitly encoded)	Y	Y	Y	N	N	N
	ES bit set?	N	N	N	Y	Y	N
	SCB bit set?	N	N	N	Y	N	Y

[a]T = True, F = False, — = don't care, Y= Yes, N = No

The values of useES, useSCB, and alwaysIncludeSCI can be written and read by management. The read-only management status parameter includingSCI is True if an SCI is explicitly encoded in each SecTAG, and False otherwise. The number of active receive SCs is controlled by the KaY but can be read by management.

If a frame is to be integrity protected, but not encrypted, with the number and value of the octets of the Secure Data exactly the same as those of the User Data, and an ICV of 16 octets, then the E bit shall be clear and the C bit clear. The E bit shall be clear and the C bit set if the frame is not encrypted but the octets of the Secure Data differ from those of the User Data or the ICV is not 16 octets.

If both confidentiality (through encryption) and integrity protection are applied to a frame then both the E bit and the C bit shall be set. The SecY shall not encode a SecTAG that has both the E bit set and the C bit clear for any frame received from the Controlled Port for transmission.

10.5.4 Cryptographic protection

If the Cipher Suite is currently protecting frames using the previous SA and its SA Key, as reflected by the value of the encipheringSA, the frame can be queued awaiting protection. The value of the encipheringSA is updated, and protection of the frame parameters is started within a minimum frame size transmission delay, after the last frame has been protected using the previous key.

The use of each of the Cipher Suites specified by this standard is specified in Clause 14, which takes precedence over any explanation in this or other clauses.

The appropriate octet counter is incremented by the number of octets in the User Data (OutOctetsEncrypted if confidentiality protection was provided, and OutOctetsProtected otherwise).

10.5.5 Transmit request

If the MPDU composed of the concatenated octets of the SecTAG, Secure Data, and ICV exceeds the size of the MSDU supported by the Common Port, the frame is discarded and a counter incremented. Details of the discarded frame may be recorded to assist network management resolution of the problem. Otherwise, the parameters of the service request are submitted to the Transmit Multiplexer.

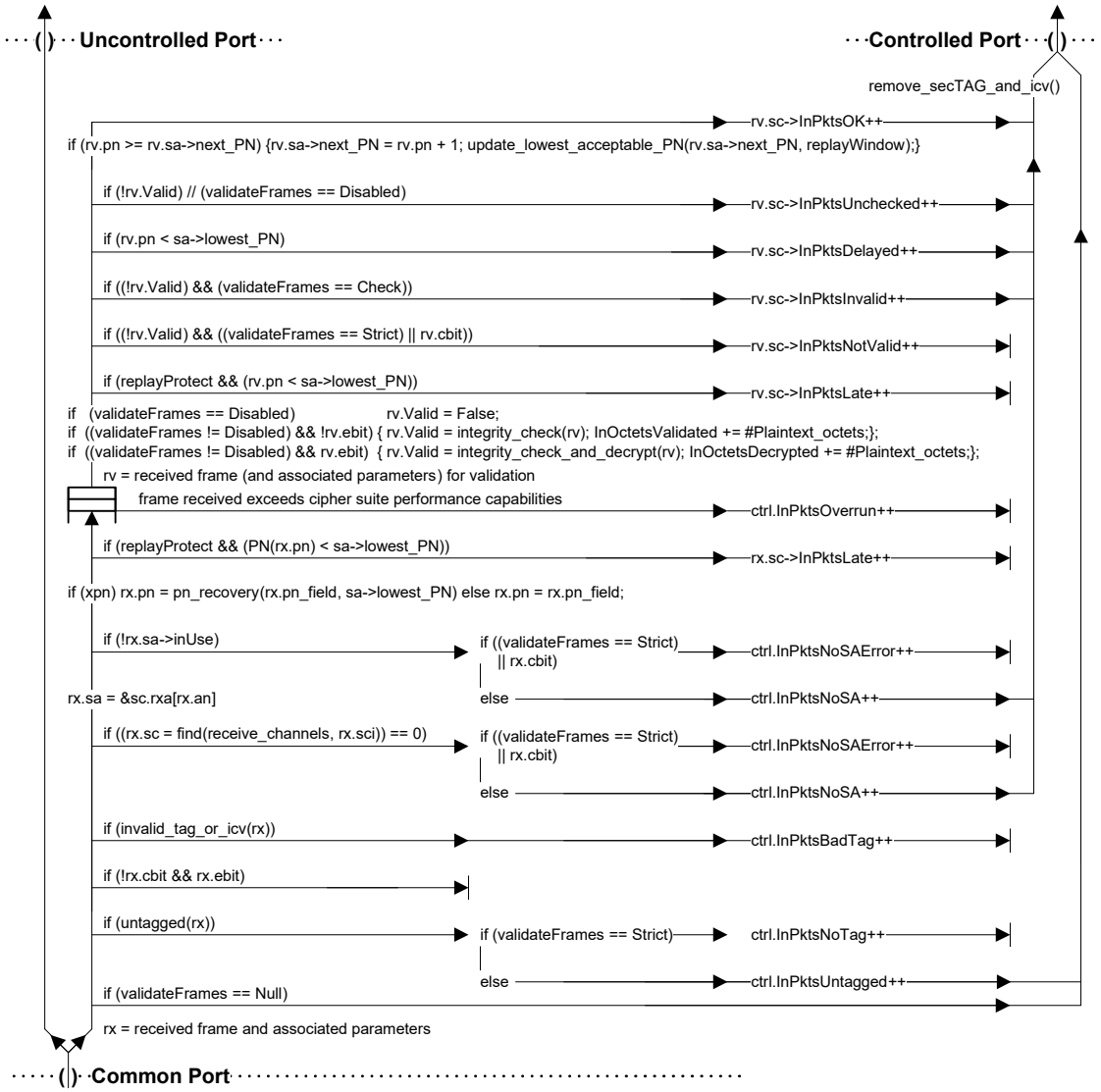
10.6 Secure frame verification

For each receive indication from the Receive Demultiplexer, the Secure Frame Verification process

a) Examines the user data for a SecTAG.
b) Validates frames with a SecTAG as specified in 9.12.
c) Extracts and decodes the SecTAG as specified in 9.3 through 9.9.
d) Extracts the User Data and ICV as specified in 9.10 and 9.11.
e) Assigns the frame to an SA (10.6.1).
f) Recovers the PN and performs a preliminary replay check against the last validated PN for the SA (10.6.2).
g) Provides the validation function (14.1, 10.6.3) of the Current Cipher Suite with
    1) The SAK
    2) The SCI for the SC used by the SecY to transmit
    3) The PN
    4) The SecTAG
    5) The sequence of octets that compose the Secure Data
    6) The ICV.
h) Receives the following parameters from the Cipher Suite validation operation
    1) A Valid indication, if the integrity check was valid and the User Data could be recovered
    2) The sequence of octets that compose the User Data.
i) Updates the replay check (10.6.4).
j) Issues an indication to the Controlled Port with the DA, SA, and priority of the frame as received from the Receive Demultiplexer, and the User Data provided by the validation operation (10.6.5).

If the management control validateFrames is not Strict, frames without a SecTAG are received, counted, and delivered to the Controlled Port; otherwise, they are counted and discarded. If validateFrames 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 validateFrames is Null, all received frames are delivered to the Controlled Port without modification, irrespective of the absence, presence, or validity of a SecTAG, and the processing described in a) through j) above and in 10.6.1 through 10.6.5 is not performed. Figure 10-4 summarizes the operation of secure frame verification management controls and counters.

Setting validateFrames to Null shall also cause the secure frame generation control protectFrames (10.5) to become False, thus allowing a port that includes a SecY to behave as if the SecY were not present. In particular, it allows a MACsec-capable bridge or EDE to forward frames that have a SecTAG but no other outer tag (such as a VLAN tag).



Tests and their consequences are annotated in this diagram using the computer language ‘C ++’ (ISO/IEC 14882), with variable names corresponding to abbreviations of the text of this clause (10), which takes precedence.

NOTE—Secure verification frame counters are identified as reported by management. Whether a given counter can be incremented depends on the management control validateFrames and on whether received frames are confidentiality protected, allowing an implementation to optimize resources. As shown in the figure, only one counter for each of the sets {InPktsUntagged, InPktsNoTag} and {InPktsNoSA, InPktsNoSAError} for the Controlled Port as a whole and only one counter for each of the sets {InPktsLate, InPktsDelayed}, {InPktsInvalid, InPktsNotValid}, and {InPktsUnchecked, InPktsOK} for each received SC can be incremented while validateFrames and confidentiality policy remain unchanged.

Figure 10-4—Management controls and counters for secure frame verification

10.6.1 Receive SA assignment

An SCI is associated with the received frame and used to locate the receive SC. If an SCI is not explicitly encoded in the SecTAG, the value established by the KaY for a single peer is used.

If the SC is not found, the received SCI may be recorded to assist network management resolution of the problem, and

a) If validateFrames is Strict or the C bit in the SecTAG is set, the InPktsNoSAError counter is incremented and the frame is discarded; otherwise

b) The InPktsNoSA counter is incremented and the frame (with the SecTAG and ICV removed) is delivered to the Controlled Port.

If the receive SC has been identified, the frame's AN is used to locate the receive SA received frame and processing continues with the preliminary replay check. If the SA is not in use:

c) If validateFrames is Strict or the C bit is set, the frame is discarded and the InPktsNoSAError counter incremented; otherwise

d) The InPktsNoSA counter is incremented and the frame delivered to the Controlled Port.

NOTE—The short phrase "the frame is discarded" is commonly used to express the more formal notion of not processing a service primitive (an indication or request) further and recovering the resources that embody the parameters of that service primitive. No further processing is applied. However, if a duplicate of the primitive has been submitted to another process (by the Receive Demultiplexer in this case) processing of that duplicate is unaffected.

10.6.2 PN recovery and preliminary replay check

If the Current Cipher Suite does not use extended packet numbering, i.e., the PN comprises 32 bits, the value of the PN is that decoded from the 4 octet PN field in the SecTAG of the received frame (9.1, 9.8).

If the Current Cipher Suite supports extended packet numbering, the PN comprises 64 bits. The least significant 32 bits of the PN are those decoded from the PN field in the SecTAG of the received frame. The 32 most significant bits of the PN are recovered for each received frame by applying the assumption that they have remained unchanged since their use in the frame with the lowest acceptable PN—unless the most significant of the 32 least significant bits of the lowest acceptable PN is set and the corresponding bit of the received PN is not set, in which case the value of the 32 most significant bits of the PN is one more than the value of the 32 most significant bits of the lowest acceptable PN. Table 10-2 provides examples.

Table 10-2—Extended packet number recovery (examples)

SecTAG PN field value	0x 2A2B 5051
Lowest acceptable PN	0x 0000 0007 1234 DEF0
PN	0x 0000 0007 2A2B 5051
SecTAG PN field value	0x 2A2B 5051
Lowest acceptable PN	0x 0000 0007 8234 DEF0
PN	0x 0000 0008 2A2B 5051
SecTAG PN field value	0x 9A2B 5051
Lowest acceptable PN	0x 0000 0007 8234 DEF0
PN	0x 0000 0007 9A2B 5051
SecTAG PN field value	0x 9A2B 5051
Lowest acceptable PN	0x 0000 0007 2234 DEF0
PN	0x 0000 0007 9A2B 5051

The recovered PN value is not guaranteed to be the same as that used by the transmitter to protect the frame, but all PN values in the range lowest acceptable PN to lowest acceptable PN plus 2^{31} will be recovered correctly. If the recovered PN value is incorrect, the Cipher Suite validation operation will not return VALID and the frame will be discarded if validateFrames is Strict (10.6.5, 10.7.8). A recovered PN value is used to update the lowest acceptable PN only if the validation operation with that PN value returns VALID.

NOTE 1—For a discussion of the PN recovery algorithm, its incidental properties and alternatives, that goes beyond the normative requirements of this standard, see Seaman [B15].

NOTE 2—If a large number of successive frames were to be lost (2^{30}–1, corresponding to approximately 9 seconds of full utilization of a 400 Gb/s link by minimum sized Ethernet frames) subsequent receipt of MACsec frames might fail to establish a correct PN value. The MACsec Key Agreement protocol (MKA) specified in IEEE Std 802.1X and its amendments communicates the value of the high order bits periodically to recover from this eventuality.

If replayProtect control is enabled and the PN recovered from the received frame is less than the lowest acceptable packet number (see 10.6.5) for the SA, the frame is discarded and the InPktsLate counter incremented.

NOTE 3—If the SC is supported by a network that includes buffering with priority queueing, such as a provider bridged network, delivered frames can be reordered.

10.6.3 Cryptographic validation

The frame can be queued awaiting validation. If the frame reception rate exceeds the Cipher Suite's validation capabilities, the frame may be discarded and the InPktsOverrun counter incremented.

If the validateFrames control is Disabled, the Cipher Suite validation is not used to validate the frame.

If validateFrames is not Disabled, and the E bit in the SecTAG is set, the Cipher Suite is used to validate and decrypt the frame. If the Cipher Suite does not provide confidentiality protection, it shall not return VALID. The InOctetsDecrypted counter is incremented by the number of octets in the resulting User Data (or an estimate of that number, if VALID is not returned).

If validateFrames is not Disabled, and the E bit in the SecTAG is clear, the Cipher Suite is used to validate the frame. If the Cipher Suite does not provide integrity protection without confidentiality it shall not return VALID. The InOctetsValidated counter is incremented by the number of octets in the resulting User Data (or an estimate of that number, if VALID is not returned).

The frame is marked valid if the Cipher Suite is used and returns VALID and is marked invalid otherwise. The use of each of the Cipher Suites specified by this standard is specified in Clause 14, which takes precedence over any explanation in this or other clauses.

10.6.4 Replay check update

If the PN of the received frame is less than the lowest acceptable packet number for the SA, and replayProtect is enabled, the frame is discarded and the InPktsLate counter incremented.

NOTE—This model of operation assumes that any queuing within the verification process occurs prior to frame validation, and the check described uses the lowest acceptable PN updated by prior frames as described in 10.6.5. Implementations can process frames as convenient, provided the externally observable result is the same.

10.6.5 Receive indication

If the received frame is marked as invalid, and the validateFrames control is Strict or the C bit in the SecTAG was set, the frame is discarded and the InPktsNotValid counter incremented. Otherwise, the frame is delivered to the Controlled Port, and the appropriate counter incremented as follows:

a) If the frame is not valid and validateFrames is set to Check, InPktsInvalid; otherwise,
b) If the received PN is less than the lowest acceptable PN (treating a 32-bit PN value of zero as 2^{32} and a 64-bit PN value of zero as 2^{64}), InPktsDelayed; otherwise,
c) If the frame is not valid, InPktsUnchecked; otherwise,
d) InPktsOK.

If the PN for the frame was equal to or greater than the nextPN variable for the SA and the frame is valid, nextPN is set to the value for the received frame, incremented by one. The lowest acceptable PN variable is set to the greater of its existing value and the value of nextPN minus the replayWindow variable.

NOTE—The lowest acceptable packet number can also be set or incremented by the KaY to ensure timely delivery.

10.7 SecY management

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

a) Reports the value of the SCI for the SecY's default traffic class SC (10.7.1).
b) Maintains the MAC Status (6.4) parameters and point-to-point MAC parameters (6.5) for the Uncontrolled (10.7.2) and Controlled (10.7.4) Ports.
c) Provides interface statistics for the Uncontrolled (10.7.3) and Controlled Ports (10.7.6), deriving the latter from the detailed statistics maintained by the SecY.
d) Provides information on the frame verification (10.7.7) and generation (10.7.16) capabilities.
e) Supports control of frame verification (10.7.8) and generation (10.7.17), including management of a Traffic Class Table that allows the user priority associated with the Controlled Port transmit request to select one of a number of transmit SCs, and an Access Priority Table.
f) Supports creation of transmit SCs (10.7.20), each corresponding to one of the values appearing in Traffic Class Table entries.
g) Supports creation of transmit SAs (10.7.22), each associated with an SAK, for the transmit SC.
h) Supports creation of receive SCs (10.7.11), each corresponding to potential member of the CA.
i) Supports creation of receive SAs (10.7.13) for each receive SC, each associated with an SAK.
j) Supports control over reception (10.7.15) and transmission (10.7.24) using individual SAs, and allows the lowest acceptable PN to be set and updated for reception.
k) Maintains statistics for receive and transmit SCs and SAs, accumulating statistics from past SAs.
l) Provides a list of the Cipher Suites with their basic capabilities and properties, and a list of those Cipher Suites implemented by the SecY with management control over their use (10.7.25).
m) Allows selection of the current Cipher Suite, from those implemented.
n) Supports installation of SAKs for the current Cipher Suite, for transmission, reception, or both.

Figure 10-5 illustrates the management information that represents a SecY'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 Fowler [B1]. The containment relationships in Figure 10-5 have been chosen primarily to reflect the necessary relationships between lifetimes of potentially transient objects. For example, a receive SC can contain a succession of SAs, but never more than one per AN at a time, and all receive SAs for an SC are deleted when the receive SC ceases to exist. A paradigm of object creation and deletion is used, instead of one of data structure reuse, to express the required bounding of the lifetime of key information.

NOTE 1—Figure 10-5 omits parameters specific to extended packet numbering [used by some but not all Cipher Suites (14.7, 14.8)] and not accessible by network management. Specifically: 1) the createReceiveSA(), ReceiveSA(), createTransmitSA(), and TransmitSA() procedures all take an additional SSCI parameter, whose value becomes a parameter of the created SA; 2) the install_key() procedure takes an additional Salt parameter, whose value becomes an inaccessible parameter of the Data_key object. These parameters are specified in 10.7.13, 10.7.22, and 10.7.28.

In Figure 10-5 the management information for each SecY is 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-2010) 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, with a SecY System corresponding 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).

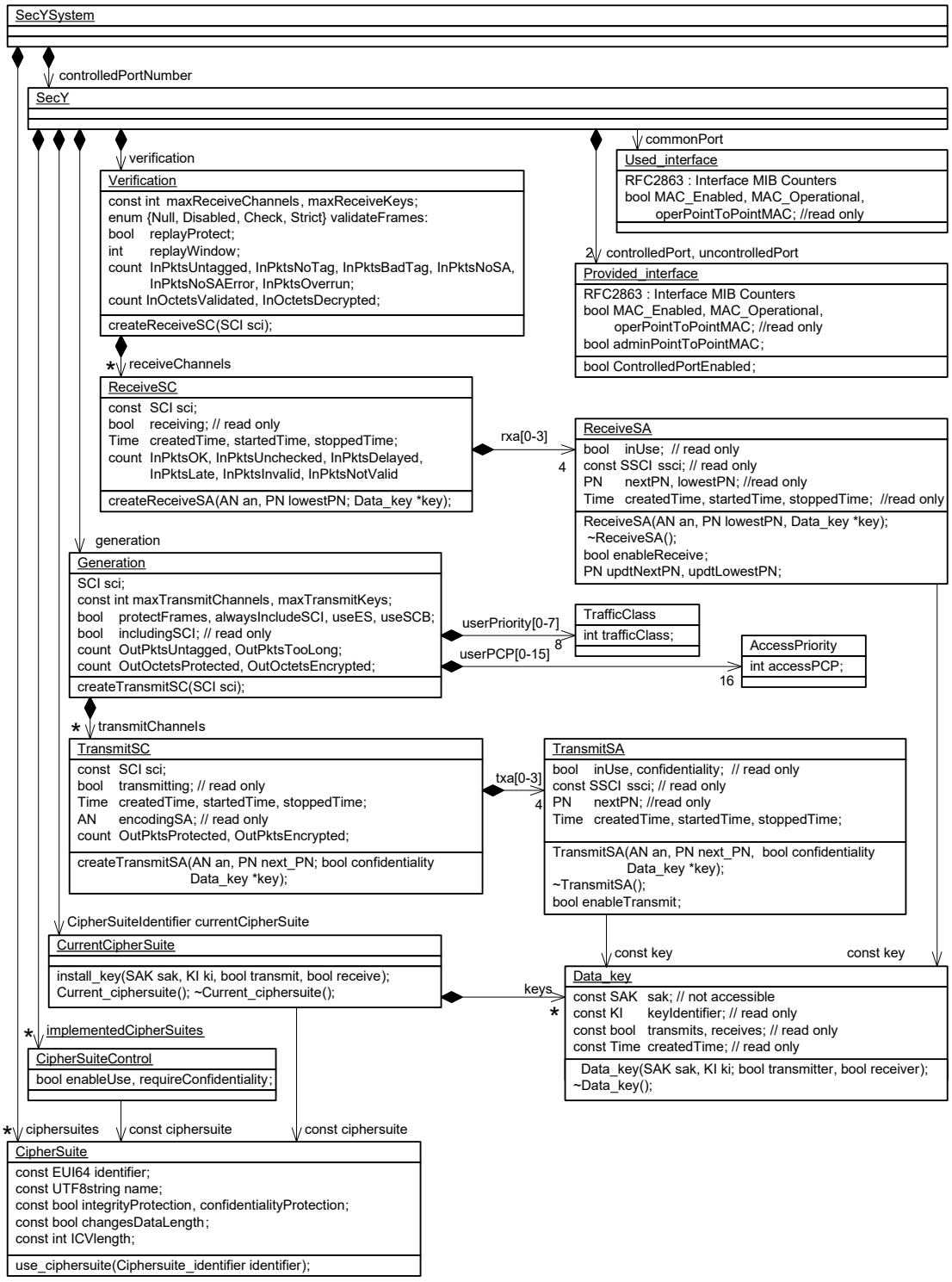


Figure 10-5—SecY managed objects

If a Bridge Port is supported by a SecY (11.3) the ifIndex value used to identify the SecY's Controlled Port will be that identifying the ISS interface (service access point) used by the Bridge Port. IEEE Std 802.1Q specifies Bridge Port Numbers that identify Bridge Ports from the point of view of a bridge's MAC Relay Entity, and port numbers in general to identify ISS interfaces. In simple, common, cases (11.3) each Bridge Port Number can and most likely will be the same as the port number (and ifIndex value) identifying the Controlled Port, though an optional mapping table is specified (12.5.1 of IEEE Std 802.1Q-2018).

IEEE Std 802.1Q can constrain the relationship between Bridge Port Numbers and other bridging parameters (see, for example, 12.13 of IEEE Std 802.1Q-2018) and if RSTP or MSTP are implemented the maximum number of Bridge Ports is 4095 (17.3.2.2 of IEEE Std 802.1Q-2018). In a system comprising multiple bridge components, each port is uniquely identified by a ComponentID and Port Number pair. The SCI values used by a SecYs supporting Bridge Ports do not have to be derived from the Bridge Port Numbers or (possibly different) controlledPortNumbers so do not further constrain those port numbers. However, the least significant 12 bits (if a SecY supports multiple traffic class SCs) and all 16 bits (otherwise) of the Port Identifier can be assigned—subject only to the requirement for SCI uniqueness (8.2.1), so that in the simple case of a bridge component with 4095 or fewer ports, each SCI's Port Identifier can convey the Bridge Port Number and use the Bridge Address for the MAC Address-based component of each SCI, if so desired.

NOTE 2—IEEE Std 802.1AEcg-2017 added the SecY System to Figure 10-5 and clarified the management use of port numbers and ifIndex values, but did not change any related normative provisions.

Conformance to this standard is strictly in terms of the external behavior required by this standard, as revealed through the relationship of the operation of the SecY to the operations supported by the SMIv2 MIB module (Clause 13) and to the specifications of protocols operated by the KaY. Interactions with the KaY through the LMI are wholly contained within the secure system, and there is no conformance with respect to syntactic elements that are used to describe that interface in this clause. Table 10-3 specifies performance requirements for SecY operation, including maximum delays for the execution of management operations.

In some situations it can be desirable to substitute control using SNMP for the operation of key agreement protocols, and Clause 13 provides all the necessary operations as an option. However, misuse of these operations can compromise security, and their availability (including the ability of an administrator to configure access to these operations) may be forbidden in some systems.

10.7.1 SCI

The SCI for the SecY's default traffic class (7.1.2, 8.2.1) can be read but not written by management.

If the SecY supports more than one transmit SC [5.4(e), 10.7.1, 10.7.17], the four most significant bits of the Port Identifier component of this SCI are zero.

10.7.2 Uncontrolled Port status

The following status parameters are provided to the user(s) of the Uncontrolled Port, including the KaY:

a) MAC_Enabled
b) MAC_Operational
c) operPointToPointMAC

Their values are identical to those for the Common Port. They can be read but not written by management.

10.7.3 Uncontrolled Port statistics

The following statistics are provided to support IETF RFC 2863 interface MIB Counters:

a) ifInOctets
b) ifInUcastPkts, ifInMulticastPkts, and ifInBroadcastPkts
c) ifInDiscards
d) ifInErrors
e) ifOutOctets
f) ifOutUcastPkts, ifOutMulticastPkts, and ifOutBroadcastPkts
g) ifOutErrors

The ifInOctets, ifInUcastPkts, ifInMulticastPkts, and ifInBroadcastPkts counts are identical to those of Common Port and are not separately recorded. The ifInDiscards and ifInErrors counts are zero, as the operation of the Uncontrolled Port provides no error checking or occasion to discard packets, beyond that provided by its users or by the entity supporting the Common Port.

The ifOutErrorscount is zero, as no checking is applied to frames transmitted by the Uncontrolled Port.The ifOutOctets, ifOutUcastPkts, ifOutMulticastPkts, and ifOutBroadcastPkts counts are the same as those for the user of the Uncontrolled Port.

10.7.4 Controlled Port status

The following status parameters are provided to the user of the Controlled Port, and can be read but not directly written by management:

a) MAC_Enabled, True if and only if
   1) ControlledPortEnabled (10.7.5) is True, and
   2) MAC_Enabled is True for the Common Port, and
   3) transmitting (10.7.21) is True for the transmit SC, and
   4) receiving (10.7.12) is True for at least one receive SC.
b) MAC_Operational, True if and only if
   1) MAC_Enabled is True, and
   2) MAC_Operational is True for the Common Port.
c) operPointToPointMAC. If adminPointToPointMAC is Auto (6.5), operPointToPointMAC is True if and only if
   1) validateFrames (10.7.8) is Strict, and receiving is enabled for receive SCs from at most one peer SecY, or
   2) validateFrames is not Strict, and operPointToPointMAC is True for the Common Port.

   Receive SCs are assumed to originate from the same peer SecY if their SCIs are the same with the exception of the four most significant bits of the Port Identifier component.

The following status parameter may be read and written by management:

d) adminPointToPointMAC (6.5)

   NOTE—Prior to IEEE Std 802.1AEcg-2017, each SecY used a single transmit SC. The adminPointToPointMAC variable can be used to configure operPointToPointMAC in the event that an earlier implementation of this standard does not recognize two receive SCs as being from the same SecY or configures two distinct SecYs (in the same CA) with SCIs that differ only in the most significant bits of the Port Identifier.

10.7.5 Controlled Port controls

The KaY uses the following parameter(s):

a) ControlledPortEnabled

By setting ControlledPortEnabled False, the KaY can prohibit use of the Controlled Port until the secure connectivity required has been configured.

10.7.6 Controlled Port statistics

The following statistics are provided to support IETF RFC 2863 interface MIB Counters:

a) ifInOctets
b) ifInUcastPkts, ifInMulticastPkts, and ifInBroadcastPkts
c) ifInDiscards
d) ifInErrors
e) ifOutOctets
f) ifOutUcastPkts, ifOutMulticastPkts, and ifOutBroadcastPkts
g) ifOutErrors

The ifInOctets count is the sum of all the octets of the MSDUs delivered to the user of the Controlled Port by the Secure Frame Verification process (10.6), plus the octets of the destination and source MAC addresses.

The ifInDiscards count is the sum of all the InPktsNoTag, InPktsLate, and InPktsOverrun counts. The ifInErrors count is the sum of all the InPktsBadTag, InPktsNoSA, and InPktsNotValid counts (10.6, Figure 10-4).

The ifOutOctets count is the sum of the all octets of the MSDUs delivered by the user of the Controlled Port to the Secure Frame Generation process (10.5), plus the octets of the destination and source MAC addresses.

The ifOutErrors count is equal to the OutPktsTooLong count (Figure 10-3). If ifOutDiscards is reported as part of IETF RFC 2863 counts, it is zero.

10.7.7 Frame verification capabilities

The SecY's frame verification capabilities are represented by the following parameters:

a) Maximum number of receive channels
b) Maximum number of keys in simultaneous use for reception

These parameters can be read but not written by management.

10.7.8 Frame verification controls

Frame verification is subject to the following controls, as specified in 10.6:

a) validateFrames, taking values of Null, Disabled, Check, or Strict, with a default of Strict
b) replayProtect, True or False, with a default of True
c) replayWindow, taking values between 0 and 2^{32}–1, with a default of 0

The validateFrames and replayProtect controls are provided to facilitate deployment. They can be read by management. Each may be written by management, but a conformant implementation shall provide a mechanism to allow write access by network management to be disabled for each parameter individually. If management access is prohibited to any of these parameters, its default value should be used.

If the Current Cipher Suite uses extended packet numbering, i.e., a 64-bit PN, the maximum value of replayWindow used in the Secure Frame Verification process (10.6) is 2^{30}–1, thus ensuring that the replayWindow does not encompass more than half of the range of PNs that can be correctly recovered (10.6.2). Any higher value set by network management is retained for possible subsequent use with a different Cipher Suite and will be reported if read by network management. This provision provides compatibility with prior revisions of this standard, though it is unlikely that such a high value of replayWindow would have been used.

10.7.9 Frame verification statistics

Any given received frame increments (10.6) exactly one of the following counts [item a) through item l)]. The following counts are maintained for the frame verification process as a whole:

a) InPktsUntagged
b) InPktsNoTag
c) InPktsBadTag
d) InPktsNoSA
e) InPktsNoSAError
f) InPktsOverrun

The following counts are maintained only for each receive SC and are discarded if the record of the SC is deleted by the KaY:

g) InPktsOK
h) InPktsUnchecked
i) InPktsInvalid
j) InPktsNotValid
k) InPktsDelayed
l) InPktsLate

The counts reported for each SC include those for current and prior SAs, with ANs that have since been reused. This allows useful counts to be maintained on high-speed LANs where an SA may be used for little more than 5 min, and an AN reused after 20 min.The times at which each SC and SA were, or are, in use are recorded (10.7.12, 10.7.14) and assist correlation of the statistics collected with network events.

10.7.10 Frame validation statistics

Investigation or validation of the performance of the cryptographic functions is supported by maintaining counts of packets (InPktsOverrun, 10.6.3, 10.7.9) that have been discarded due to inability to validate frames at the received rate, and by accumulation of the following counts:

a) InOctetsValidated, the number of octets of User Data recovered from received frames that were integrity protected but not encrypted.
b) InOctetsDecrypted, the number of octets of User Data recovered from received frames that were both integrity protected and encrypted.

These counts are incremented even if the User Data recovered failed the integrity check or could not be recovered. In the latter case, an estimate of the number of User Data octets is used, as judged by the load imposed on the validation function.

10.7.11 Receive SC creation

A receive SC, with a given SCI that remains unchanged for the life of the SC, is created following a request from the KaY. Each SC has a unique SCI.

Receive SCs and SAs (10.7.13) may also be created and controlled by management, but a conformant implementation shall provide a mechanism to allow creation and setting of control parameters by network management to be disabled.

10.7.12 Receive SC status

The following status parameters can be read, but not written, by management:

a) receiving, True if inUse (10.7.14) is True for any of the SAs for the SC, and False otherwise
b) createdTime, the system time when the SC was created
c) startedTime, the system time when receiving last became True for the SC
d) stoppedTime, the system time when receiving last became False for the SC

When the SC is created, receiving is False, and startedTime and stoppedTime are equal to createdTime.

The record of the SC should be retained after it is no longer used, subject to the availability of system resources, to provide information about immediate past operation.

10.7.13 Receive SA creation

A receive SA is created for an existing SC on request from the KaY, with the following parameters:

a) The association number, AN, for the SA
b) nextPN (10.6, 10.6.5)
c) lowestPN, the lowest acceptable PN value for a received frame (10.6, 10.6.2, 10.6.4, 10.6.5)
d) A reference to an SAK that is unchanged for the life of the SA

and, if the Current Cipher Suite uses extended packet numbering (14.7, 14.8), the KaY also supplies the following parameter:

e) SSCI for the SA

   Each SA that uses the same SAK has a different SSCI when these Cipher Suites are used. When the SA is created, its SCI and SSCI are provided (for use in subsequent validation operations) to the instance of the Current Cipher Suite identified by the referenced SAK. A receive SA will not be created if the SSCI supplied duplicates that for a different SCI (for the same SAK, for transmission or reception).

Frame verification statistics (10.7.9) for the SA are set to zero when the SA is created. Any prior SA with the same AN for the SC is deleted. Creation of the SA fails unless the referenced SAK exists and is installed (i.e., is available for use). A management protocol dependent reference is associated with each SA. This reference allows each SA to be distinguished from any previously created for the same SCI and AN.

MKA, specified in IEEE Std 802.1X, does not distribute SSCIs explicitly. A KaY assigns SSCI values as follows. The KaY with numerically greatest SCI uses the SSCI value 0x00000001, the KaY with the next to the greatest SCI uses the SSCI value 0x00000002, and so on. This assignment procedure is not necessarily applicable to any other key agreement protocol.

NOTE—At any given time (when configured by a KaY using MKA as specified in IEEE Std 802.1X), this and other Cipher Suites (including those specified in 14.5, 14.6, and 14.7) use the same SAK for all SAs (each with a different SCI) within the same CA and with the same AN. MKA guarantees that each KaY that uses a given SAK has a unique SCI, and these SCIs are present in every MKPDU that conveys a (key-wrapped) SAK.The number of SCIs (and hence the number of SSCIs) is ultimately limited by the maximum number of current members in a group CA that MKA can support (less than 100) but is likely to be further limited by the port-based network control application (see Clause 7 of IEEE Std 802.1X-2010).

10.7.14 Receive SA status

The following parameters can be read, but not directly written, by management:

a) inUse
b) nextPN (10.6, 10.6.5)
c) lowestPN, the lowest acceptable PN value for a received frame (10.6, 10.6.2, 10.6.4, 10.6.5)
d) createdTime, the system time when the SA was created
e) startedTime, the system time when inUse last became True for the SA
f) stoppedTime, the system time when inUse last became False for the SA
g) keyIdentifier (10.7.28), identifying the SAK used by the SA

and, if the Current Cipher Suite uses extended packet numbering (14.7, 14.8), the following parameter:

h) ssci, the SSCI for this receive SA

If inUse is True, and MAC_Operational is True for the Common Port, the SA can receive frames.

The keyIdentifier 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.

10.7.15 Receive SA control

The KaY uses the following parameters to control the use of each receive SA:

a) enableReceive
b) updtNextPN
c) updtLowestPN

When the SA is created, enableReceive and inUse are False and the SA cannot be used to receive frames. The SA shall be able to receive, and inUse shall be True, when enableReceive is set. The SA shall stop receiving, and inUse shall be False, when enableReceive is reset.

The value of nextPN (or lowestPN as appropriate) shall be set to the greater of its existing value and the supplied of updtNextPN (or updtLowestPN). Initially, following creation, the values of nextPN and lowestPN will have been set to the values supplied by KaY.

10.7.16 Frame generation capabilities

The SecY’s frame generation capabilities are represented by the following parameter(s):

a) Maximum number of transmit channels
b) Maximum number of keys in simultaneous use for transmission

These parameters can be read but not written by management.

NOTE—An individual SecY can support multiple traffic class SCs (10.7.17). When MKA is used (see Annex E), an SAK distributed by the Key Server is used by all newly created SAs (each supporting one of the SCs in the CA) so a SecY need only support two keys for transmission and reception at a time (allowing for rollover without frame loss, from one SAK to its successor), irrespective of the number of its traffic class SCs and peers in the CA.

10.7.17 Frame generation controls

Frame generation is subject to the following controls:

a) protectFrames (10.5), True or False, with a default of True
b) alwaysIncludeSCI (10.5.3), True or False, with a default of False
c) useES (10.5.3), True or False, with a default of False
d) useSCB (10.5.3), True or False, with a default of False

The protectFrames control is provided to facilitate deployment. The protectFrames, alwaysIncludeSCI, useES, and useSCB controls can be read by management and may be written, but a conformant implementation shall provide a mechanism to allow write access by network management to be disabled. If management access is prohibited, the default or a value determined by the KaY should be used.

The following status parameter can be read, but not written, by management:

e) includingSCI (10.5.3), True if and only if the SC bit is set and the SCI explicitly encoded in each SecTAG transmitted

The SecY may map each frame to a transmit SC using a Traffic Class Table and the frame's user priority. Up to eight transmit SCs may be implemented, allowing separate transmit SCs for each possible user priority. However, the reason for the possible use of multiple transmit SCs is to take advantage of the fact that their separate SAs use different PN values and thus to minimize the size of the replayWindow, and in particular to facilitate strict reception ordering and replay protection when the Common Port is supported by a service (such as a Provider Bridged Network, see 11.7) that can reorder frames of different priority. In such cases, the useful number of traffic classes might be two or three, corresponding to the differentiated classes of service provided. While the Traffic Class Table mirrors that specified by IEEE Std 802.1Q for the management of bridge queues, a SecY has a minimal implementation dependent buffering requirement and there is no reason to suppose that any given implementation might provide more timely service if the Common Port does not provide priority differentiated services.

NOTE 1—IEEE Std 802.1AEcg-2017, introducing the use of multiple transmit SCs, was developed contemporaneously with IEEE Std 802.3br™-2016 [B5], which added a capability that allows a high priority Ethernet frame to preempt one of lower priority and thus be received in its entirety prior to the latter. This provides another example of a service that can reorder frames on the basis of priority and for which the use of a separate transmit SC with separate PN number spaces can be used to allow strict ordering and strict replay protection for preemptible and preempting frames separately.

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 SecY may map the user priority of each frame's transmit request at the Controlled Port to the access priority to be used for the corresponding transmit request at the Common Port using the Access Priority Table. The table index and its output both comprise 4 bits, representing both the priority (most significant three bits) and drop_eligible (least significant bit) of the user priority and access priority. The default value of each table entry is that of its index, thus leaving the priority and drop_eligible bits unchanged. This default is appropriate if the service provided by the Common Port already implements its own mapping from requested priority to its own priority or other parameters used to make decisions that affect frame reordering, and that mapping matches the Traffic Class Table's mapping of user priority to transmit SC. The default is also appropriate if the administrator is willing to tolerate the degree of misordering, and the replayWindow

size that implies, resulting from allocating frames of different access priority to the same SC in the interest of providing a differentiated service to the higher priority frames without using additional transmit SCs. Otherwise, it is recommended that the Access Priority Table be configured so that frames allocated to the same transmit SC use the same access priority.

NOTE 2—Where MACsec is used to support an instance of the ISS that in turn supports the EISS, the priority originally requested by the EISS user is encoded in the VLAN tag within the ISS MSDU and is thus protected by MACsec and is communicated unchanged to the peer EISS user, unaffected by local access priority mapping decisions.

10.7.18 Frame generation statistics

Any given transmitted frame (10.5) increments exactly one of the following counts [item a) through item d)]. The following counts are maintained for the frame generation process as a whole:

a) OutPktsUntagged
b) OutPktsTooLong

The following counts are maintained for each transmit SC:

c) OutPktsProtected
d) OutPktsEncrypted

The counts reported for each SC include those for current and prior SAs, with ANs that have since been reused. This allows useful counts to be maintained on high-speed LANs where an SA may be used for little more than 5 min, and an AN reused after 20 min.The times at which each SC and SA were, or are, in use are recorded (10.7.21, 10.7.23) and assist correlation of the statistics collected with network events.

NOTE—The OutPktsProtected and OutPktsEncrypted counts can be correctly reported, without the need for each frame to increment separate real-time counters. The packets for a given SA are either all encrypted (confidentiality protected) or all only integrity protected, so the counts for active SAs can be derived from the nextPN values (less any contribution to OutPktsTooLong made after PN assignment to discarded frames) and summed with that those previously accumulated for the SC. When an SA is replaced by a successor with the same AN, its counts are added to those accumulated for the SC.

10.7.19 Frame protection statistics

Investigation or validation of the performance of the cryptographic functions is supported by accumulation of the following counts:

a) OutOctetsProtected, the number of octets of User Data in transmitted frames that were integrity protected but not encrypted.
b) OutOctetsEncrypted, the number of octets of User Data in transmitted frames that were both integrity protected and encrypted.

10.7.20 Transmit SC creation

A transmit SC, with a given SCI that remains unchanged for the life of the SC, is created, as requested by the KaY, for the default traffic class SC and for each of the other SCs identified by the Traffic Class Table (if implemented). The KaY is responsible for ensuring the uniqueness of the SCI of any SC in a CA that might use the same SAK.

Transmit SCs and SAs (10.7.22) may also be created and controlled by management, but a conformant implementation shall provide a mechanism to allow creation and setting of control parameters by network management to be disabled.

10.7.21 Transmit SC status

The following status parameters can be read, but not directly written, by management:

a) transmitting, True if inUse (10.7.23) is True for any of the SAs for the SC, and False otherwise
b) encodingSA (10.5.1)
c) createdTime, the system time when the SC was created
d) startedTime, the system time when transmitting last became True for the SC
e) stoppedTime, the system time when transmitting last became False for the SC

When the SC is created, transmitting is False and startedTime and stoppedTime are equal to createdTime.

10.7.22 Transmit SA creation

An SA is created for a transmit SC on request from the KaY, with the following parameters:

a) AN, the association number for the SA
b) nextPN, the initial value of Transmit PN (10.5.2) for the SA
c) confidentiality, True if the SA is to provide confidentiality as well as integrity for transmitted frames
d) A reference to an SAK that is unchanged for the life of the SA

and, if the Current Cipher Suite uses extended packet numbering (14.7, 14.8), the KaY also supplies the following parameter:

e) SSCI for the SA

   Each SA that uses the same SAK has a different SSCI when these Cipher Suites are used. When the SA is created, its SCI and SSCI are provided (for use in subsequent protection operations) to the instance of the Current Cipher Suite identified by the referenced SAK. A transmit SA will not be created if the SSCI supplied duplicates that for a different SCI (for the same SAK, for transmission or reception).

Frame generation statistics (10.7.18) for the SA are set to zero when the SA is created. Any prior SA with the same AN is deleted. Creation of the SA fails unless the referenced SAK exists and is installed (i.e., is available for use). A management protocol dependent reference is associated with each SA. This reference allows the transmit SA to be distinguished from any previously created with the same AN.

MKA, specified in IEEE Std 802.1X, does not distribute SSCIs explicitly. A KaY assigns SSCI values as specified in 10.7.13.

10.7.23 Transmit SA status

The following parameters can be read, but not directly written, by management:

a) inUse
b) createdTime, the system time when the SA was created
c) startedTime, the system time when inUse last became True for the SA
d) stoppedTime, the system time when inUse last became False for the SA
e) nextPN (10.5, 10.5.2)
f) confidentiality, True if the SA is providing confidentiality as well as integrity for transmitted frames
g) keyIdentifier (10.7.28), identifying the SAK used by the SA

and, if the Current Cipher Suite uses extended packet numbering (14.7, 14.8), the following parameter:

h) ssci, the SSCI for this transmit SA

If inUse is True, and MAC_Operational is True for the Common Port, the SA can transmit frames.

The keyIdentifier 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.

10.7.24 Transmit SA controls

The KaY uses the following parameters to control the use of each transmit SA:

a) enableTransmit

When the SA is created, enableTransmit and inUse are False, and the SA is not used to transmit frames. The SC parameter encodingSA shall be set to the value of the AN for the SA and inUse set True, when enableTransmit is set. The SA shall stop transmitting, and inUse reset, when enableTransmit is reset.

10.7.25 Implemented Cipher Suites

The following per Cipher Suite read-only capability information is provided by the system of which the SecY is a part:

a) Cipher Suite Identifier, a globally unique 64-bit (EUI-64) identifier
b) Cipher Suite Name, a human readable and displayable UTF-8 (IETF RFC 2279 [B6]) string
c) integrityProtection, True if integrity protection without confidentiality can be provided
d) confidentialityProtection, True if confidentiality with integrity protection can be provided
e) offsetConfidentiality, True if a selectable offset for confidentiality can be provided
f) changesDataLength, True if the data length is changed
g) ICVlength, number of octets in the ICV

The Cipher Suite Identifier and Cipher Suite Name are both assigned by the document that specifies use of the Cipher Suite with this standard. If the Cipher Suite provides integrityProtection and confidentialityProtection, the SecY shall be capable of receiving frames with either, as signaled by the E and C bits in the SecTAG.

The confidentialityProtection parameter shall be True if and only if the Cipher Suite implementation is capable of being configured so that, when confidentiality is selected, all the octets of the MSDU are integrity and confidentiality protected.

The offsetConfidentiality parameter shall be True if and only if the Cipher Suite implementation is capable of both integrityProtection and confidentialityProtection, and of being configured so that, when confidentiality is selected, a selectable number (0, 30, or 50) of the initial octets of the MSDU are only integrity protected, and appear in the MPDU immediately after the SecTAG in the order and with the values in the MSDU (Figure 8-1), while the remaining octets are confidentiality and integrity protected.

NOTE—IEEE Std 802.1AE-2006 specified the confidentiality offset option to facilitate early MACsec deployment on systems that needed to examine the initial octets of IP version 4 or version 6 frames to decide where to store received frames, before decrypting the frame. The XPN Cipher Suites do not support confidentiality offsets.

10.7.26 SecY Cipher Suite use

The Cipher Suite capabilities implemented for each SecY can be read by management. The following controls may be written by management, but a conformant implementation shall provide a mechanism to allow write access by network management to be disabled for each parameter individually:

a) enableUse, True if use of the Cipher Suite is permitted
b) requireConfidentiality, True if the Cipher Suite can only be used to provide both confidentiality and integrity (and not integrity only, or confidentiality with an offset)

The MKA Key Server selects the Cipher Suite to be used to protect communication within a CA. If enableUse 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 enableUse and requireConfidentiality are both True for the selected Cipher Suite, confidentiality protection is used.

NOTE—A system might contain distinct SecY implementations with differing detailed Cipher Suite capabilities. Each of the latter can be represented by a distinct set of Cipher Suite implementation capability information (10.7.25), with each SecY's capabilities represented by a list of references (each with separate use controls) to some of those sets.

10.7.27 Cipher Suite selection

The KaY uses the following parameter to select the Current Cipher Suite:

a) currentCipherSuite, the Cipher Suite Identifier (10.7.25) for the cipher suite

If offsetConfidentiality (10.7.25) is not False for the Cipher Suite, the following parameter is specified:

b) confidentialityOffset, the number of initial octets of each MSDU without confidentiality protection

The CurrentCipherSuite 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. The confidentialityOffset applies to all frames transmitted and received with confidentiality protection. If both confidentialityProtection and offsetConfidentiality are supported, then it takes the values 0, 30, and 50.

If the Current Cipher Suite is changed, all keys created for that Cipher Suite are deleted, and (as a consequence) inUse will become False for all SAs, with the further consequence that MAC_Operational will become False for the Controlled Port.

10.7.28 SAK creation

An SAK is installed, i.e., an instance of the Current Cipher Suite for a given SAK is created, on request from the KaY with the following parameters:

a) The SAK value
b) keyIdentifier, used by network management to reference the key
c) transmit, True if the key is to be installed for transmission
d) receive, True if the key is to be installed for reception

and, if the Current Cipher Suite uses extended packet numbering, the following parameter:

e) Salt (McGrew [B11]), a 96-bit parameter provided to the Current Cipher Suite for subsequent protection and validation operations

MKA does not include explicit parameters for distributing a Salt. Each KaY computes this parameter as follows. The 64 least significant bits of the Salt are the 64 least significant bits of the MKA Key Server's Member Identifier (MI), the 16 next most significant bits of the Salt comprise the exclusive-or of the 16 next most significant bits of that MI with the 16 most significant bits of the 32-bit MKA Key Number (KN), and the 16 most significant bits of the Salt comprise the exclusive-or of the 16 most significant bits of that MI with the 16 least significant bits of the KN. This way of obtaining a Salt is not necessarily applicable to any other key agreement protocol.

10.7.29 SAK status

The following parameters can be read, but not directly written, by management:

a) transmits, True if the key has been installed for transmission, i.e., can be used by a transmit SA
b) receives, True if the key has been installed for reception, i.e., can be used by a receive SA
c) createdTime, the system time when the SAK record was created

10.8 Addressing

Frames transmitted between end stations using the MAC Service carry the MAC Address of the source and destination peer end stations in the source and destination address fields of the frames, respectively. Communicating peer SecYs can secure communication for all or part of the path used by such frames, and are not directly addressed by the communicating peers, nor are the frames modified to include additional addresses. Each SecY does not have a MAC Address of its own, but is associated with a local entity that forms part of the secure system.

The addressing used by Key Agreement Entities and the means they use to identify SecYs within the same secure system are outside the scope of this specification.

While destination and source MAC addresses are not required to identify SecYs, they are parameters of the MAC Internal Sublayer Service (ISS) used and provided by a SecY, and are covered by the ICV, generated by a Cipher Suite implementation while remaining unencrypted. To facilitate ICV calculation and verification, all frames processed by SecYs use 48-bit MAC addresses.

10.9 Priority

While priority is a parameter of both an ISS M_UNITDATA.request and corresponding M_UNITDATA.indications, end-to-end communication of the requested priority is not a service attribute (6.1). Protocols supporting the ISS can use the requested priority to perform local actions in the originating station, and do not necessarily attempt to communicate the parameter. Accordingly, the requested and indicated priorities do not contribute to the ICV, and are not explicitly included in the encoded MSDU by a transmitting SecY.

NOTE—If communication of priority is desired, either guaranteed unchanged or available to a service provider for possible modification to meet the admission control and service characteristics of a particular network, use of the EISS in conjunction with the ISS is indicated. See Clause 7.

10.10 SecY performance requirements

Table 10-3 places requirements on SecY performance to ensure that MACsec operates correctly.

Table 10-3—SecY performance requirements

Parameter	Permitted values
SecY transmit delay	< Wire transmit time for maximum sized MPDU + (4 times wire transmit time for 64 octet MPDUs)
SecY transmit delay variance	< SecY transmit delay
SecY receive delay	< Wire transmit time for maximum sized MPDU + (4 times wire transmit time for 64 octet MPDUs)
SecY receive delay variance	< SecY receive delay
SC and SA creation and control delay	< 0.1 second
Transmit SAK install delay	< 1 second (8.2.2)
Transmit SAK switch delay	< Wire transmit time for 64 octet MPDU (8.2.2)
Receive SAK install delay	< 1 second
Receive SAK switch delay	No frame loss

All times are in seconds.

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™ [B3], IEEE Std 802.1Q).

11. MAC Security in systems

This clause specifies how MAC Security is incorporated within the architecture of

a) End stations (11.2)
b) MAC Bridges (11.3)
c) VLAN-aware Bridges (11.4)
d) Systems that incorporate Link Aggregation (11.5)
e) Systems that incorporate Link Layer Discovery Protocol (LLDP, 11.6)
f) Provider Bridges and VLAN-aware Bridges attached to Provider Bridged Networks (11.7)
g) LANs that provide independently secured access for multiple end stations (11.8).

The figures in this clause illustrate the relative position of components within the MAC Service interface stacks (11.1) of each of these systems. Both the secure MAC Service provided by the Controlled Port and the insecure service provided to the Uncontrolled Port are shown.

NOTE—For more information on the Controlled and Uncontrolled Ports and the operation of the SecY, see Clause 10.

11.1 MAC Service interface stacks

Each LAN MAC, e.g., that is specified in IEEE Std 802.3, is capable of providing the MAC Service directly to LLC and its clients, as illustrated in Figure 11-1.

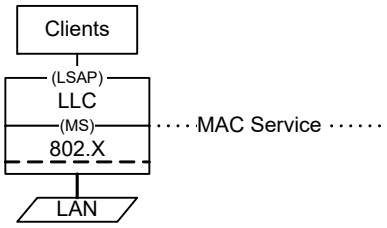


Figure 11-1—Direct support of the MAC Service by a media access method

NOTE 1—The term *802.X* refers to any one of the IEEE 802 LAN media access control method technologies.

Alternatively, media access method independent functions, such as VLAN tagging of frames (IEEE Std 802.1Q) and MAC Security (as specified by this standard), can be used to support the MAC Service (IEEE Std 802.1AC), the MAC Internal Sublayer Service (ISS, IEEE Std 802.1AC), or the Enhanced Internal Sublayer Service (EISS, IEEE Std 802.1Q). These functions use an ISS access point provided by media access method independent or media access method dependent convergence functions. See Figure 11-2.

Each SecY uses an ISS access point and provides the ISS at its Controlled and Uncontrolled Ports. This allows use of MAC Security with other media-independent functions. However, interoperability between systems using MAC Security requires not only interoperability between SecY implementations and use of the same LAN MAC technology, but also that the same, or compatible, media interface functions are used with the same relative position within the interface stack, as specified in this clause.

NOTE 2—MAC Bridges and VLAN-aware Bridges provide interoperability between access points for the MAC Service, the ISS, and the EISS, using the following common elements of those service specifications. The MAC Service, the ISS, and the EISS all use the same request and indication primitives. The parameters used by the ISS for each primitive are a superset of those of the MAC Service. An EISS access point effectively provides access to multiple ISS instances.

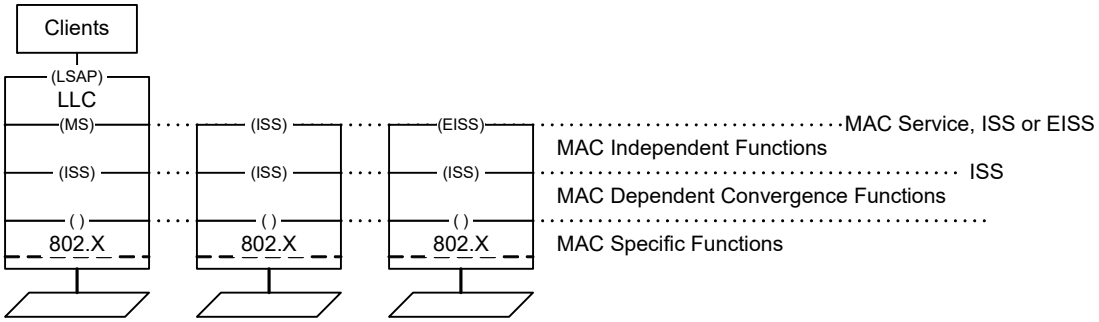


Figure 11-2—Provision of MAC Service with media-independent functions

11.2 MACsec in end stations

The ISS provided by the SecY is trivially mapped to and from the MAC Service provided within an end station. Service indications for unwanted destination MAC addresses are discarded, and the source MAC address of service requests is that of the station. Figure 11-3 shows MAC Security as the sole media-independent function within a station.

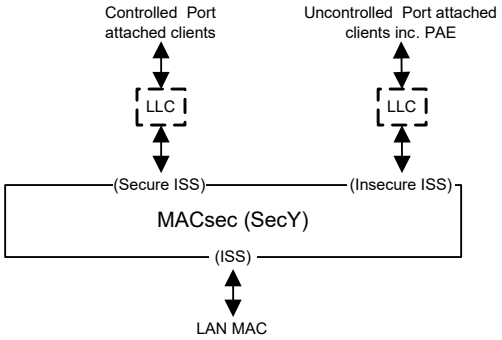


Figure 11-3—MACsec in an end station

11.3 MACsec in MAC Bridges

MAC Bridges are specified in IEEE Std 802.1Q. The MAC Relay Entity forwards frames between the ISS access points supported by each of the Bridge Ports. To provide MAC Security for such a system, each of the insecure interfaces presented by a LAN supports MACsec, which in turn supports the functions described in 8.5 of IEEE Std 802.1Q-2018. Figure 11-4 shows a bridge with and without MACsec.

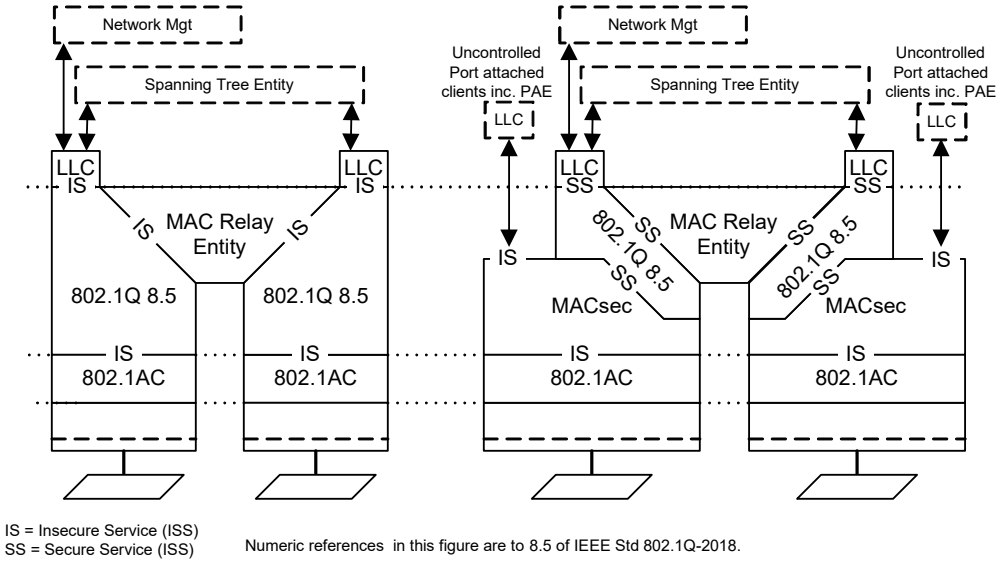


Figure 11-4—MACsec in a VLAN-unaware MAC Bridge

NOTE—If the MAC Bridge aggregates multiple LANs to support a single Bridge Port, each individual LAN supports its own SecY, which provides the secure MAC Service to the Link Aggregation sublayer, as specified in 11.5. Each aggregated port then provides secure service to the Bridge Port transmit and receive functions.

Figure 11-5 shows the interface stack for each of the Bridge Ports.

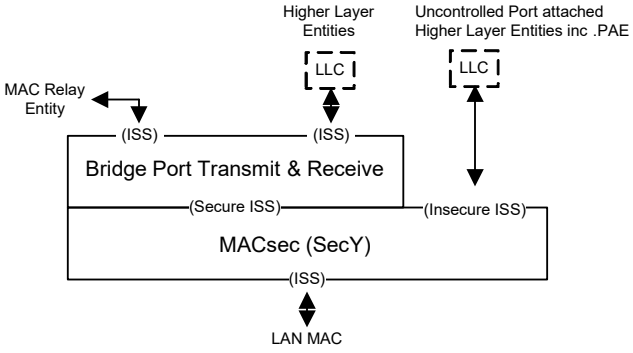
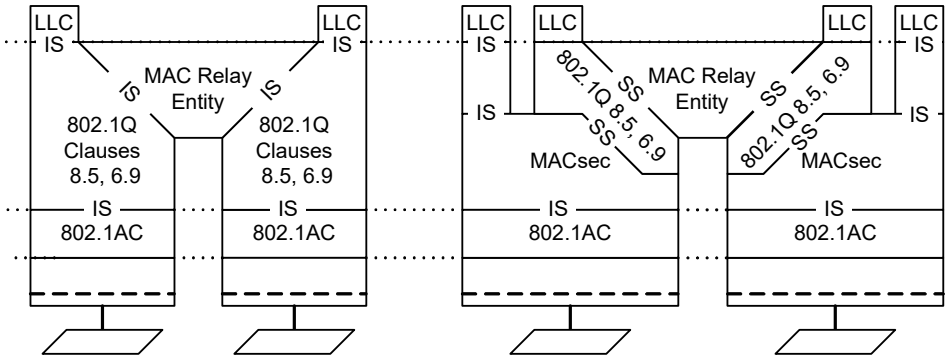


Figure 11-5— VLAN-unaware MAC Bridge Port with MACsec

11.4 MACsec in VLAN-aware Bridges

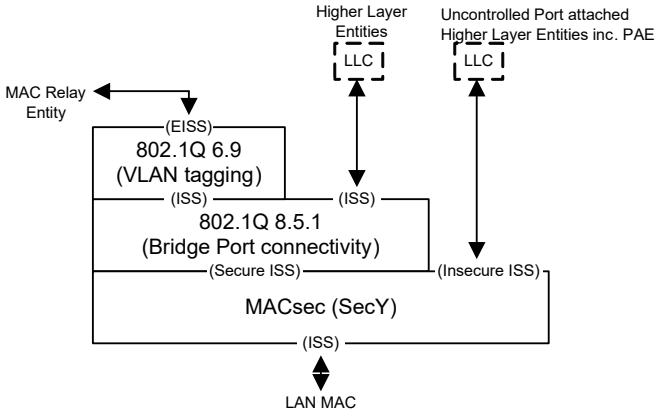
VLAN-aware Bridges are specified in IEEE Std 802.1Q. Figure 11-6 illustrates the addition of MAC Security.



Numeric references in this figure are to 8.5 and 6.9 of IEEE Std 802.1Q-2018.

Figure 11-6—Addition of MAC Security to a VLAN-aware MAC Bridge

Figure 11-7 shows the interface stack for each of the VLAN-aware Bridge Ports.



Numeric references in this figure are to 8.5.1 and 6.9 of IEEE Std 802.1Q-2018.

Figure 11-7— IEEE 802.1Q VLAN-aware Bridge Port with MACsec

Figure 6-2 shows the frame format and placement of the VLAN tag within the frame relative to MACsec. Thus if there is encryption, the VLAN tag is not in the clear.

NOTE—If the use of a protocol analyzer and other monitoring tools based on capture and analysis of packets on the wire is desired, integrity protection only, without confidentiality, should be used.

The position of MACsec, below both the Bridge Port connectivity and VLAN tagging functions, has the following consequences:

a) Each Bridge Port uses a single SecY, with a single transmit SC and a single receive SC for each of the other bridges and stations attached to the LAN, to support all VLANs.

b) Interoperability with MAC Bridges, that are not VLAN-aware, is supported in the same way as VLAN-aware and unaware bridges without MAC Security.

c) Higher-layer entities attached to the Bridge Port, such as the Spanning Tree Protocol Entity and protocol stacks for network management, do not need to be supported by separate SecYs. In particular a MACsec protected point-to-point link between two bridges continues to function as a point-to-point link despite the end station functions associated with each Bridge Port.

d) Changes in the operation of MAC Security do not cause differences in the network connectivity used by the MAC Relay Entity and in the network connectivity perceived by the Controlled Port attached higher-layer entities that execute control protocols for the relay function.

11.5 MACsec and Link Aggregation

Link Aggregation is specified in IEEE Std 802.1AX [B4]. 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. If the two links are being aggregated dynamically, as provided for by the Link Aggregation Control Protocol (LACP), the operation of LACP will be protected. In addition, if the authentication provided by the KaYs determines that the two links do not connect to the same partner system, local system management can change the aggregation keys. Changes in link aggregation do not cause changes to the MACsec CAs, SCs, SAs, or SAKs.

NOTE 1—LACP aggregation keys have nothing to do with cryptography. See IEEE Std 802.1AX [B4] for details.

Figure 11-8 shows part of an interface stack with MAC Security and Link Aggregation. The insecure service access points for each of the SecYs are independently provided to the KaY associated with each SecY, and may or may not be aggregated separately.

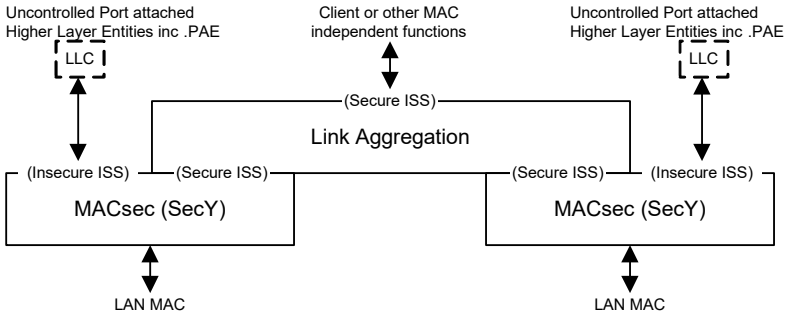


Figure 11-8—MACsec and Link Aggregation in an interface stack

Figure 11-9 shows the addition of link aggregation to the interface stack for a VLAN-aware Bridge Port that also uses MACsec.

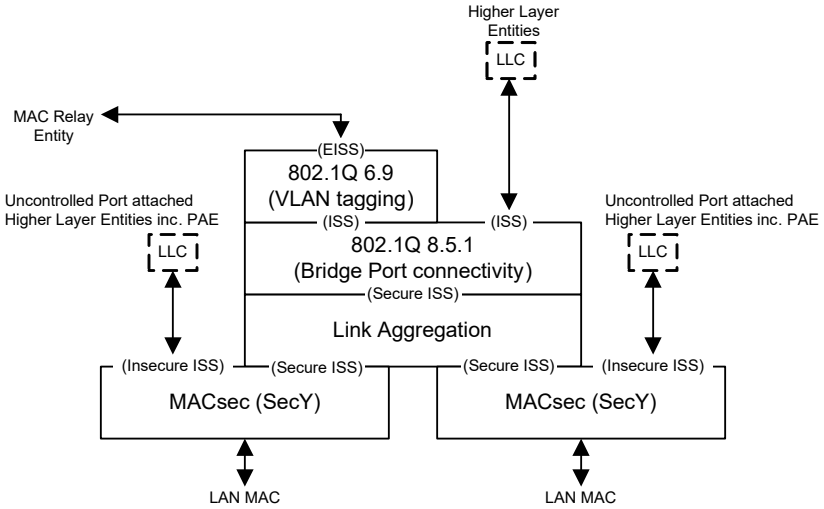


Figure 11-9—IEEE 802.1Q VLAN-aware Bridge Port with MACsec and Link Aggregation

11.6 Link Layer Discovery Protocol (LLDP)

LLDP is specified in IEEE Std 802.1AB. When used in conjunction with MACsec each LLDP Agent should make use of the Secure ISS provided by MACsec for the attached LAN as shown in Figure 11-10.

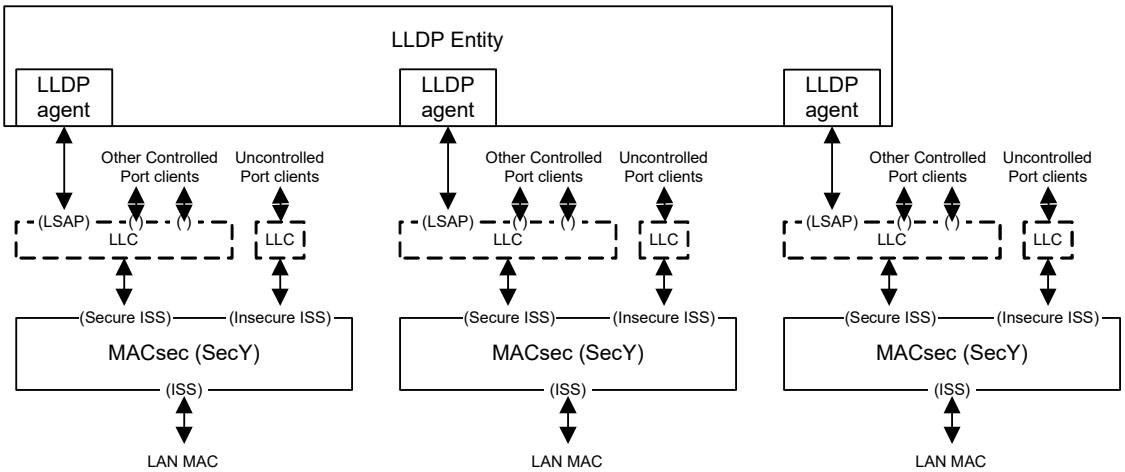


Figure 11-10—MACsec with LLDP

11.7 MACsec in Provider Bridged Networks

Provider Bridges (IEEE Std 802.1Q) enable service providers to use VLANs to offer the equivalent of separate LANs to different users. Data for each of the virtual LANs is segregated within the provider's network by using a Service VLAN TAG (S-TAG) that is distinguished, by EtherType, from the Customer VLAN-TAGs (C-TAGs) used within each customer's network. See Figure 11-11.

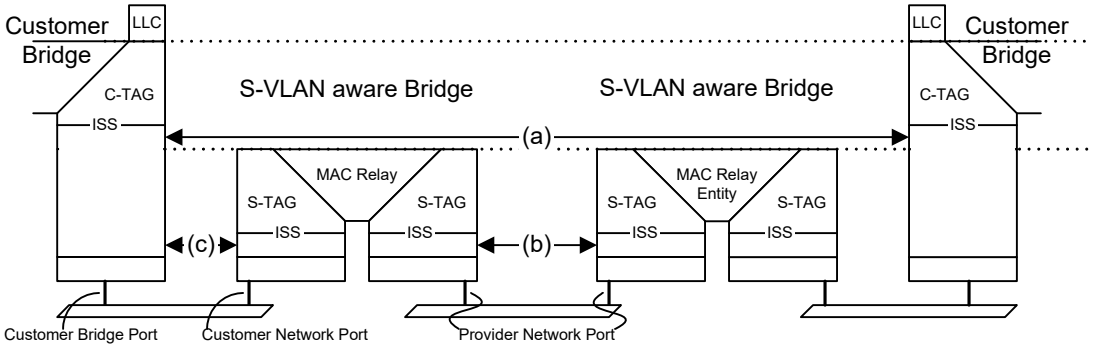


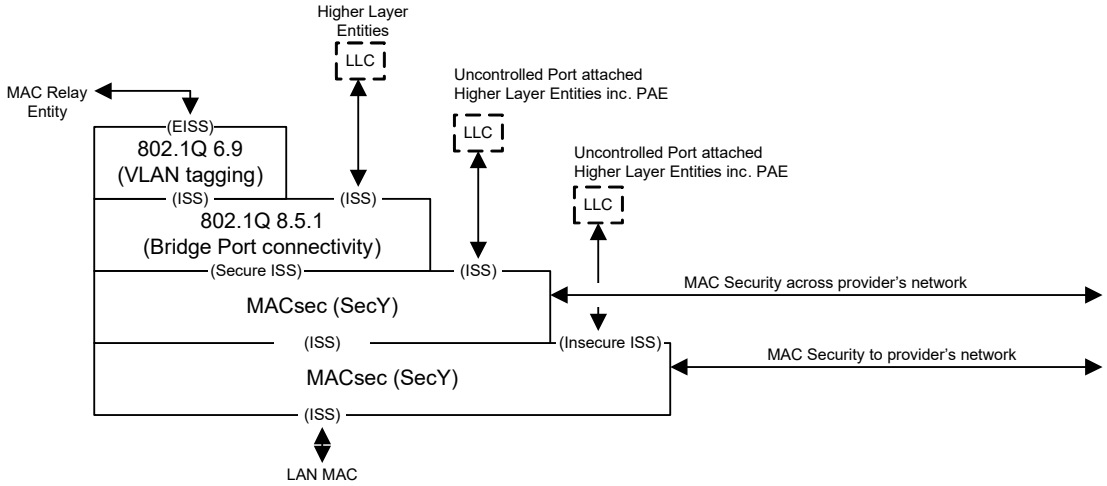
Figure 11-11—Internal organization of the MAC sublayer in a Provider Bridged Network

NOTE—Figure 11-11 is based on Figure 15-1 of IEEE Std 802.1Q-2018.

MACsec can be used to secure communication between

a) A customer's bridges or other equipment, across the provider's network
b) Adjacent S-VLAN aware Bridges, within the provider's network
c) A customer's bridge and the provider's network.

If it is the customer's intention to secure only one of item a) or item c), then the use of one of the interface stacks illustrated in Figure 11-3 (for an end station), Figure 11-5 (for a MAC Bridge), or Figure 11-7 (for a VLAN-aware Bridge) within the customer equipment is sufficient.



Numeric references in this figure are to 8.5.1 and 6.9 of IEEE Std 802.1Q-2018.

Figure 11-12—Interface stack for MAC Security to and across provider's network

Use of the interface stack illustrated in Figure 11-7 within the provider's S-VLAN aware Bridge Ports is sufficient to secure either item b) or item c) as required. If item c) is not to be secured, MACsec is either omitted from the interface stack for the Customer Network Port (see Figure 11-11), or the Bridge Port connectivity function (8.5.1 of IEEE Std 802.1Q-2018) uses the service provided by the Uncontrolled Port.

If it is the intention to secure both item a) and item c) from the Customer Bridge Port, then the use of two independent SecY's within the port's interface stack is required as shown in Figure 11-12.

Figure 11-13 shows the addition of the service access priority selection function described in 6.13 of IEEE Std 802.1Q-2018 to the interface stack of Figure 11-12, together with the use of Link Aggregation to support attachment to the provider's network with two LANs.

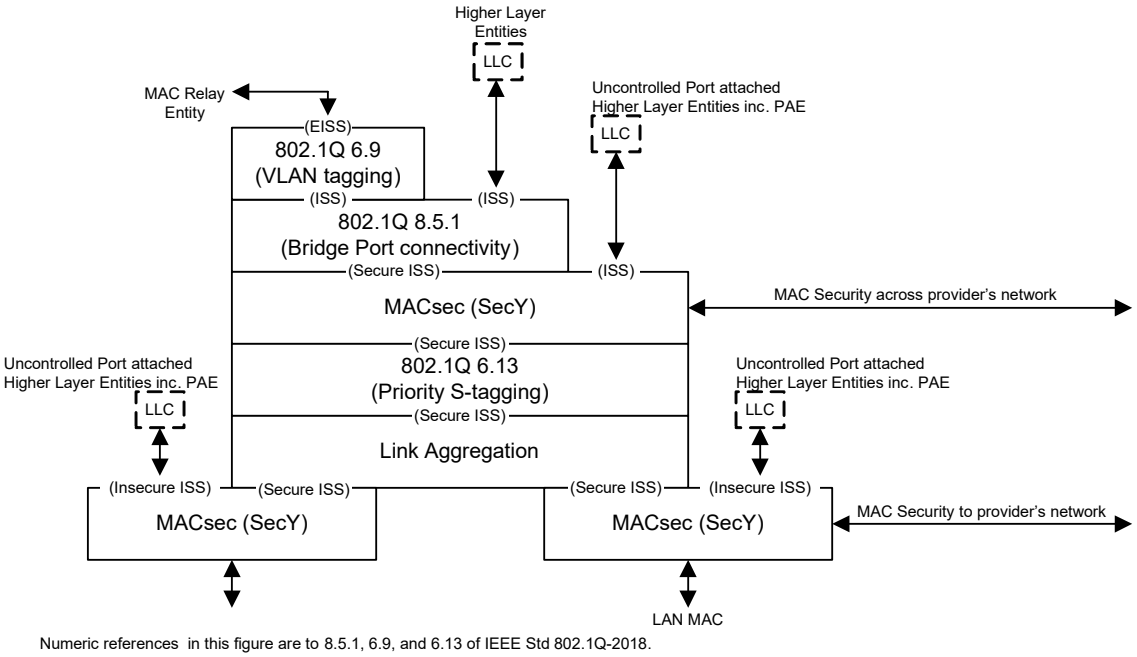


Figure 11-13—Provider network with priority selection and aggregation

11.8 MACsec and multi-access LANs

MACsec can be used to support the equivalent of multiple LANs from one station to each of a number of others using the service provided by a single LAN. Each station that connects to more than one of the multiple LANs does so by using a distinct SecY for each of those connections. MACsec frames for each of the multiple LANs are distinguished from frames for the others by the SCI of the originating SecY. If a station has more than one SecY, the SCIs for each SecY's transmit SC or SCs are based on the MAC Address allocated to that station but use a different Port Identifier component (9.9). Figure 11-14 shows one station (A in the figure) with two connections, one to each of two others (B, C).

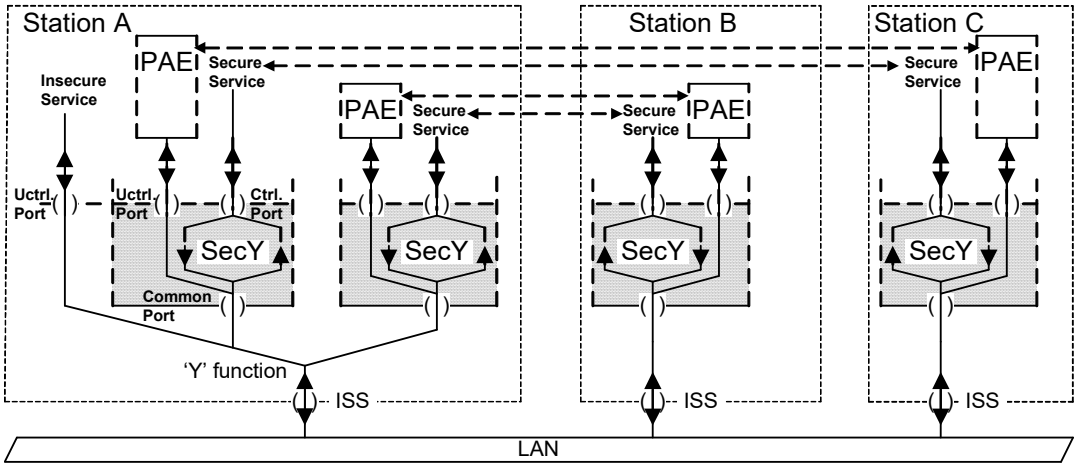


Figure 11-14—An example multi-access LAN

Frames transmitted by each SecY's Uncontrolled Port can include a SecTAG, with an SCI value used by the SecY's Controlled Port. These frames are distinguished by setting the E bit in the SecTAG TCI True and the C bit False, and are discarded by the frame verification process for the Controlled Port (10.6). The connectivity between Uncontrolled Ports using the SecTAG thus matches the secure connectivity provided between the corresponding Controlled Ports. The protocol entities attached to the SecY's Uncontrolled Port add and remove this SecTAG as required.

NOTE—Frames including a SecTAG and E bit True and C bit False were not used by any standard protocol at the time of the development of IEEE Std 802.1AEcg-2017, but this normative provision remains for possible future use by protocols that need to associate Uncontrolled Port frames with individual SCIs.

Frames transmitted through a SecY's Uncontrolled Port to a multi-access LAN can omit the SecTAG, provided that only one bi-directional unicast communication is supported between any pair of stations. The recipient uses the source address of the frame to identify the peer SecY.

Each multi-access capable station also supports an Uncontrolled Port (shown to the left in station A in Figure 11-14) that allows arbitrary frames to be transmitted on the LAN and received, if they are not MACsec frames, by any of the systems. These Uncontrolled Ports support the protocols required to discover peer multi-access capable systems, and to associate SCIs (and hence SecYs and KaYs) with each connection. The entities that operate such discovery and association protocols in stations, such as station A, that are capable of supporting multiple SecYs on a single LAN, are typically capable of instantiating some number of SecYs and associated entities on demand. The Controlled Ports thus provided to higher-layer entities can be transient, and are referred to as "virtual Ports".

Where a protocol entity for each SecY's Uncontrolled Port transmits frames without a SecTAG, it is possible for there to be no externally observable difference between the operation of entities attached to those ports and of an equivalent entity or entities attached to the Uncontrolled Port for the station as a whole. Whether to emphasize common functions or peer relationships is a choice for each protocol's specification.

Figure 11-15 shows part of an interface stack for a multi-access capable system. The 'Y' function can simply copy all indications from its lower service access point to all upper access points, and any request from an upper service access point to the lower access point. Each KaY and SecY will discard indications for SCIs that do not match one of their receive SCs. Alternatively, the 'Y' function can selectively deliver indications for known SCIs to the appropriate SecY, as instructed by the higher-layer entity responsible for virtual port creation and its association. Its detailed specification is determined by the specification of that entity.

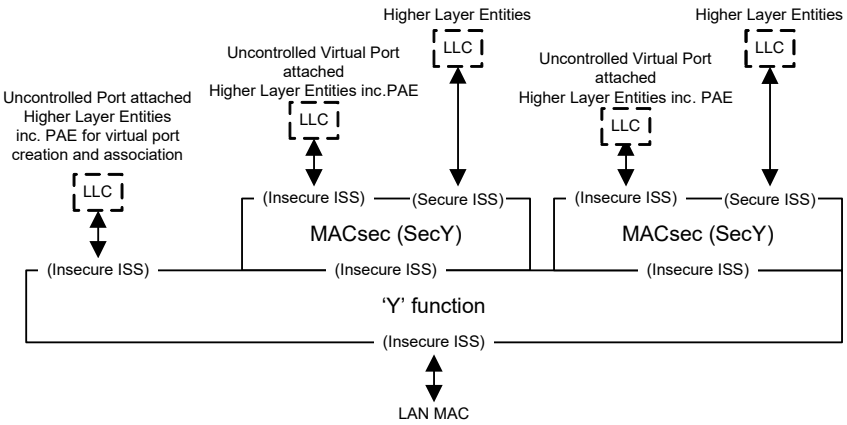


Figure 11-15—Multi-access LAN interface stack

The connectivity provided by a multi-access LAN depends on the security provided and can change as security is deployed, enabled, or disabled. Because this can lead to difficulties in the management of bridged networks, multi-access LANs should not be used to support LANs with two or more attached bridges. They are appropriate for the attachment of end stations or hosts at the periphery of the network.

12. MACsec and EPON

Clause 64 and Clause 65 of IEEE Std 802.3-2018 specify an Ethernet passive optical network (EPON) that uses a physical fiber tree topology to provide efficient point-to-multipoint connectivity from a single OLT to one or more ONUs. Clause 64 specifies the instantiation of multiple MAC entities within the OLT, each with an associated service access point that provides point-to-point connectivity to a specific ONU separate from the connectivity provided to other ONUs. An additional MAC instance provides a Single Copy Broadcast (SCB) service access point that allows a single copy of a frame to be received by all ONUs.

MACsec provides a separate instance of the secure MAC Service to provide bi-directional connectivity between each ONU and the OLT, as illustrated in Figure 12-1, and thus ensures the confidentiality, integrity, and origin authenticity of each data frame sent and received by the OLT and each ONU. These guarantees are provided irrespective of the ability of an attacker to transmit or receive frames to or from the OLT or any ONU, even if that attacker can exactly mimic the EPON media access method specific behavior of any of the securely communicating participants.

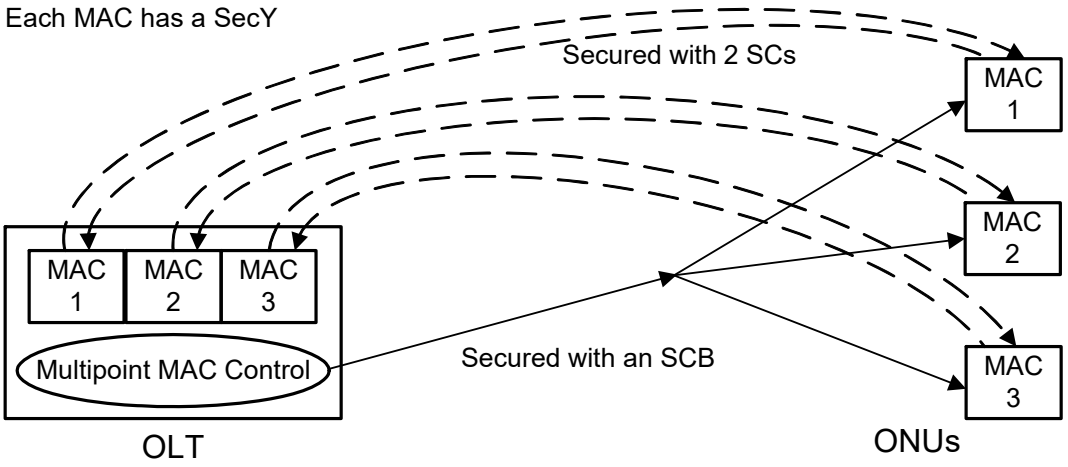


Figure 12-1—MACsec with EPON, showing SCs and SCB

In the OLT, each instance of the secure MAC Service is provided by a distinct SecY that uses the insecure instance of the MAC Service provided by one of the point-to-point MAC entities in the OLT.

MACsec can support the SCB service access point with a dedicated SC. Appropriate distribution to the ONUs of the encryption and authentication keys for the sequence of SAs that compose the SC ensures the confidentiality, integrity, and origin of each frame sent using the SCB.

NOTE 1—Since the SCB MAC interfaces in the OLT lacks a peer interface in each ONU, the keys for the sequence of SAs that support them are distributed to the Key Agreement Entities of all authorized ONUs using the insecure bi-directional MAC Service associated with each of the point-to-point MAC instances.

NOTE 2—An ONU can elect to discard frames from the SCB as these are readily identifiable by the EPON MAC. However, if such frames are received, their integrity and origin should be secured, particularly if the system comprising the ONU bridges or routes such frames. Otherwise, an attacker could use frames that appear to be sent using the SCB to penetrate the attached network, even if the point-to-point EPON connectivity has been correctly secured.

13. MAC Security Entity MIB

13.1 Introduction

This clause contains an SMIv2 Management Information Base (MIB) for managing the operation of a MAC Security Entity (SecY), based on the specifications contained in Clause 10 and Clause 11. This clause includes a MIB module that is compliant to SMIv2.

13.2 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 memo specifies a MIB module that is compliant to the SMIv2, which is described in IETF RFC 2578, IETF RFC 2579, and IETF RFC 2580.

13.3 Relationship to other MIBs

13.3.1 System MIB Group

It is assumed that a system implementing this MIB will also implement the “system” group defined in IETF RFC 3418 (or at least that subset of the system group defined in IETF RFC 1213).

13.3.2 Relationship to the Interfaces MIB

It is assumed that a system implementing this MIB module will implement 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.

The MACsec defines a secure shim layer, SecY, in the interface stack. The IEEE SecY MIB specifies the detail attributes of the secure shim layer in the interface stack. As such, it needs to integrate with IF-MIB. For interface stack diagram, refer to Figure 13-1.

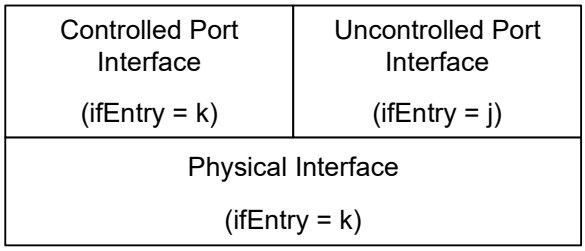


Figure 13-1—MACsec Interface Stack

The SecY’s Controlled Port is a service access point that provides one instance of the secure MAC Service, and the SecY’s Uncontrolled Port is a service access point providing one instance of the insecure MAC Service. According to IETF RFC 2863 these two service access points should be defined as sublayers in the interface stack. Each should have its own conceptual row in the ifTable, though these should be created

together. The two interfaces co-exist without interference—one is not "on top of " the other. The interface type for Controlled Port is defined as macSecControlledIF(231) and the interface type for Uncontrolled Port is defined as macSecUncontrolledIF(232).

The MAC_Enabled and MAC_Operational parameters (6.4) could be mapped to the ifAdminStatus and ifOperStatus objects. The ifAdminStatus object for Controlled Port interface and Uncontrolled port interface should be read only.

MIB tables in this SecY MIB represent information of the Controlled Port interface and are indexed by the interface number pointing to the Controlled Port sublayer interface in the interface stack.

The attributes in Table 13-1 are part of the required ifGeneralInformationGroup object group specified in IETF RFC 2863, and are not duplicated in the SecY MIB.

Table 13-1—Use of ifGeneralInformationGroup Objects

ifGeneralInformationGroup Objects	Use for MACsec
ifDescr	See interfaces MIB (IETF RFC 2863).
ifType	controlled port: macSecControlledIF(231). uncontrolled port: macSecUncontrolledIF(232).
ifSpeed	controlled port: same as the physical interface ifSpeed. uncontrolled port: same as the physical interface ifSpeed.
ifPhysAddress	This object should have an octet string with zero length for a SecY's controlled port and uncontrolled port.
ifAdminStatus	See interfaces MIB, read only for controlled port and uncontrolled port.
ifOperStatus	See interfaces MIB.
ifLastChange	See interfaces MIB.
ifName	See interfaces MIB.
ifLinkUpDownTrapEnable	See interfaces MIB, Default set as follows: controlled port: disabled(2). uncontrolled port: disabled(2).
ifHightSpeed	See interfaces MIB controlled port: same as the physical interfaces's ifHighSpeed. uncontrolled port: same as the physical interfaces's ifHighSpeed.
ifConnectorPresent	See interfaces MIB. Default set as follows: controlled port: false(2). uncontrolled port: false(2).
ifAlias	See interfaces MIB.
ifTableLastChange	See interfaces MIB.

The attributes in Table 13-2 are part of the required ifCounterDiscontinuityGroup object group specified in IETF RFC 2863 and are not duplicated in the SecY MIB.

Table 13-2—Use of ifCounterDiscontinuityGroup Object

ifCounterDiscontinuityGroup Object	Use for MACsec
ifCounterDiscontinuityTime	See interfaces MIB. controlled port: always 0, no discontinuity. uncontrolled port: always 0, no discontinuity.

ifStackTable will be used to identify the layer relationships between Controlled Port interface and physical interface and between Uncontrolled Port interface and physical interface. Use of ifStackTable is necessary to represent the interface stack with MACsec service capability. Refer to Figure 13-1.

The ifStackTable is then used to show the relationships between the various MACsec interfaces, as illustrated Table 13-3.

Table 13-3—Use of ifStackTable

HigherLayer	LowerLayer
j	i
k	i

The attributes in Table 13-4 are part of the required ifStackGroup2 object group specified in IETF RFC 2863, and are not duplicated in the SecY MIB.

Table 13-4—Use of ifStackGroup2 Objects

ifStackGroup2 Objects	Use for MACsec
ifStackStatus	See interfaces MIB.
ifStackLastChange	See interfaces MIB.

The use of the ifPacketGroup object group specified in IETF RFC 2863 is described in 10.7.6 for the Controlled Port interface and in 10.7.3 for the Uncontrolled Port interface.

The ifRcvAddressTable is not applicable for Controlled Port and Uncontrolled Port interfaces.

13.4 Security considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. All such objects are sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

a) secyIfTable, secyIfCipherTable, secyIfTCTable, and secyIfAPTable contain system level information for each interface supported by the SecY. SET access to these tables by unauthorized

persons can disable the MAC security protection functions, block network connectivity, and impact network performance. A comparison of the secyIfTable and the IF-MIB can identify which ports are not protected by a SecY.

b) secyRxSANextPN (deprecated) in secyRxSATable provides the capability to change the replay protection window. SET access to this object by unauthorized persons can affect the MACsec replay protection function, block network connectivity, and impact network performance.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) are sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and to encrypt the values of these objects when sending them over the network via SNMP.

The MIB module provides statistics from the interface level (SecY) to each secure association (SA). These statistics provide information for the diagnosis or debugging of the migration from a non-secure environment to a secure environment and can be used to observe the activities of MACsec operation. This information is useful for security monitoring by authorized personnel, but is also potentially useful to attackers; therefore, it needs to be protected against unauthorized access.

These are the tables and objects and their sensitivity/vulnerability:

c) secyTSATable (and secyTxSATable deprecated) provides information on each transmitting SA. secyTSAConfidentiality exposes whether confidentiality is supported or not for the SA. This information could help an attacker focus their attacks on traffic without confidentiality protection.
d) secyRxSATable contains information about receiving SAs. secyRxSANextPN is used in replay protection to determine which frames should be discarded. Read access to these related parameters could allow an attacker to know the PN range that an attempted replay must fall within.
e) secyCipherSuiteTable provides information about the capabilities of the cipher suites supported by the implementation. Access to this information could allow an attacker to focus their attacks on implementations with specific cipher suites and specific weaknesses.
f) secyRxSAStatsTable (deprecated) and secyRxSCStatsTable contain statistics for each receiving SA and each receiving SC. Read access could allow an attacker to compare these statistics with Figure 10-5 to determine which aspect of their attack failed, and to modify their attack until a different counter is incremented, indicating that they have succeeded in meeting a particular requirement.
g) secyStatsTable contains statistics about the SecY. This information is SecY interface level statistics information, and also read access to this information can help an attacker determine if a system might be vulnerable.
h) The global parameters secyIfMaxPeerSCs, secyIfRxMaxKeys, and secyIfTxMaxKeys might be used by an attacker when attempting to overload the system capabilities to cause a denial of service attack.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

If SNMP is to be used, activation of the security features provided by the SNMPv3 framework (see IETF RFC 3410 [B7], section 8), including the SNMPv3 cryptographic mechanisms (for authentication and privacy) are required for the security goals of this standard to be met.

Further, implementers should not deploy SNMP versions prior to SNMPv3. Instead, implementers should deploy SNMPv3 to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

13.5 Structure of the MIB module

A single MIB module is defined in this clause. Within the MIB module, each SecY is identified by the InterfaceIndex used by the Interfaces MIB (13.3.2, Figure 13-1) for the Controlled Port sublayer interface. This facilitates identification of the SecY when investigating an interface stack, and discovery of the other entities (via the Interfaces MIB) that are related to a particular SecY, including the associated PAE.

At the top level, the MIB module identifies MIB notifications, MIB objects, and MIB conformance information, though no notifications are defined. Figure 13-1 illustrates the structure of the MIB module, as described in this clause.

NOTE 1—MIB conformance is represented by objects, with an initial OID (object identifier) of secyMIBConformance, but in this MIB description the term MIB objects refers specifically to objects with an initial OID of secyMIBObjects.

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). MIB objects are further classified into management and statistics objects.

The entries in each of the management object tables are indexed in one of the following ways:

a) By the interface index, if the objects in each entry are for the SecY identified by that index).
b) By the interface index and SCI, if the objects are for an SCI used by the SecY.
c) By the interface index, SCI, and AN, if the objects are for an SA used by the SecY.
d) By a cipher suite index, defined in the module, for information specific to a given cipher suite but applicable to all SecYs in the system.
e) By a cipher suite index and SCI, for cipher suite information for a given SecY.

Each of the statistics object table entries augments a particular management object table entry, effectively using the same index. The per SecY, per SC, and per SA management tables for Controlled Port transmission and reception are summarized in Table 13-5 and Table 13-6, and the corresponding statistics tables in Table 13-7. Management tables for system-wide and per SecY Cipher Suite information are summarized in Table 13-8.

The MIB conformance objects are organized into compliance statements and conformance groups.

The compliance statement (secyMIBCompliance) for the original revision of this MIB module, published in IEEE Std 802.1AE-2006, mandated implementation of each of the groups specified in that revision, and each of the management object tables and statistics object tables specified has a corresponding group that includes all the objects for each entry in the table, though read-only access is permitted for a number of objects. This compliance statement and some of the tables have now been deprecated, and implementation of all the tables and objects it specifies is no longer a requirement of this standard for a conformance claim of network management MIB support. Implementations may continue to support this original compliance statement to support interoperability, either as an additional or as the only supported compliance statement, but it is expected to become obsolete in some future revision.

The current revision of this MIB module adds support for multiple traffic classes (with a complementary reduction in per SC statistics collection) and extended packet number reporting, with an additional compliance statement (secyMIBTcCompliance) and new conformance groups for additional tables. An implementation of a SecY for which support of network management using the MIB module is claimed should support secyMIBTcCompliance. An implementation that does not require multiple traffic class support can still benefit from reduced statistics collection and from the resolution of minor inconsistencies with the normative text elsewhere in the standard (which takes precedence).

NOTE 2—Annex G provides additional information on the management and MIB revisions.

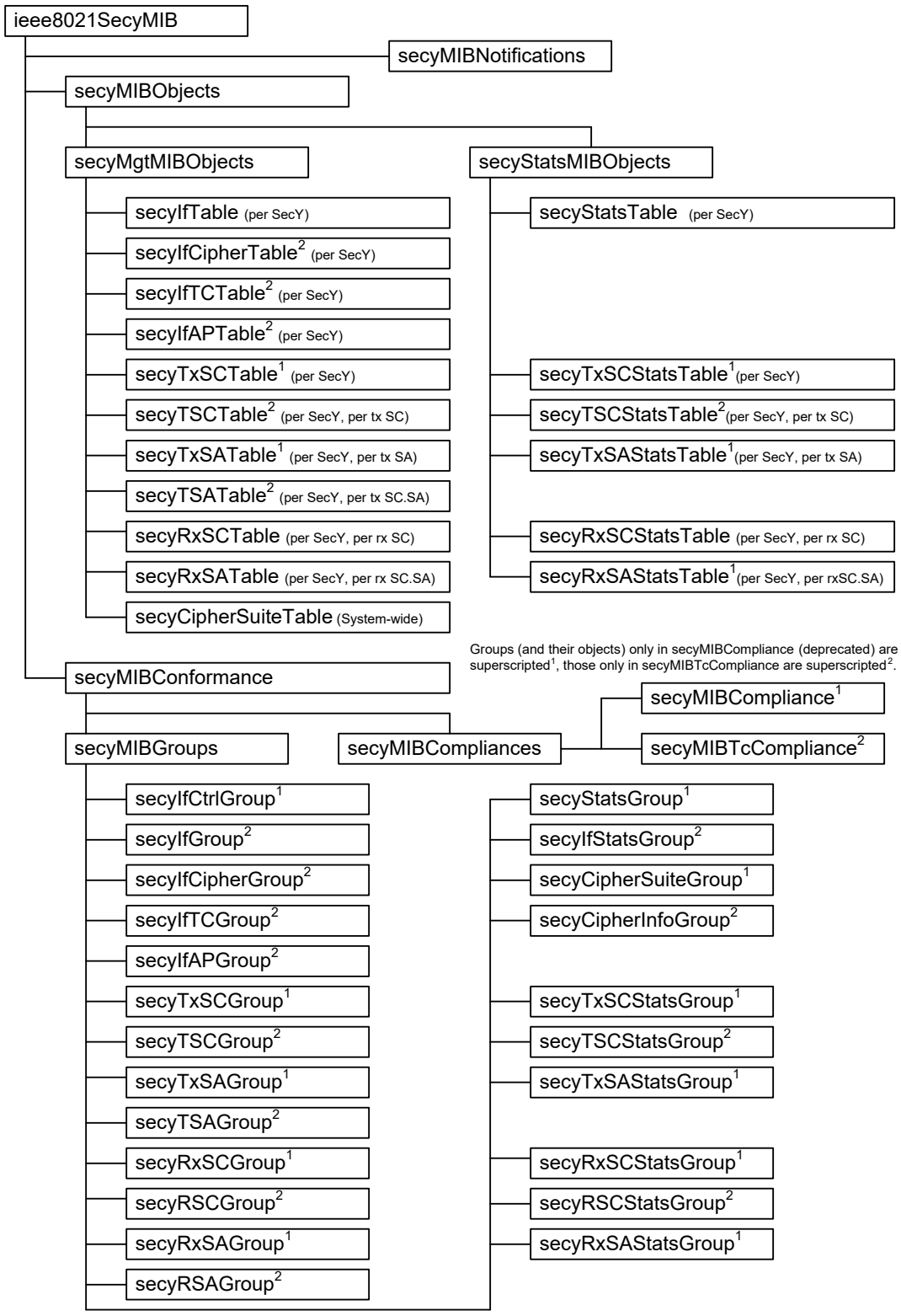


Figure 13-2—SecY MIB structure

Table 13-5—Controlled Port service management

Table	Table Entry objects	Figure 10-5 reference and definition
secyIfTable secyIfEntry [secyIfInterfaceIndex] secyIfCtrlGroup[1a] secyIfGroup[2]	secyIfInterfaceIndex secyIfSCI[2] secyIfMaxPeerSCs secyIfRxMaxKeys secyIfTxMaxKeys secyIfProtectFramesEnable secyIfValidateFrames secyIfReplayProtectEnable secyIfReplayProtectWindow secyIfCurrentCipherSuite secyIfAdminPt2PtMAC secyIfOperPt2PtMAC secyIfIncludeSCIEnable secyIfUseESEnable secyIfUseSCBEnable secyIfIncludingSCI[2] secyIfMaxTSCs[2]	Aligned with IF-MIB InterfaceIndex (10.1) Generation.sci (7.1.2, 8.2.1, 10.7.1) Verification.maxReceiveChannels (10.7.7) Verification.maxReceiveKeys (10.7.7) Generation.maxTransmitKeys (10.7.16) Generation.protectFrames (10.7.17, Fig 10-3) Verification.validateFrames (10.7.8, Fig 10-4) Verification.replayProtect (10.7.8, Fig 10-4) Verification.replayWindow (10.7.8, Fig 10-4) CurrentCipherSuite.ciphersuite (10.7.25) controlledPort.adminPointToPointMAC (6.5, 10.7.4) controlledPort.operPointToPointMAC (6.5, 10.7.4) Generation.alwaysIncludeSCI (10.5.3,10.7.17) Generation.useES (10.5.3,10.7.17) Generation.useSCB (10.5.3, 10.7.17) Generation.includingSCI (10.5.3, 10.7.17, Fig 10-3) Generation.maxTransmitChannels (10.7.16)
secyIfCipherTable[2] secyIfCipherEntry[2] [secyIfInterfaceIndex, secyCipherSuiteIndex] secyIfCipherGroup[2]	secyIfCipherImplemented[2b] secyIfCipherEnableUse[2] secyIfCipherRqConfidentiality[2]	implementedCipherSuites (10.7.26) CipherSuiteControl.enableUse (10.7.26) CipherSuiteControl.requireConfidentiality (10.7.26)
secyIfTCTable[2] secyIfTCEntry[2] [secyIfInterfaceIndex, secyTCUserPriority[2]] secyTCGroup[2]	secyIfTCUserPriority[2] secyIfTCTrafficClass[2]	Generation.userpri (10.5.1, 10.7.17) TrafficClass.trafficClass (10.5.1, 10.7.17)
secyIfAPTable[2] secyAPEntry[2] [secyIfInterfaceIndex, secyAPUserPCP[2]] secyIfAPGroup[2]	secyIfAPUserPCP[2] secyIfAPAccessPCP[2]	Generation.userpcp (10.5, 10.7.17) AccessPriority.accessPCP (10.5, 10.7.17)

[a] Tables, table entries, groups and objects that appear only in secyMIBCompliance (deprecated) are superscripted[1], and those that appear only in secyTcMIBCompliance (recommended) are superscripted[2]. Those that are used in both are not superscripted.

[b] If the CipherSuite referenced by secyCipherSuiteIndex is not implemented for the SecY identified by the secyIfInterfaceIndex, the corresponding instance of this object is not required, if it is present secyCipherSuiteAvailable (read-only) and secyCipherSuiteEnable (normally read-write) will be False and not writable.

Table 13-6—Transmit and receive SC management

Table	Table Entry objects	Figure 10-5 reference and definition
secyTxSCTable[1a] secyTxSCEntry[1] [secyIfInterfaceIndex] secyTxSCGroup[1]	secyTxSCI[1] secyTxSCState[1] secyTxSCEncodingSA[1] secyTxSCEncipheringSA[1] secyTxSCCreatedTime[1] secyTxSCStartedTime[1] secyTxSCStoppedTime[1]	TransmitSC.sci (7.1.2, 10.7.1) TransmitSC.transmitting (10.7.21, 10.7.23) TransmitSC.encodingSA (10.5.1, 10.7.21) deprecated (10.5.4) TransmitSC.createdTime (10.7.21) TransmitSC.startedTime (10.7.21) TransmitSC.stoppedTime (10.7.21)
secyTSCTable secyTSCEntry[2] [secyIfInterfaceIndex, secyTSCI] secyTSCGroup[2]	secyTSCI[2] secyTSCState[2] secyTSCEncodingSA[2] secyTSCCreatedTime[2] secyTSCStartedTime[2] secyTSCStoppedTime[2]	TransmitSC.sci (7.1.2, 10.7.17, 10.7.20) TransmitSC.transmitting (10.7.21, 10.7.23) TransmitSC.encodingSA (10.5.1, 10.7.21) TransmitSC.createdTime (10.7.21) TransmitSC.startedTime (10.7.21) TransmitSC.stoppedTime (10.7.21)
secyTxSATable[1] secyTxSAEntry[1] [secyIfInterfaceIndex, secyTxSA] secyTxSAGroup[1]	secyTxSA[1] secyTxSAState[1] secyTxSANextPN[1] secyTxSAConfidentiality[1] secyTxSASAKUnchanged[1] secyTxSACreatedTime secyTxSAStartedTime secyTxSAStoppedTime	TransmitSC.txa (10.7.22) TransmitSA.inUse (10.7.23) TransmitSA.nextPN (10.5, 10.7.23) TransmitSA.confidentiality (10.7.23) deprecated TransmitSA.createdTime (10.7.23) TransmitSA.startedTime (10.7.23) TransmitSA.stoppedTime (10.7.23)
secyTSATable[2] secyTSAEntry[2] [secyIfInterfaceIndex, secyTSCI, secyTSA] secyTSAGroup[2]	secyTSA[2] secyTSAState[2] secyTSANextXPN[2] secyTSAConfidentiality[2] secyTSACreatedTime[2] secyTSAStartedTime[2] secyTSAStoppedTime[2] secyTSAKeyIdentifier[2] secyTSASSCI[2]	TransmitSC.txa (10.7.22) TransmitSA.inUse (10.7.23) TransmitSA.nextPN (10.5, 10.7.23) TransmitSA.confidentiality (10.7.23) TransmitSA.createdTime (10.7.23) TransmitSA.startedTime (10.7.23) TransmitSA.stoppedTime (10.7.23) TransmitSA.keyIdentifier (10.7.23) TransmitSA.ssci (10.7.23)
secyRxSCTable secyRxSCEntry [secyIfInterfaceIndex, secyRxSCI] secyRxSCGroup[1] secyRcSCGroup[2]	secyRxSCI secyRxSCState secyRxSCCurrentSA[1] secyRxSCCreatedTime secyRxSCStartedTime secyRxSCStoppedTime	ReceiveSC.sci (10.7.11) ReceiveSC.receiving (10.7.12, 10.7.14, 10.7.15) deprecated ReceiveSC.createdTime (10.7.12) ReceiveSC.startedTime (10.7.12) ReceiveSC.stoppedTime (10.7.12)
secyRxSATable secyRxSAEntry [secyIfInterfaceIndex, secyRxSCI, secyRxSA] secyRxSAGroup[1] secyRSAGroup[2]	secyRxSA secyRxSAState secyRxSANextPN[1] secyRxSANextXPN[2] secyRxSALowestXPN[2] secyRxSASAKUnchanged[1] secyRxSACreatedTime secyRxSAStartedTime secyRxSAStoppedTime secyRxSAKeyIdentifier[2] secyRxSASSCI[2]	ReceiveSC.rxa (10.7.13) ReceiveSA.inUse (10.7.14) deprecated ReceiveSA.nextPN (10.7.14) ReceiveSA.lowestPN (10.6.2, 10.6.4, 10.6.5, 10.7.14, Fig 10-4) deprecated ReceiveSA.createdTime (10.7.14) ReceiveSA.startedTime (10.7.14) ReceiveSA.stoppedTime (10.7.14) ReceiveSA.keyIdentifier (10.7.14) ReceiveSA.ssci (10.7.14)

[a] Tables, table entries, groups and objects that appear only in secyMIBCompliance (deprecated) are superscripted[1], and those that appear only in secyTcMIBCompliance (recommended) are superscripted[2]. Those that are used in both are not superscripted.

Table 13-7—Transmit and receive statistics

Table	Table Entry objects	Figure 10-5 reference and definition
secyStatsTable secyStatsEntry augments secyIfEntry secyStatsGroup[1a] secyIfStatsGroup[2] secyCipherStatsGroup[2]	secyStatsTxUntaggedPkts secyStatsTxTooLongPkts secyStatsRxUntaggedPkts secyStatsRxNoTagPkts secyStatsRxBadTagPkts secyStatsRxUnknownSCIPkts[1] secyStatsRxNoSCIPkts[1] secyStatsRxOverrunPkts secyStatsRxNoSAPkts[2] secyStatsRxNoSAErrorPkts[2] secyStatsTxOctetsProtected[2] secyStatsTxOctetsEncrypted[2] secyStatsRxOctetsValidated[2] secyStatsRxOctetsDecrypted[2]	Generation.OutPktsUntagged (10.7.18, Fig 10-3) Generation.OutPktsTooLong (10.7.18, Fig 10-3) Verification.InPktsUntagged (10.7.18, Fig 10-4) Verification.InPktsNoTag (10.7.9, Fig 10-4) Verification.InPktsBadTag (10.7.9, Fig 10-4) deprecated deprecated Verification.InPktsOverrun (10.7.9, Fig 10-4) Verification.InPktsNoSA (10.7.9, Fig 10-4) Verification.InPktsNoSAError(10.7.9, Fig 10-4) Generation.OutOctetsProtected (10.7.9, Fig 10-3) Generation.OutsOctetsEncrypted (10.7.9, Fig 10-3) Verification.InOctetsValidated (10.6.3, Fig 10-4) Verification.InOctetsValidated (10.6.3, Fig 10-4)
secyTxSCStatsTable[1] secyTxSCStatsEntry[1] augments secyTxSCEntry secyTxSCStatsGroup[1]	secyTxSCStatsProtectedPkts [1] secyTxSCStatsEncryptedPkts[1] secyTxSCStatsOctetsProtected[1] secyTxSCStatsOctetsEncrypted[1]	TransmitSC.OutPktsProtected (10.7.18, Fig 10-3) TransmitSC.OutPktsEncrypted (10.7.18, Fig 10-3) deprecated deprecated
secyTSCStatsTable[2] secyTSCStatsEntry[2] augments secyTSCEntry secyTSCStatsGroup[2]	secyTSCStatsProtectedPkts[2] secyTSCStatsEncryptedPkts[2]	TransmitSC.OutPktsProtected (10.7.18, Fig 10-3) TransmitSC.OutPktsEncrypted (10.7.18, Fig 10-3)
secyTxSAStatsTable[1] secyTxSAStatsEntry[1] augments secyTxSAEntry[1] secyTxSAStatsGroup[1]	secyTxSAStatsProtectedPkts[1] secyTxSAStatsEncryptedPkts[1]	deprecated deprecated
secyRxSCStatsTable secyRxSCStatsEntry augments secyRxSCEntry secyRxSCStatsGroup[1] secyRSCStatsGroup[2]	secyRxSCStatsUnusedSAPkts[1] secyRxSCStatsNoUsingSAPkts[1] secyRxSCStatsLatePkts secyRxSCStatsNotValidPkts secyRxSCStatsInvalidPkts secyRxSCStatsDelayedPkts secyRxSCStatsUncheckedPkts secyRxSCStatsOKPkts secyRxSCStatsOctetsValidated[1] secyRxSCStatsOctetsDecrypted[1]	deprecated deprecated ReceiveSC.InPktsLate (10.7.9, Fig 10-4) ReceiveSC.InPktsNotValid (10.7.9, Fig 10-4) ReceiveSC.InPktsInvalid (10.7.9, Fig 10-4) ReceiveSC.InPktsDelayed (10.7.9, Fig 10-4) ReceiveSC.InPktsUnchecked (10.7.9, Fig 10-4) ReceiveSC.InPktsOK(10.7.9, Fig 10-4) deprecated deprecated
secyRxSAStatsTable[1] secyRxSAStatsEntry augments secyRxSAEntry secyRxSAStatsGroup[1]	secyRxSAStatsUnusedSAPkts[1] secyRxSAStatsNotUsingSAPkts[1] secyRxSAStatsNotValidPkts[1] secyRxSAStatsInvalidPkts[1] secyRxSAStatsOKPkts[1]	deprecated deprecated deprecated deprecated deprecated

[a] Tables, table entries, groups and objects that appear only in secyMIBCompliance (deprecated) are superscripted[1], and those only in secyTcMIBCompliance (recommended) are superscripted[2]. Those that are used in both are not superscripted.

Table 13-8—Cipher Suite information

Table Table Entry [Index]	Table Entry Objects	Figure 10-5 reference and definition
secyCipherSuiteTable	secyCipherSuiteIndex	
secyCipherSuiteEntry	secyCipherSuiteId	CipherSuite.identifier (10.7.25, Table 14-1)
[secyCipherSuiteIndex]	secyCipherSuiteName	CipherSuite.name (10.7.25, Table 14-1)
	secyCipherSuiteCapability	CipherSuite.integrityProtection, confidentialityProtection (10.7.25)
secyCipherSuiteGroup[1]		
secyCipherInfoGroup[2]	secyCipherSuiteProtection[1]	deprecated
	secyCipherSuiteProtectionOffset[1]	deprecated
	secyCipherSuiteDataLengthChange	CipherSuite.changesDataLength (10.7.25)
	secyCipherSuiteICVLength	CipherSuite.ICVlength (10.7.25)
	secyCipherSuiteRowStatus[1]	deprecated

13.6 MAC Security Entity (SecY) MIB definitions[15]

```
-- ****************************************************************
-- IEEE8021-SECY-MIB
--
-- Definitions of managed objects supporting IEEE 802.1AE MACsec.
-- ****************************************************************

IEEE8021-SECY-MIB DEFINITIONS ::= BEGIN

-- ----------------------------------------------------------------
-- IEEEE802.1AE 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    "201712071816Z"
    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
                                  P.O. Box 1331
                                  Piscataway
                                  NJ 08855-1331
                                  USA
                         E-mail:  STDS-802-1-L@IEEE.ORG"
    DESCRIPTION
    "The MAC security entity (SecY) MIB module. A SecY is a protocol
     shim providing MAC Security (MACsec) in an interface stack.
```

[15] Copyright release for MIBs: Users of this standard may freely reproduce the MIB definition contained in this clause so that it can be used for its intended purpose.

```
     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. 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)'.

     _____________________________________|_____________________________
     |                                    |                            |
     |  Controlled Port Interface         |   Uncontrolled Port Interface  |
     |  (ifEntry = j,ifType =             |   (ifEntry = k, ifType =       |
     |   macSecControlledIF(231))         |    macSecUncontrolledIF(232))  |
     |____________________________________|____________________________|
     |                                                                 |
     |                    Physical Interface                           |
     |                      (ifEntry = i)                              |
     |                (ifType = ethernetCsmacd(6))                     |
     |_________________________________________________________________|
        Example MACsec Interface Stack. i, j, k are ifIndexes each
        indicating a row in the ifTable.
    "
    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 }

secyMIBNotifications OBJECT IDENTIFIER ::= { ieee8021SecyMIB 0 }

secyMIBObjects OBJECT IDENTIFIER ::= { ieee8021SecyMIB 1 }

secyMIBConformance OBJECT IDENTIFIER ::= { ieee8021SecyMIB 2 }

--
-- 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   "IEEE 802.1AE Clause 7.1.2 and 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   "IEEE 802.1AE Clause 7.1.3, Figure 7.7"
    SYNTAX Unsigned32 (0..3)

secyMgmtMIBObjects OBJECT IDENTIFIER ::= { secyMIBObjects 1 }

secyStatsMIBObjects OBJECT IDENTIFIER ::= { secyMIBObjects 2 }

--
-- SecY Interface Management Table
--

secyIfTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyIfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "A table with an entry for each service interface in this system
     with MAC Security capability, i.e. for each SecY.

     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 "IEEE 802.1AE Clause 10.7, Table 13-1"
    ::= { secyMgmtMIBObjects 1 }

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 ::= 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,     -- 802.1AEcg
    secyIfIncludingSCI              TruthValue,  -- 802.1AEcg
    secyIfMaxTSCs                   Unsigned32   -- 802.1AEcg
}

secyIfInterfaceIndex    OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION "An interface index, aligned with ifIndex in the
                 ifTable, pointing to the SecY's Controlled Port."
    REFERENCE   "IEEE 802.1AE Clause 10.1"
    ::= { secyIfEntry 1 }

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   "IEEE 802.1AE Clause 10.7.7"
    ::= { secyIfEntry 2 }

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   "IEEE 802.1AE Clause 10.7.7"
    ::= { secyIfEntry 3 }

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   "IEEE 802.1AE Clause 10.7.16"
    ::= { secyIfEntry 4 }

secyIfProtectFramesEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Enables or disables protection of transmitted frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.17, Figure 10-3"
    DEFVAL { true }
    ::= { secyIfEntry 5 }

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) from received frames.
     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   "IEEE 802.1AE Clause 10.7.8, Figure 10-4"
    DEFVAL { strict }
    ::= { secyIfEntry 6 }

secyIfReplayProtectEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Enables or disables replay protection."
    REFERENCE   "IEEE 802.1AE Clause 10.7.8, Figure 10-4"
```

```
    DEFVAL { true }
    ::= { secyIfEntry 7 }

secyIfReplayProtectWindow    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "Packets"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The replay protection window size."
    REFERENCE   "IEEE 802.1AE Clause 10.7.8, Figure 10-4"
    DEFVAL { 0 }
    ::= { secyIfEntry 8 }

secyIfCurrentCipherSuite    OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The Cipher Suite currently used by this SecY,
                 identified by the secyCipherSuiteTable entry index.
                 Should be read-only if secyIfCipherTable implemented."
    REFERENCE   "IEEE 802.1AE Clause 10.7.25"
    ::= { secyIfEntry 9 }

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   "IEEE 802.1AE Clause 6.5, 10.7.4"
    DEFVAL { auto }
    ::= { secyIfEntry 10 }

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   "IEEE 802.1AE Clause 6.5, 10.7.4"
    ::= { secyIfEntry 11 }

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   "IEEE 802.1AE Clause 10.5.3 alwaysIncludeSCI, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 12 }

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   "IEEE 802.1AE Clause 10.5.3 useES, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 13 }

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   "IEEE 802.1AE Clause 10.5.3 useSCB, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 14 }

secyIfSCI       OBJECT-TYPE
    SYNTAX      SecySCI
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The SCI for the SecY's default traffic class."
    REFERENCE   "IEEE 802.1AE Clause 7.1.2, 10.7.1"
    ::= { secyIfEntry 15 }

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   "IEEE 802.1AE Clause 10.5.3 includingSCI, 10.7.17"
    DEFVAL { false }
    ::= { secyIfEntry 16 }

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   "IEEE 802.1AE Clause 10.7.16"
    ::= { secyIfEntry 17 }

--
-- Tx SC Management Table : systems not supporting traffic class SCs
--

secyTxSCTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table with an entry for each SecY's transmit SC."
    REFERENCE   "IEEE 802.1AE Clause 10.7.17, 10.7.20, Table 13-2"
    ::= { secyMgmtMIBObjects 2 }

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 ::= SEQUENCE {
    secyTxSCI              SecySCI,
    secyTxSCState          INTEGER,
    secyTxSCEncodingSA     RowPointer,
    secyTxSCEncipheringSA  RowPointer, -- deprecated
    secyTxSCCreatedTime    TimeStamp,
```

```
    secyTxSCStartedTime    TimeStamp,
    secyTxSCStoppedTime    TimeStamp
}

secyTxSCI       OBJECT-TYPE
    SYNTAX      SecySCI
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The SCI for the SecY's transmit SC."
    REFERENCE   "IEEE 802.1AE Clause 7.1.2, 10.7.1"
    ::= { secyTxSCEntry 1 }

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.
                 inUse(1) : one or more SAs are in use.
                 notInUse(2) : no SAs are in use."
    REFERENCE   "IEEE 802.1AE Clause 10.7.21 transmitting, 10.7.23"
    ::= { secyTxSCEntry 2 }

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 will point to an entry in the
     secyTxSATable. If no such information is available, the value shall
     be the OBJECT IDENTIFIER { 0 0 }."
    REFERENCE   "IEEE 802.1AE Clause 10.5.1, 10.7.21"
    ::= { secyTxSCEntry 3 }

secyTxSCEncipheringSA    OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "The SA currently used to encipher frames for transmission.
     The row pointer will point to an entry in the secyTxSATable.
     If no such information is available, the value shall be the
     OBJECT IDENTIFIER { 0 0 }."
    REFERENCE   "IEEE 802.1AE Clause 10.5.4"
    ::= { secyTxSCEntry 4 }

secyTxSCCreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC was created."
    REFERENCE   "IEEE 802.1AE Clause 10.7.21"
    ::= { secyTxSCEntry 5 }

secyTxSCStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC last started
                 transmitting MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.21"
    ::= { secyTxSCEntry 6 }

secyTxSCStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC last stopped
                 transmitting MACsec frames."
```

```
    REFERENCE   "IEEE 802.1AE Clause 10.7.21"
    ::= { secyTxSCEntry 7 }

--
-- Traffic Class capable transmit SC Management Table : 802.1AEcg
--

secyTSCTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of entries for each SecY's traffic class SCs."
    REFERENCE   "IEEE 802.1AE Clause 7.1.2, 10.7.17, 10.7.20"
    ::= { secyMgmtMIBObjects 10 }

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 ::= SEQUENCE {
    secyTSCI              SecySCI,
    secyTSCState          INTEGER,
    secyTSCEncodingSA     RowPointer,
    secyTSCCreatedTime    TimeStamp,
    secyTSCStartedTime    TimeStamp,
    secyTSCStoppedTime    TimeStamp
}

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   "IEEE 802.1AE Clause 7.1.2, 10.7.17, 10.7.20"
    ::= { secyTSCEntry 1 }

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 for the traffic class SC are in use.
     notInUse(2) : no SAs for the traffic class SC are in use."
    REFERENCE   "IEEE 802.1AE Clause 10.7.20"
    ::= { secyTSCEntry 2 }

secyTSCEncodingSA    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 will point to an entry in the
     secyTxSATable. If no such information is available, the value shall
     be the OBJECT IDENTIFIER { 0 0 }."
    REFERENCE   "IEEE 802.1AE Clause 10.5.1, 10.7.21"
    ::= { secyTSCEntry 3 }

secyTSCCreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC was created."
    REFERENCE   "IEEE 802.1AE Clause 10.7.21"
    ::= { secyTSCEntry 4 }

secyTSCStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC last started
                 transmitting MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.21"
    ::= { secyTSCEntry 5 }

secyTSCStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmitting SC last stopped
                 transmitting MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.21"
    ::= { secyTSCEntry 6 }

--
-- Tx SA Management Table : systems not supporting traffic class SCs
--

secyTxSATable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table with an entry for each transmit SA for each of
                 the system's SecYs."
    REFERENCE   "IEEE 802.1AE Clause 10.7.22, Table 13-2"
    ::= { secyMgmtMIBObjects 3 }

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

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

secyTxSA    OBJECT-TYPE
    SYNTAX      SecyAN
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The association number (AN) for this transmit SA."
    REFERENCE   "IEEE 802.1AE Clause 10.7.22"
    ::= { secyTxSAEntry 1 }

secyTxSAState    OBJECT-TYPE
    SYNTAX      INTEGER {
                    inUse(1),
                    notInUse(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current status of the transmitting SA.
```

```
                inUse(1)    : this SA is in use.
                notInUse(2) : this SA is not in use."
    REFERENCE   "IEEE 802.1AE Clause 10.7.22"
    ::= { secyTxSAEntry 2 }

secyTxSANextPN    OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The next packet number (PN) for this SA."
    REFERENCE   "IEEE 802.1AE Clause 10.5, 10.7.23"
    ::= { secyTxSAEntry 3 }

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   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTxSAEntry 4 }

secyTxSASAKUnchanged    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "A reference to an SAK that is unchanged for the life
                 of the transmitting SA."
    REFERENCE   "IEEE 802.1AE Clause 10.7.22"
    ::= { secyTxSAEntry 5 }

secyTxSACreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA was created."
    REFERENCE   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTxSAEntry 6 }

secyTxSAStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA last started
                 transmitting MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTxSAEntry 7 }

secyTxSAStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA last stopped
                 transmitting MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTxSAEntry 8 }

--
-- Traffic Class capable transmit SA Management Table : 802.1AEcg
--

secyTSATable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTSAEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table with an entry for each transmit SA for each of
                 the system's SecYs."
    REFERENCE   "IEEE 802.1AE Clause 10.7.22, Table 13-2"
    ::= { secyMgmtMIBObjects 11 }

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 ::= SEQUENCE {
    secyTSA                   SecyAN,
    secyTSAState              INTEGER,
    secyTSANextXPN            Counter64,
    secyTSAConfidentiality    TruthValue,
    secyTSAKeyIdentifier      SnmpAdminString,
    secyTSASSCI       Integer32,
    secyTSACreatedTime        TimeStamp,
    secyTSAStartedTime        TimeStamp,
    secyTSAStoppedTime        TimeStamp
}

secyTSA     OBJECT-TYPE
    SYNTAX      SecyAN
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The association number (AN) for this transmit SA."
    REFERENCE   "IEEE 802.1AE Clause 10.7.22"
    ::= { secyTSAEntry 1 }

secyTSAState   OBJECT-TYPE
    SYNTAX      INTEGER {
                    inUse(1),
                    notInUse(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current status of the transmit SA.

                 inUse(1) : this SA is in use.
                 notInUse(2) : this SA is not in use."
    REFERENCE   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTSAEntry 2 }

secyTSANextXPN  OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The next packet number (PN) for this SA."
    REFERENCE   "IEEE 802.1AE Clause 10.5, 10.7.23"
    ::= { secyTSAEntry 3 }

secyTSAConfidentiality    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   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTSAEntry 4 }

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, IEEE 802.1AE Clause 10.7.23"
    ::= { secyTSAEntry 5 }

secyTSASSCI     OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The SSCI for this SA, 0 if an XPN Cipher Suite is not
                being used."
    REFERENCE   "IEEE 802.1X, IEEE 802.1AE Clause 10.7.23"
```

```
    ::= { secyTSAEntry 6 }

secyTSACreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA was created."
    REFERENCE   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTSAEntry 7 }

secyTSAStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA last started
                 transmitting MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTSAEntry 8 }

secyTSAStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this transmit SA last stopped
                 transmitting MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.23"
    ::= { secyTSAEntry 9 }

--
-- Rx SC Management Table
--

secyRxSCTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyRxSCEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table for the system's SecY's receive SCs."
    REFERENCE   "IEEE 802.1AE Clause 10.7.11, Table 13-2"
    ::= { secyMgmtMIBObjects 4 }

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

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

secyRxSCI       OBJECT-TYPE
    SYNTAX      SecySCI
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The SCI for the receive SC."
    REFERENCE   "IEEE 802.1AE Clause 10.7.11"
    ::= { secyRxSCEntry 1 }

secyRxSCState   OBJECT-TYPE
    SYNTAX      INTEGER {
                    inUse(1),
                    notInUse(2)
                }
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION "The state of the receive SC.

                 inUse(1) : one or more SAs for this SC are in use.
                 notInUse(2) : no SAs for this SC is in use."
    REFERENCE   "IEEE 802.1AE Clause 10.7.12 receiving,
                                    10.7.14 inUse, 10.7.15"
    ::= { secyRxSCEntry 2 }

secyRxSCCurrentSA    OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "The current receiving association number of the SC in use.
     The row pointer will point 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   "IEEE 802.1AE Clause 10.6.1, 10.7.13"
    ::= { secyRxSCEntry 3 }

secyRxSCCreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this receiving SC was created."
    REFERENCE   "IEEE 802.1AE Clause 10.7.12"
    ::= { secyRxSCEntry 4 }

secyRxSCStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this receiving SC last started
                 receiving MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.12"
    ::= { secyRxSCEntry 5 }

secyRxSCStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this receiving SC last stopped
                 receiving MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.12"
    ::= { secyRxSCEntry 6 }

--
-- Rx SA Management Table
--

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   "IEEE 802.1AE Clause 10.7.13"
    ::= { secyMgmtMIBObjects 5 }

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 ::= SEQUENCE {
    secyRxSA              SecyAN,
    secyRxSAState         INTEGER,
    secyRxSANextPN        Unsigned32,
```

```
    secyRxSASAKUnchanged   TruthValue,
    secyRxSACreatedTime    TimeStamp,
    secyRxSAStartedTime    TimeStamp,
    secyRxSAStoppedTime    TimeStamp,
    secyRxSANextXPN        Counter64,       -- 802.1AEcg
    secyRxSALowestXPN      Counter64,       -- 802.1AEcg
    secyRxSAKeyIdentifier  SnmpAdminString, -- 802.1AEcg
     secyRxSASSCI   Integer32 -- 802.1AEcg
}

secyRxSA        OBJECT-TYPE
    SYNTAX      SecyAN
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The association number (AN) for this receive SA."
    REFERENCE   "IEEE 802.1AE Clause 10.7.13"
    ::= { secyRxSAEntry 1 }

secyRxSAState    OBJECT-TYPE
    SYNTAX      INTEGER {
                    inUse(1),
                    notInUse(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current state for this receive SA."
    REFERENCE   "IEEE 802.1AE Clause 10.7.14"
    ::= { secyRxSAEntry 2 }

secyRxSANextPN    OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "One more than the highest PN conveyed in the SecTAG of a frame
     received on this SA that has been successfully validated (if
     validateFrames has not been disabled). Deprecated: use
     secyRxSANextXPN for both 32-bit PN and 64-bit XPN PN values. If
     this object is implemented and an XPN Cipher Suite is used, it
     contains the lowest 32-bits of the XPN."
    REFERENCE   "IEEE 802.1AE Clause 10.6.5, 10.7.14, Figure 10-4"
    ::= { secyRxSAEntry 3 }

secyRxSASAKUnchanged    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "A reference to an SAK that is unchanged for the life
                 of the receiving SA."
    REFERENCE   "IEEE 802.1AE Clause 10.7.13"
    ::= { secyRxSAEntry 4 }

secyRxSACreatedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this receiving SA was created."
    REFERENCE   "IEEE 802.1AE Clause 10.7.14"
    ::= { secyRxSAEntry 5 }

secyRxSAStartedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The system time when this receiving SA last started
                 receiving MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.14"
    ::= { secyRxSAEntry 6 }

secyRxSAStoppedTime    OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION "The system time when this receiving SA last stopped
                 receiving MACsec frames."
    REFERENCE   "IEEE 802.1AE Clause 10.7.14"
    ::= { secyRxSAEntry 7 }

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   "IEEE 802.1AE Clause 10.6.5, 10.7.14, Figure 10-4"
    ::= { secyRxSAEntry 8 }

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   "IEEE 802.1AE Clause 10.6.2, 10.6.4, 10.6.5, 10.7.14,
        Figure 10-4"
    ::= { secyRxSAEntry 9 }

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, IEEE 802.1AE Clause 10.7.14"
    ::= { secyRxSAEntry 10 }

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
                being used."
    REFERENCE   "IEEE 802.1X, IEEE 802.1AE Clause 10.7.14"
    ::= { secyRxSAEntry 11 }

--
--  SecY Selectable Cipher Suites
--

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   "IEEE 802.1AE Clause 10.7.25"
    ::= { secyMgmtMIBObjects 6 }

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

SecyCipherSuiteEntry ::= SEQUENCE {
    secyCipherSuiteIndex              Unsigned32,
```

```
    secyCipherSuiteId                   OCTET STRING,
    secyCipherSuiteName                 SnmpAdminString,
    secyCipherSuiteCapability           BITS,
    secyCipherSuiteProtection           BITS,
    secyCipherSuiteProtectionOffset     INTEGER,
    secyCipherSuiteDataLengthChange     TruthValue,
    secyCipherSuiteICVLength            Unsigned32,
    secyCipherSuiteRowStatus            RowStatus
}

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

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   "IEEE 802.1AE Clause 10.7.25, Table 14-1"
    ::= { secyCipherSuiteEntry 2 }

secyCipherSuiteName    OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (1..128))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "The Cipher Suite Name, 128 octets or fewer."
    REFERENCE   "IEEE 802.1AE Clause 10.7.25, Table 14-1"
    ::= { secyCipherSuiteEntry 3 }

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   "IEEE 802.1AE Clause 10.7.24, 10.7.25"
    ::= { secyCipherSuiteEntry 4 }

secyCipherSuiteProtection    OBJECT-TYPE
    SYNTAX      BITS {
                    integrity(0),
                    confidentiality(1),
                    offsetConfidentiality(2)
                }
    MAX-ACCESS  read-create
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "The secyIfCipherSuite table should be used instead of this object
     to allow per SecY Cipher Suite configuration.

     The options provided by this control are a subset of those
     defined by the object secyCipherSuiteCapability.
     If secyCipherSuiteCapability has the integrity bit on,the integrity
     bit can be turned on for this object.
     If secyCipherSuiteCapability has the integrity and confidentiality
     bits on, the confidentiality bit of this object can be turned on
     and the integrity bit must be on.
     If secyCipherSuiteCapability has the integrity and
     offsetConfidentiality bits on, the offsetConfidentiality bit can be
```

```
     turned on and the integrity bit must be on.

     integrity(0) : on or off the function of supporting integrity
                    protection for this cipher suite.

     confidentiality(1) : on or off the function of supporting
                    confidentiality for this cipher suite.

     offsetConfidentiality(2) : on or off the function of supporting
                    offset confidentiality for this cipher suite."
    REFERENCE   "IEEE 802.1AE Clause 10.7.25"
    DEFVAL { { integrity } }
    ::= { secyCipherSuiteEntry 5 }

secyCipherSuiteProtectionOffset    OBJECT-TYPE
    SYNTAX      Integer32 (0 | 30 | 50)
    UNITS       "bytes"
    MAX-ACCESS  read-create
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION
    "The confidentiality protection offset options of this cipher suite.
     Options should depend on the choice of secyCipherSuiteProtection.
     If the value of secyCipherSuiteProtection only turns on integrity
     bit, users can only choose 0 byte for this object.
     If the value of secyCipherSuiteProtection only turns on integrity
     and confidentiality bits, users can only choose 0 byte for this
     object.
     If the value of secyCipherSuiteProtection only turns on integrity
     and offsetConfidentiality bits, users can choose 30 or 50 bytes for
     this object.
     If the value of secyCipherSuiteProtection turns on integrity and
     confidentiality and offsetConfidentiality bits, users can choose 0
     or 30 or 50 bytes for this object."
    REFERENCE   "IEEE 802.1AE Clause 10.7.25, 10.7.26"
    DEFVAL { 0 }
    ::= { secyCipherSuiteEntry 6 }

secyCipherSuiteDataLengthChange    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION "True if cipher suite changes the length of the data."
    REFERENCE   "IEEE 802.1AE Clause 10.7.25, Figure 9-1"
    ::= { secyCipherSuiteEntry 7 }

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   "IEEE 802.1AE Clause 10.7.25, Figure 9-1"
    ::= { secyCipherSuiteEntry 8 }

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   "IEEE 802.1AE Clause 10.7.25"
    ::= { secyCipherSuiteEntry 9 }

--
-- SecY Interface Ciphers Table : 802.1AEcg
--
```

```
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   "IEEE 802.1AE Clause 10.7.26, Table 13-1"
    ::= { secyMgmtMIBObjects 7 }

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

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   "IEEE 802.1AE Clause 10.7.26"
    DEFVAL { true }
    ::= { secyIfCipherEntry 1 }

secyIfCipherEnableUse    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "Enables use of the Cipher Suite by this SecY."
    REFERENCE   "IEEE 802.1AE Clause 10.7.26"
    DEFVAL { true }
    ::= { secyIfCipherEntry 2 }

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   "IEEE 802.1AE Clause 10.7.26"
    DEFVAL { true }
    ::= { secyIfCipherEntry 3 }

--
-- SecY Interface Traffic Class Table : 802.1AEcg
--

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
        shall be stored in persistent memory and remain unchanged across
        a re-initialization of the system's management entity."
```

```
    REFERENCE   "IEEE 802.1AE Clause 10.5.1, 10.7.17, Table 13-1"
    ::= { secyMgmtMIBObjects 8 }

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 ::= SEQUENCE {
    secyIfTCUserPriority          Integer32,
    secyIfTCTrafficClass          Integer32
}

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   "IEEE 802.1AE Clause 10.7.17"
    ::= { secyIfTCEntry 1 }

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   "IEEE 802.1AE Clause 10.7.17"
    DEFVAL { 0 }
    ::= { secyIfTCEntry 2 }

--
-- SecY Interface Access Priority Table : 802.1AEcg
--

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
        shall be stored in persistent memory and remain unchanged across
        a re-initialization of the system's management entity."
    REFERENCE   "IEEE 802.1AE Clause 10.5.1, 10.7.17, Table 13-1"
    ::= { secyMgmtMIBObjects 9 }

secyIfAPEntry   OBJECT-TYPE
    SYNTAX      SecyIfAPEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table entry selecting the  Access Priority Code Point
                 for a given SecY and User Priority Code Point."
    INDEX   { secyIfInterfaceIndex, secyIfAPUserPCP }
    ::= { secyIfAPTable 1 }

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

secyIfAPUserPCP OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION "A User Priority Code Point."
    REFERENCE   "IEEE 802.1AE Clause 10.5, 10.7.17"
    ::= { secyIfAPEntry 1 }

secyIfAPAccessPCP    OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The Access Priority Code Point for this SecY and User
                 PCP. Defaults to the User PCP value. "
    REFERENCE   "IEEE 802.1AE Clause 10.5, 10.7.17"
    ::= { secyIfAPEntry 2 }

--
-- TX SA Statistics : systems not supporting traffic class SCs
--

secyTxSAStatsTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSAStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "A table of statistics for each transmit SA for each of
                 the system's SecYs."
    REFERENCE   "IEEE 802.1AE Clause 10.7.18, figure 10-4"
    ::= { secyStatsMIBObjects 1 }

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 will be stop incrementing.

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

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

secyTxSAStatsProtectedPkts    OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "The number of integrity protected but not encrypted
                 packets for this transmit SA. Zero if
                 secyTxSAConfidentiality is True, and one less than
                 secyTxSANextPN otherwise."
    REFERENCE   "IEEE 802.1AE Clause 10.7.18, figure 10-4"
    ::= { secyTxSAStatsEntry 1 }

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


--
-- TX SC Statistics : systems not supporting traffic class SCs
```

```
--

secyTxSCStatsTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF SecyTxSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of statistics for each SecY's transmit SC."
    REFERENCE   "IEEE 802.1AE Clause 10.7.18, 10.7.19, Figure 10-3"
    ::= { secyStatsMIBObjects 2 }

secyTxSCStatsEntry    OBJECT-TYPE
    SYNTAX      SecyTxSCStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "An entry containing counts for a transmit 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 { secyTxSCEntry }
    ::= { secyTxSCStatsTable 1 }

SecyTxSCStatsEntry ::= SEQUENCE {
    secyTxSCStatsProtectedPkts      Counter64,
    secyTxSCStatsEncryptedPkts      Counter64,
    secyTxSCStatsOctetsProtected    Counter64, -- deprecated
    secyTxSCStatsOctetsEncrypted    Counter64  -- deprecated
}

secyTxSCStatsProtectedPkts    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   "IEEE 802.1AE Clause 10.7.18, Figure 10-3"
    ::= { secyTxSCStatsEntry 1 }

secyTxSCStatsEncryptedPkts    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   "IEEE 802.1AE Clause 10.7.18, Figure 10-3"
    ::= { secyTxSCStatsEntry 4 }

secyTxSCStatsOctetsProtected    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 but not encrypted for this transmit SC."
    REFERENCE   "IEEE 802.1AE Clause 10.7.19, Figure 10-3"
    ::= { secyTxSCStatsEntry 10 }

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   "IEEE 802.1AE Clause 10.7.19, Figure 10-3"
    ::= { secyTxSCStatsEntry 11 }
--
-- Traffic Class capable transmit SC Statistics : 802.1AEcg
--
```

```
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   "IEEE 802.1AE Clause 10.7.18, 10.7.19, Figure 10-3"
    ::= { secyStatsMIBObjects 12 }

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 ::= SEQUENCE {
    secyTSCStatsProtectedPkts      Counter64,
    secyTSCStatsEncryptedPkts      Counter64
}

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   "IEEE 802.1AE Clause 10.7.18, Figure 10-3"
    ::= { secyTSCStatsEntry 1 }

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   "IEEE 802.1AE Clause 10.7.18, Figure 10-3"
    ::= { secyTSCStatsEntry 2 }


--
-- RX SA Statistics Information
--

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsMIBObjects 3 }

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 will be stop incrementing.
```

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

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

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 1 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 4 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 13 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 16 }

secyRxSAStatsOKPkts          OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "For this SA, the number of validated packets."
    REFERENCE   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSAStatsEntry 25 }
```

```
--
-- RX SC Statistics Information
--

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   "IEEE 802.1AE Clause 10.7.9, 10.7.10, Figure 10-4"
    ::= { secyStatsMIBObjects 4 }

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 ::= 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    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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 1 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 2 }

secyRxSCStatsLatePkts        OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of packets discarded, for this SC, because the
         the received PN was lower than the lowest acceptable PN
         (secyRxSALowestXPN) and secyIfReplayProtectEnable was true."
    REFERENCE   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 3 }

secyRxSCStatsNotValidPkts     OBJECT-TYPE
    SYNTAX      Counter64
```

```
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The number of packets discarded, for this SC, because validation
        failed and secyIfvalidateFrames was 'strict' or the data was
        encrypted (so the original frame could not be recovered)."
    REFERENCE   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 4 }

secyRxSCStatsInvalidPkts     OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The number of packets, for this SC, that failed validation but
       could be received because secyIfvalidateFrames  was 'check' and
       the  data was not encrypted (so the original frame could be
       recovered)."
    REFERENCE   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 5 }

secyRxSCStatsDelayedPkts     OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The number of received packets, for this SC, with PN lower
        than the lowest acceptable PN (secyRxSALowestXPN) and
        secyIfReplayProtectEnable false."
    REFERENCE   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 6 }

secyRxSCStatsUncheckedPkts   OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of packets received for this SC,  while
                 secyValidateFrames was 'disabled'."
    REFERENCE   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 7 }

secyRxSCStatsOKPkts          OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of packets received for this SC
                 successfully validated and within the replay window."
    REFERENCE   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyRxSCStatsEntry 8 }

secyRxSCStatsOctetsValidated     OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "The number 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    OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "Octets"
    MAX-ACCESS  read-only
    STATUS      deprecated -- 802.1AEcg
    DESCRIPTION "The number 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 }

--
-- SecY statistics table
--

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   "IEEE 802.1AE Clause 10.7.9, 10.7.18, Figure 10-3, 10.5"
    ::= { secyStatsMIBObjects 5 }

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

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    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   "IEEE 802.1AE Clause 10.7.18, Figure 10-3"
    ::= { secyStatsEntry 1 }

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   "IEEE 802.1AE Clause 10.7.18, Figure 10-3"
    ::= { secyStatsEntry 2 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 3 }
```

```
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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 4 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 5 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 6 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 7 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 8 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 9 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 10 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 11 }

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   "IEEE 802.1AE Clause 10.7.9, Figure 10-4"
    ::= { secyStatsEntry 12 }

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   "IEEE 802.1AE Clause 10.6.3, Figure 10-3"
    ::= { secyStatsEntry 13 }

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   "IEEE 802.1AE Clause 10.6.3, Figure 10-3"
    ::= { secyStatsEntry 14 }
--
-- Conformance
--

secyMIBCompliances  OBJECT IDENTIFIER ::= { secyMIBConformance 1 }

secyMIBGroups  OBJECT IDENTIFIER ::= { secyMIBConformance 2 }

-- Compliance

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 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 }
--
-- Units of Conformance
-- Controlled Port service management MIB Groups

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 --- 802.1AEcg, updates secyIfCtrlGroup
    DESCRIPTION "SecY service management (secyIfTable objects) for
                 systems supporting traffic class SCs."
    ::= { secyMIBGroups 12 }

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

secyIfAPGroup    OBJECT-GROUP
    OBJECTS {
        secyIfAPAccessPCP
    }
    STATUS      current --- 802.1AEcg
    DESCRIPTION "Access Priority Code Point control (secyIfAPTable)."
    ::= { secyMIBGroups 15 }

-- Transmit SC and SA MIB Groups

secyTSCGroup    OBJECT-GROUP
    OBJECTS {
        secyTSCState,
        secyTSCEncodingSA,
        secyTSCCreatedTime,
        secyTSCStartedTime,
        secyTSCStoppedTime
    }
    STATUS      current --- 802.1AEcg, updates secyTxSCGroup
    DESCRIPTION "Transmit SC management (secyTSCTable objects) for
                 systems supporting traffic class SCs."
```

```
    ::= { secyMIBGroups 16 }

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    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    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 }

-- Receive SC and SA MIB Groups

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

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

```
secyRSAGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSAState,
        secyRxSANextXPN,
        secyRxSALowestXPN,
        secyRxSAKeyIdentifier,
        secyRxSASSCI,
        secyRxSACreatedTime,
        secyRxSAStartedTime,
        secyRxSAStoppedTime
    }
    STATUS      current --- 802.1AEcg, updates secyRxSAGroup
    DESCRIPTION "Receive SA (secyRxSATable objects)."
    ::= { secyMIBGroups 19 }

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    OBJECT-GROUP
    OBJECTS {
        secyCipherSuiteId,
        secyCipherSuiteName,
        secyCipherSuiteCapability,
        secyCipherSuiteDataLengthChange,
        secyCipherSuiteICVLength
    }
    STATUS      current --- 802.1AEcg, updates secyCipherSuiteGroup
    DESCRIPTION "Cipher Suite implementation information
                 (secyCipherSuiteTable objects)."
    ::= { secyMIBGroups 21 }

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

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

secyCipherStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyStatsTxOctetsProtected,
```

```
        secyStatsTxOctetsEncrypted,
        secyStatsRxOctetsValidated,
        secyStatsRxOctetsDecrypted
    }
    STATUS      current --- 802.1AEcg
    DESCRIPTION
        "Cipher Suite performance statistics (from secyStatsTable)."
    ::= { secyMIBGroups 24 }

-- Transmit and Receive SA and SC statistics MIB Groups

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

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

secyTSCStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyTSCStatsProtectedPkts,
        secyTSCStatsEncryptedPkts
    }
    STATUS      current --- 802.1AEcg, updates secyTxSCStatsGroup
    DESCRIPTION "Transmit SC statistics (secyTSCStatsTable objects)."
    ::= { secyMIBGroups 22 }

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

secyRSCStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSCStatsLatePkts,
        secyRxSCStatsNotValidPkts,
        secyRxSCStatsInvalidPkts,
        secyRxSCStatsDelayedPkts,
        secyRxSCStatsUncheckedPkts,
        secyRxSCStatsOKPkts
    }
    STATUS      current --- 802.1AEcg, updates secyRxSCStatsGroup
    DESCRIPTION "Receive SC statistics (secyRxSCStatsTable objects)."
    ::= { secyMIBGroups 23 }

secyRxSCStatsGroup    OBJECT-GROUP
    OBJECTS {
        secyRxSCStatsUnusedSAPkts,
        secyRxSCStatsNoUsingSAPkts,
        secyRxSCStatsLatePkts,
        secyRxSCStatsNotValidPkts,
        secyRxSCStatsInvalidPkts,
```

```
        secyRxSCStatsDelayedPkts,
        secyRxSCStatsUncheckedPkts,
        secyRxSCStatsOKPkts,
        secyRxSCStatsOctetsValidated,
        secyRxSCStatsOctetsDecrypted
    }
    STATUS      deprecated
    DESCRIPTION
        "Receive SC statistics objects."
    ::= { secyMIBGroups 10 }

-- Controlled Port service statistics MIB Groups

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

END
```

14. Cipher Suites

A Cipher Suite is an interoperable specification of cryptographic algorithms together with the values of parameters (for example, key size) to be used by those algorithms. Specification of the cryptographic functions required by MACsec in terms of Cipher Suites increases interoperability by providing a clear default and a limited number of alternatives.

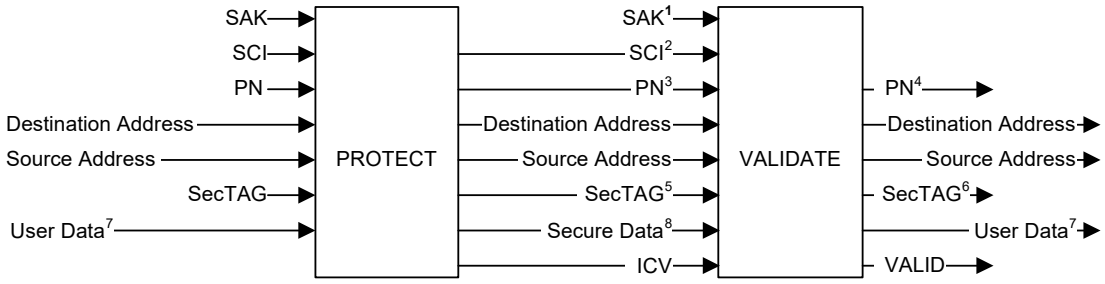
This clause specifies

a) Terms that describe the use of each Cipher Suite by the MAC Security Entity (SecY)
b) Capabilities required of each Cipher Suite
c) Requirements this standard places on Cipher Suite specification
d) Mandatory and optional Cipher Suites for use in conjunction with this standard
e) Criteria for the use of additional Cipher Suites in conjunction with MACsec for implementations for which a claim of conformance to this standard is made.

NOTE—The choice and combination of cryptographic methods is notorious for the introduction of unexpected security exposures. Each Cipher Suite uses an algorithm or combination of algorithms whose interactions have been studied by the professional security community. Each Cipher Suite specification (14.5–14.8) in this clause comprises the necessary combination (e.g., concatenation of named strings) and mapping of parameters and parameter names used in the other clauses of this standard to the parameters and parameter names used by a public established standard that specifies the cryptographic operations.

14.1 Cipher Suite use

A Cipher Suite is initialized with one or more Cipher Suite dependent keys, and then used to protect protocol parameters. Any implementation of the same Cipher Suite, initialized with the same key values, can be used to validate and recover the protected parameters. The protect and validate operations are illustrated in Figure 14-1, and their inputs and outputs specified after the figure.



[1] The SAK to be used on receipt of the frame is identified by the SCI and the AN.

[2] The SCI is extracted from the SCI field of the SecTAG if present. A value conveyed by key agreement (point-to-point only) is used otherwise.
In the GCM-AES-128 and GCM-AES-256 Cipher Suites (14.5, 14.6), the SCI is always included in the IV parameter whether included in the SecTAG or not (and thus always contributes to the ICV). However the Cipher Suite parameter A includes the SCI if and only if the SCI is included in the SecTAG.
In the GCM-AES-XPN-128 and GCM-AES-XPN-256 Cipher Suites (14.7, 14.8), the {SCI, SAK} tuple (or equivalently the SA) identifies the SSCI (conveyed by key agreement) that is included in the IV parameter, and the Cipher Suite parameter A includes the SCI if and only if the SCI is included in the SecTAG.

[3] The 32 least significant bits of the PN are conveyed in the SecTAG

[4] The validated PN can be used for replay protection.

[5] All the transmitted octets of the SecTAG are protected, including the optional SCI field if present

[6] The validated received SecTAG contains bits of the TCI, and optionally the SCI, these can be used for service multiplexing (11.7).

[7] The length, in octets, of the User Data is conveyed by the User Data parameter, and is protected by Cipher Suite operation.

[8] The length, in octets, of the Secure Data is conveyed by the MACsec frame, unless it is short, when it is conveyed by the SL parameter in the SecTAG TCI

Figure 14-1—Cipher Suite Protect and Validate operations

Protect	(	SAK, SCI, PN, Destination Address, Source Address, SecTAG, User Data	Validate	(	SAK, SCI, PN, Destination Address, Source Address, SecTAG, Secure Data, ICV
	)	Secure Data, ICV		)	User Data, VALID

The SAK (Secure Association Key, 7.1) is the value of the Cipher Suite dependent key(s).

The SCI (Secure Channel Identifier, 7.1.2) is a 64-bit identifier that is globally unique amongst all correctly configured Cipher Suite implementation instances protecting MACsec protocol parameters.

The PN (Packet Number, 8.3) is a number that is never zero, is incremented each time a protect request is made for a given SCI, and is never repeated for an SCI unless the SAK is changed. The size of the PN depends on the Cipher Suite, and is 32 bits unless otherwise specified. Cipher Suites that provide extended packet numbering use a 64-bit PN. Irrespective of the size of the PN, only the least significant 32 bits are conveyed in the SecTAG. If extended packet numbering is used, the most significant 32 bits are recovered for each received frame as specified in 10.6.2.

The Destination Address and Source Address are the MAC addresses of the frame. MAC Addresses are specified as octet strings, using the canonical format specified in IEEE Std 802.

The SecTAG is as specified in Clause 9.

The ICV (integrity check value, 8.3) is a string of octets. VALID is a Boolean parameter. If TRUE the validation was successful.

Given the SAK, SCI, PN, Source Address, Destination Address, SecTAG, and the User Data, the Protect operation returns the Secure Data and ICV.

Given the same SAK, SCI, PN, Source Address, Destination Address, and SecTAG, and the Secure Data and ICV, the Verify operation returns the original User Data and VALID. If any of the parameters were modified, VALID is returned False.

14.2 Cipher Suite capabilities

Any Cipher Suite used with MACsec shall

a) Provide integrity protection for the SCI, PN, Source Address, Destination Address, SecTAG, and from 0 through 2^{16}– 1 octets of User Data on each invocation.
b) Provide integrity and confidentiality (if specified) for at least 2^{32}– 1 invocations, each with a different PN, without requiring a fresh SAK.
c) Given any specific number of octets of User Data, generate a predictable number of octets of Secure Data and ICV.

and may

d) Provide confidentiality protection for all the octets of the User Data.
e) Provide confidentiality protection for all the octets of the User Data following an initial number of octets, as specified in 10.7.25.

and shall not

f) Generate Secure Data that when added to the number of octets in the ICV contains over 896 octets more than the User Data.

NOTE—A Cipher Suite may introduce additional fields into the Secure Data even if confidentiality is not provided.

g) Modify or constrain the values of the SCI, PN, Source Address, Destination Address, or SecTAG fields, other than as specified in this clause (Clause 14).

h) Require an SAK exceeding 1024 bits long (in total for all keys that compose the SAK).

i) Require different keys for the protect and validate operations.

An implementation of MACsec for which conformance to this standard is claimed includes at least one Cipher Suite that provides integrity without confidentiality, with the Secure Data the same as the User Data, and the ICV comprising 16 octets. This requirement is met by the mandatory Default Cipher Suite.

14.3 Cipher Suite specification

Each Cipher Suite specification shall comprise an interoperable specification of the protection and verification procedures in terms of the parameters specified in 14.1 and shall state

a) Whether confidentiality of the User Data is provided

b) The maximum difference in the lengths of the User Data and Secure Data

c) The length of the ICV

d) The length and properties of the keys required, including assumptions of the scope of uniqueness.

NOTE—While this standard provides definitive specifications of the Cipher Suites that support full conformance, those specifications make the greatest possible use of other public and established standards, and are principally concerned with ensuring unambiguous application of those standards in the context of MACsec.

14.4 Cipher Suite conformance

An implementation of MACsec that claims full conformance to this standard shall implement the mandatory Cipher Suites in Table 14-1, may implement one or more of the Optional Cipher Suites in the table, and shall not implement any other Cipher Suite. Every conformant implementation shall include at least one Cipher Suite that does not encrypt User Data.

Table 14-1 assigns a Cipher Suite reference number for use in protocol identification within a MACsec context, provides a short name for use in this standard, indicates the type of cryptographic algorithm used and the security services provided, specifies whether the Cipher Suite is mandatory or optional for conformance to this standard, and references the clause of this standard that provides the definitive description of the Cipher Suite.

NOTE—In IEEE Std 802.1AE-2006 (the first edition of this standard), the Cipher Suite Identifier for GCM-AES-128 was incorrectly shown as 00-80-02-00-01-00-00-01 in Table 14-1. Prior to the inclusion of GCM-AES-256, GCM-AES-128 was the only conformant Cipher Suite. IEEE Std 802.1X uses a reserved encoding for the Default Cipher Suite rather than the Cipher Suite Identifier to identify GCM-AES-128.

14.4.1 Conformance with Cipher Suite variance

An implementation of MACsec that claims conformance to this standard with Cipher Suite variance, shall implement the mandatory Cipher Suites in Table 14-1, may implement one or more of the optional Cipher Suites in Table 14-1, and may implement alternate Cipher Suites that meet the requirements of 14.2 and 14.3, and the following guidelines, and shall not implement any other Cipher Suite, or other combination of cryptographic algorithms and parameters.

Table 14-1—MACsec Cipher Suites

Cipher Suite Identifier	Cipher Suite Name	Services provided		Mandatory/Optional	Defining Clause
		Integrity without confidentiality	Integrity and confidentiality		
00-80-C2-00-01-00-00-01	GCM-AES-128	Yes	Yes	Mandatory	14.5
00-80-C2-00-01-00-00-02	GCM-AES-256	Yes	Yes	Optional	14.6
00-80-C2-00-01-00-00-03	GCM-AES-XPN-128	Yes	Yes	Optional	14.7
00-80-C2-00-01-00-00-04	GCM-AES-XPN-256	Yes	Yes	Optional	14.8

The use of additional Cipher Suites shall meet the following guidelines:

a) Algorithms chosen have an effective key length of at least 128 bits. In schemes built on block ciphers, the underlying block cipher has a block width of at least 128 bits.

b) If serviced by separate algorithms, the properties of the authentication and confidentiality mechanisms are combinable in accordance with well-established security results. Either the encryption happens before authentication, or the encryption is performed through keystream generation.

c) Either of the following holds true:

   1) The underlying cryptographic cipher is approved by either a national or international standards body or a government agency; or

   2) The following conditions i) through iv) apply:

      i) The Cipher Suite provides message authentication using a message authentication algorithm with a publicly available proof of security against forgery attacks, even in a model where the attacker has the ability to choose messages for the sender.

      ii) If confidentiality is provided, the confidentiality mechanism has a publicly available proof of security in a model where the attacker has the ability to adaptively choose both plaintext and cipher text.

      iii) Mechanisms for confidentiality and message authentication are used in a way that is consistent with their proof of security. For example, if using the Cipher Block Chaining (AES-CBC) mode of operation the IV is performed through keystream generation.

      iv) Mechanisms for confidentiality and message authentication are used in a way that is consistent with their proof of security. For instance, if using the Cipher Block Chaining (AES-CBC) mode of operation, the IV is randomly selected with each message, and not sequentially.

14.5 Default Cipher Suite (GCM-AES-128)

The Default Cipher Suite uses the Galois/Counter Mode of Operation with the AES-128 symmetric block cipher, as specified in this clause by reference to the terms *K, IV, A, P, C, T* used in NIST SP 800-38D.

K is the 128 bit SAK. The 64 most significant bits of the 96-bit *IV* are the octets of the SCI, encoded as a binary number (9.1). The 32 least significant bits of the 96-bit *IV* are the octets of the PN, encoded as a binary number (9.1). *T* is the ICV and is 128 bits long. When the bit-strings *A*, *P*, and *C* are specified in terms of octet strings, earlier octets compose earlier bits, and more significant bits in each octet are earlier.

NOTE 1—IETF RFC 5116 [B9], McGrew [B11], and McGrew and Viega [B12] provide additional information about GCM and its security properties and use.

NOTE 2—The bit strings obtained by transforming MAC Address and data octets using these rules do not correspond to IEEE 802.3 “wire order” for frame transmission.

When the Default Cipher Suite is used for Integrity Protection

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG and User Data concatenated in that order.
- *P* is null.
- The Secure Data is the octets of the User Data, without modification.

When the Default Cipher Suite is used for Confidentiality Protection without a confidentiality offset

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG concatenated in that order.
- *P* is the octets of the User Data.
- The Secure Data is *C*.

When the Default Cipher Suite is used for Confidentiality Protection with a confidentiality offset

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG and the first confidentialityOffset (10.7.27) octets of the User Data (or all the octets of the User Data if that comprises fewer than confidentialityOffset octets) concatenated in that order.
- *P* is the remaining octets (if any) of the User Data.
- The Secure Data is the first confidentialityOffset octets of the User Data concatenated with *C*, in that order (or all the octets of the User Data if that comprises fewer than confidentialityOffset octets).

NOTE 3—IEEE Std 802.1AE-2006 specified the confidentiality offset option to facilitate early MACsec deployment on systems that needed to examine the initial octets of IP version 4 or version 6 frames to decide where to store received frames, before decrypting the frame. The XPN Cipher Suites standardized in IEEE Std 802.1AEbw-2013 do not support confidentiality offsets.

14.6 GCM-AES-256

GCM-AES-256 uses the Galois/Counter Mode of operation with the AES-256 symmetric block cipher, as specified in this clause by reference to the terms *K, IV, A, P, C, T* used in NIST SP 800-38D.

K is the 256 bit SAK. The 64 most significant bits of the 96-bit *IV* are the octets of the SCI, encoded as a binary number (9.1). The 32 least significant bits of the 96-bit *IV* are the octets of the PN, encoded as a binary number (9.1). *T* is the ICV, and is 128 bits long. When the bit-strings *A*, *P*, and *C* are specified in terms of octet strings, earlier octets compose earlier bits, and more significant bits in each octet are earlier.

NOTE 1—The bit strings obtained by transforming MAC Address and data octets using these rules do not correspond to IEEE 802.3 “wire order” for frame transmission.

When this Cipher Suite is used for Integrity Protection

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG and User Data concatenated in that order.
- *P* is null.
- The Secure Data is the octets of the User Data, without modification.

When this Cipher Suite is used for Confidentiality Protection without a confidentiality offset

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG concatenated in that order.
- *P* is the octets of the User Data.
- The Secure Data is *C*.

When this Cipher Suite is used for Confidentiality Protection with a confidentiality offset

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG and the first confidentialityOffset (10.7.27) octets of the User Data (or all the octets of the User Data if that comprises fewer than confidentialityOffset octets) concatenated in that order.
- *P* is the remaining octets (if any) of the User Data.
- The Secure Data is the first confidentialityOffset octets of the User Data concatenated with *C*, in that order (or all the octets of the User Data if that comprises fewer than confidentialityOffset octets).

NOTE 2—IEEE Std 802.1AE-2006 specified the confidentiality offset option to facilitate early MACsec deployment on systems that needed to examine the initial octets of IP version 4 or version 6 frames to decide where to store received frames, before decrypting the frame. The XPN Cipher Suites standardized in IEEE Std 802.1AEbw-2013 do not support confidentiality offsets.

14.7 GCM-AES-XPN-128

Each instance of the GCM-AES-XPN-128 Cipher Suite, i.e., the protection and validation capabilities created for a given SAK at the request of the KaY (10.7.26, Figure 10-5) maintains an instance of the following parameter as specified in 10.7.26:

a) Salt, a 96-bit value distributed by key agreement protocol to all members of the CA.

and an instance of the following parameter for each SCI, as supplied by the KaY when an SA that uses the SCI and the given SAK is created (10.7.13, 10.7.21):

b) SSCI, a 32-bit value that is unique for each SCI using a given SAK.

NOTE 1—The maximum number of SSCIs for a given SAK is thus limited by the maximum number of SCIs (equivalently, by the maximum number of simultaneous members in a CA as requirements placed on the KaY (8.2.7) prohibit the use of the same SAK in multiple CAs). A claim of conformance to this standard requires a statement of the maximum number of receive SCs supported (5.3m, A.5, A.12, A.13). The total number of SCIs will be one greater (to include the transmit SC) or two greater [for an EPON OLT supporting an SCB (Clause 12)]. Whether and to what extent the same SAK is used by different SAs (each with a different SCI, and hence a different SSCI for that SAK) depends on the key agreement protocol, and the number of members in a CA will also be ultimately limited by the capabilities of the key agreement protocol. The practical requirements of the port-based network control application (see Clause 7 of IEEE Std 802.1X-2010) are likely to be more limited.

GCM-AES-XPN-128 uses the Galois/Counter Mode of operation with the AES-128 symmetric block cipher, as specified in this clause by reference to the terms *K, IV, A, P, C, T* used in NIST SP 800-38D.

K is the 128-bit SAK. The 32 most significant bits of the 96-bit *IV* are the octets of the SSCI for the SCI, encoded as a binary number (9.1) and exclusive-or'd with the 32 most significant bits of the Salt. The 64 least significant bits of the 96-bit *IV* are the octets of the PN, encoded as a binary number (9.1) and exclusive-or'd with the 64 least significant bits of the Salt. *T* is the ICV, and is 128 bits long. When the bit-strings *A*, *P*, and *C* are specified in terms of octet strings, earlier octets compose earlier bits, and more significant bits in each octet are earlier.

NOTE 2—The bit strings obtained by transforming MAC Address and data octets using these rules do not correspond to IEEE 802.3 "wire order" for frame transmission.

When this Cipher Suite is used for Integrity Protection

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG and User Data concatenated in that order.
- *P* is null.
- The Secure Data is the octets of the User Data, without modification.

When this Cipher Suite is used for Confidentiality Protection without a confidentiality offset

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG concatenated in that order.
- *P* is the octets of the User Data.
- The Secure Data is *C*.

This Cipher Suite does not provide Confidentiality Protection with a confidentiality offset.

14.8 GCM-AES-XPN-256

Each instance of the GCM-AES-XPN-256 Cipher Suite, i.e., the protection and validation capabilities created for a given SAK at the request of the KaY (10.7.26, Figure 10-5) maintains an instance of the following parameter as specified in 10.7.26:

a) Salt, a 96-bit value distributed by key agreement protocol to all members of the CA.

and an instance of the following parameter for each SCI, as supplied by the KaY when an SA that uses the SCI and the given SAK is created (10.7.13, 10.7.21):

b) SSCI, a 32-bit value that is unique for each SCI using a given SAK.

NOTE 1—The maximum number of SSCIs for a given SAK is thus limited by the maximum number of SCIs (equivalently, by the maximum number of simultaneous members in a CA as requirements placed on the KaY (8.2.7) prohibit the use of the same SAK in multiple CAs). A claim of conformance to this standard requires a statement of the maximum number of receive SCs supported (5.3m, A.5, A.12, A.13). The total number of SCIs will be one greater (to include the transmit SC) or two greater [for an EPON OLT supporting an SCB (Clause 12)]. Whether and to what extent the same SAK is used by different SAs (each with a different SCI, and hence a different SSCI for that SAK) depends on the key agreement protocol, and the number of members in a CA will also be ultimately limited by the capabilities of the key agreement protocol. The practical requirements of the port-based network control application (see Clause 7 of IEEE Std 802.1X-2010) are likely to be more limited.

GCM-AES-XPN-256 uses the Galois/Counter Mode of operation with the AES-256 symmetric block cipher, as specified in this clause by reference to the terms *K, IV, A, P, C, T* used in NIST SP 800-38D.

K is the 256-bit SAK. The 32 most significant bits of the 96-bit *IV* are the octets of the SSCI for the SCI, encoded as a binary number (9.1) and exclusive-or'd with the 32 most significant bits of the Salt. The 64 least significant bits of the 96-bit *IV* are the octets of the PN, encoded as a binary number (9.1), and exclusive-or'd with the 64 least significant bits of the Salt. *T* is the ICV, and is 128 bits long. When the bit-strings *A*, *P*, and *C* are specified in terms of octet strings, earlier octets compose earlier bits, and more significant bits in each octet are earlier.

NOTE 2—The bit strings obtained by transforming MAC Address and data octets using these rules do not correspond to IEEE 802.3 "wire order" for frame transmission.

When this Cipher Suite is used for Integrity Protection

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG and User Data concatenated in that order.
- *P* is null.
- The Secure Data is the octets of the User Data, without modification.

When this Cipher Suite is used for Confidentiality Protection without a confidentiality offset

- *A* is the Destination MAC Address, Source MAC Address, and the octets of the SecTAG concatenated in that order.
- *P* is the octets of the User Data.
- The Secure Data is *C*.

This Cipher Suite does not provide Confidentiality Protection with a confidentiality offset.

15. Ethernet Data Encryption devices

An Ethernet Data Encryption device (EDE) is a frame forwarding device with two physical ports that uses IEEE Std 802.1Q, IEEE Std 802.1AE, and IEEE Std 802.1X to provide integrity and confidentiality for frames forwarded on network hops open to attack. One port (red side) receives and transmits unprotected frames, while frames transmitted and received on the other (black side) are protected by MACsec.

This clause describes and specifies the following:

a) The common characteristics of EDEs—the rationale for identifying some of the many possible MACsec capable bridging systems as EDEs—and provides an EDE taxonomy (15.1).
b) How connectivity between adjacent bridges in a customer bridged network or a Provider Bridged Network (PBN) can be secured by an EDE-M (15.2) comprising a two-port VLAN-unaware MAC Bridge with a MAC Security Entity (SecY) supporting one port.
c) Requirements for securing connectivity across a PBN (15.3).
d) How connectivity across a PBN can be secured by an EDE-M (15.4), or by an EDE-CS (15.5), EDE-CC (15.6), or EDE-SS (15.7)—each comprising two VLAN bridge components, and each using MACsec to preserve frame data integrity, data origin authenticity, and confidentiality, while allowing the provider to use the frame's VLAN tag to perform service selection and to convey Priority Code Point (PCP) and drop-eligible (DEI) information.
e) Interoperability between EDEs and other MACsec-capable bridging systems (15.8).
f) Considerations applicable to User Network Interface (UNI) access and Connectivity Fault Management (CFM) use when EDEs are used (15.9).

An understanding of architectural concepts common to this and other IEEE 802.1 standards is essential to understanding this standard's specification of EDEs. The reader is encouraged to review Clause 7 of IEEE Std 802.1AC-2016 and IEEE Std 802.1Q's use of bridge components in its specification of Provider Edge Bridges (PEBs), Backbone Edge Bridges (BEBs), PBNs, and provider network service interfaces (see, in particular, Clause 15, Clause 16, and Clause 25 of IEEE Std 802.1Q-2018).

15.1 EDE characteristics

The specification of EDEs arises from the desire to separate, so far as is possible, the implementation and use of MAC security within bridged networks from other implementation and management concerns. Reduction of the scope (in terms both of quantity and variety) of the functionality co-resident with the implementation and management of MACsec and its associated authentication, authorization, and key agreement functions can have two benefits. First, there is less software to validate, and fewer people and organizations might have to be involved in the development of each EDE. Second, it might be possible to assign management responsibility for EDEs and other bridging systems to separate smaller administrative organizations, each with its own particular expertise. Against these benefits are to be set the costs arising from additional items of equipment and the operational coordination necessary. This standard does not attempt to judge the balance of these benefits and costs, which are implementation and deployment specific. In particular there is no suggestion that the specification of EDEs means that these are always preferred to the use of other bridging systems specified in IEEE Std 802.1Q with MACsec supporting particular ports.

Restricting an EDE to having two and only two physical ports reduces the requirement for traffic class processing, particularly if the ports operate at close to the same data rate. A two-port EDE might also have no need to appear as a node in certain protocols at all, with frames for those protocols (or specific instances of those protocols as identified by destination group MAC address) being relayed simply from one port to another. Since other types of bridging systems will be usually attached to an EDE, there is no need to learn from the source address of frames.

NOTE 1—IEEE Std 802.1Q provides for some functional simplification in two-port systems or components. Provider Edge Bridges can simply forward frames addressed to the Nearest Customer Bridge Address if the C-VLAN component of a PEB has a single Provider Edge Port (PEP), i.e., it provides connectivity to a single provider network service instance (stated in terms of the equivalent condition of connecting to the S-VLAN component through a single Customer Network Port in 13.41 of IEEE Std 802.1Q-2018). If the enhanced filtering utility criteria (8.7.2 of IEEE Std 802.1Q-2018) can never be met (a common condition for a two-port component supporting point-to-point connectivity), no source address learning need ever occur, and the size of the Filtering Database can be restricted to that necessary to accommodate the Permanent Database.

This standard specifies various types of EDE, distinguishing them by their bridging components. An EDE-M comprises a single VLAN-unaware MAC Bridge component, an EDE-CS (both a C-VLAN and an S-VLAN component), an EDE-CC two C-VLAN components, and an EDE-SS two S-VLAN components. The architecture and use of each type is explained in 15.2 through 15.7.

NOTE 2—The first-time reader of this specification is encouraged to read 15.2 through 15.7 before attempting to understand this taxonomy and its inherent possibilities in detail. In brief, describing each type of EDE in terms of existing components (just as IEEE Std 802.1Q specifies a Provider Edge Bridge in terms of the C-VLAN and S-VLAN components that comprise Customer Bridges and Provider Bridges respectively) makes for economy of specification and simplifies analysis. Since the existing components can already be connected in a network, and any part of a valid network can be considered a valid (if large) system, no new interoperability challenges arise.

15.2 Securing LANs with EDE-Ms

In the simplest EDE network configuration, the protected (black-side) of each of a pair of EDE-Ms is attached to a single LAN providing point-to-point connectivity between the EDEs as shown in Figure 15-1.

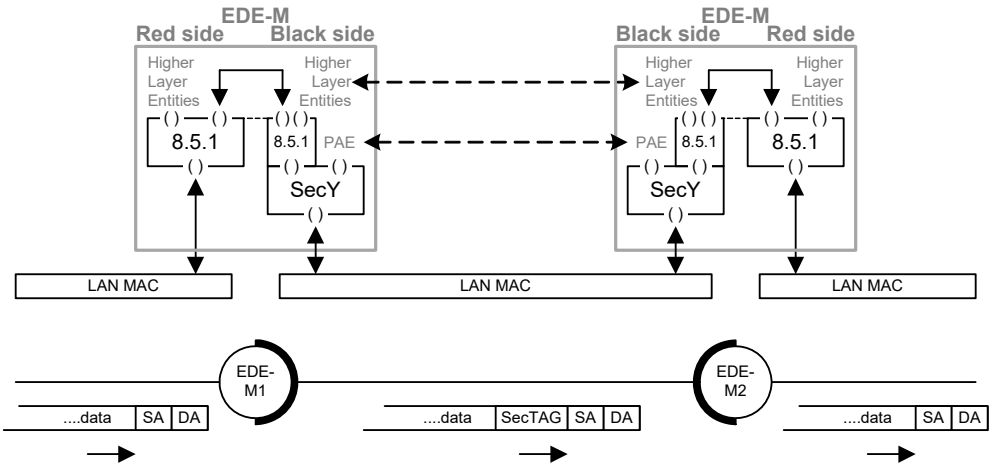


Figure 15-1—EDE-Ms connected by a point-to-point LAN

The upper part of Figure 15-1 shows the interface stacks, each attached to a LAN, in two EDE-Ms. The lower part shows the path, from the red side of one of the EDEs through to the red side of the other, and the change to a frame transmitted along that path.

NOTE 1—The architecture of a bridge is often drawn as in Figure 11-4, showing the MAC Relay entity below the MAC Service boundary to show that the relay is transparent to higher layer protocols. In this clause (Clause 15) it is convenient to use figures that focus on the interface stacks supporting relay and higher layer entities as in Figure 11-5. Numbers in the figures in this clause (e.g., 8.5.1 in Figure 15-1, 6.9 in Figure 15-2) refer to relevant clauses of IEEE Std 802.1Q-2018. The representation of EDE-M1 and EDE-M2 in Figure 15-1 provides a simple indication that frames transmitted and received by their black-side ports have had SecTAGs (and MACsec processing) applied. Similarly, other figures in this clause use light and dark patterns to indicate the possible presence of a C-VLAN or an S-VLAG tag respectively. For example, in Figure 15-3 an untagged or C-tagged frame from B1 is Sec-tagged by EDE-1, has an S-TAG added by PB1, and passes through PB2 before the outer tags are processed and removed en-route to B2. This notation can be used in larger network diagrams where showing the frame on each connecting link is inconvenient.

A pair of EDE-Ms can secure a point-to-point LAN connecting the ports of two Customer Bridges or Provider Bridges, as in Figure 15-2.

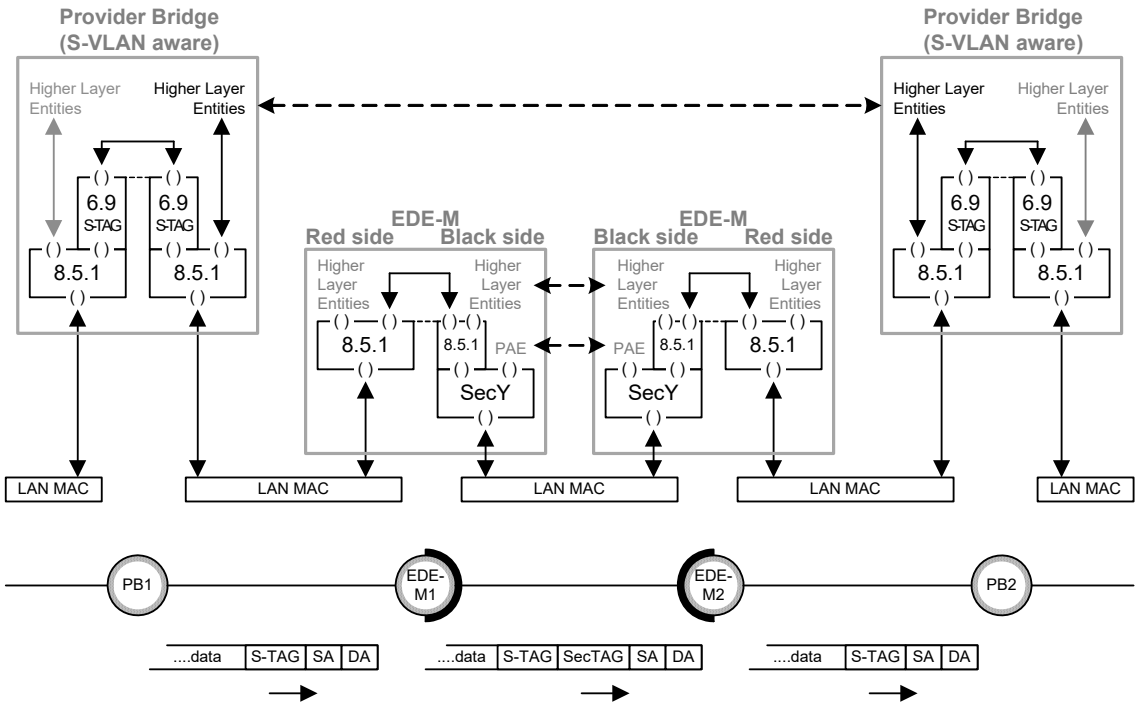


Figure 15-2—EDE-Ms securing a point-to-point LAN between Provider Bridges

Each EDE-M's PAE uses the Nearest non-TPMR group address[16] (01-80-C2-00-00-03) as the destination address for EAPOL PDUs. Frames with this destination address are filtered by all bridges specified in IEEE 802.1 standards (including EDEs) other than TPMRs. A PAE receiving a frame with this destination address can be certain that no IEEE 802.1 standards-conformant bridge lies between itself and the originator of the frame. If such a bridge is interposed between the EDE-Ms, they will not exchange EAPOL PDUs and will not, as a consequence, use MACsec to protect the frames (if any) that they forward. This avoids unintended use of the EDE-Ms in a misconfigured network. Similarly neither EDE-M shown can receive EAPOL PDUs from other EDEs connected to PB1 or PB2. This helps to prevent accidental creation of MACsec protected connectivity through (but without the participation of) an intervening bridge, and the undesirable consequence of making it impossible for that bridge to understand frames that it was intended to receive.

NOTE 2—EAPOL PDUs are used to initiate and reinitiate EAP authentication exchanges, convey EAP PDUs in support of those exchanges, and convey MACsec Key Agreement PDUs (MKPDUs). At the time of the development of IEEE Std 802.1AEcg-2017, EAPOL destination addresses were specified in 11.1 and Table 11-1 of IEEE Std 802.1X-2010. Filtering of Reserved Addresses by bridges was specified in 8.6.3, 8.13.4, and Table 8-1, Table 8-2, and Table 8-3 of IEEE Std 802.1Q-2018.

Each EDE-M also filters the addresses specified by IEEE Std 802.1Q as TMPR component Reserved Addresses. When an EDE-M's PAE is configured to use the Nearest non-TPMR group address, the other Reserved Addresses specified for MAC Bridge, C-VLAN, and S-VLAN components are forwarded, making the EDE-Ms and the connection they protect transparent to protocols using those addresses.

[16]This address was identified as the "IEEE Std 802.1X PAE address" in IEEE Std 802.1Q-2005 and as the "PAE group address" in IEEE Std 802.1X-2010.

15.3 Securing connectivity across PBNs

IEEE Std 802.1Q specifies support of the MAC Service by Provider Bridged Networks and their principles of operation. Individual instances of the MAC Service are segregated within the PBN by S-VLAN tag, and access to those instances are provided by port-based, C-tagged, or S-tagged service interfaces. A given service instance can support service interfaces of more than one type. For example, in a hub-and-spoke configuration, it might be convenient to use a C-tagged service interface at the central site but a port-based interface at the remote sites, since the latter communicate directly only with the central site. Such an arrangement avoids having to configure each remote site differently. Alternately, each remote site might be configured to use the same identical S-TAG value. In the first of these alternatives, the PBN adds or removes the tag at the service interface; in the latter, it is translated. S-VLAN tag translation within the PBN also allows the service provider to re-allocate service instances without changing the customers' service instance selection. The VLAN tags used for service selection also carry priority and drop_eligible fields that can be policed and changed by the service provider.

NOTE—This summary of PBN service instance selection, segregation, and priority handling serves only to provide context for the provisions for this standard. For an authoritative specification, refer to IEEE Std 802.1Q.

Since VLAN tags used for communicating priority information and for service instance selection can be modified by the PBN service provider, they cannot be protected by MACsec, as any change would then result in discard on verification failure. It is also preferable for the service provider not to have to make any change to accommodate MACsec-protected frames. From the service provider's point of view, a frame whose initial EtherType is the MACsec EtherType is untagged (unless the provider's interface is itself a member of the CA that is protecting frames between the interface and customer's equipment). Figure 15-3 illustrates the passage of two MACsec-protected frames from one customer bridge network to another through port-based interfaces provided by a PBN. The first frame comprises a MAC DA, SA, and arbitrary data, with each of these frame fields reaching its destination unmodified. The second comprises a MAC DA and SA, a C-VLAN tag, and arbitrary data. Each of these frame fields also reaches its destination unmodified—the C-VLAN tag is just one possibility for the initial octets of the arbitrary data of the first example frame—and its protection ensures that an attacker cannot simply change the VLAN assignment of the frame.

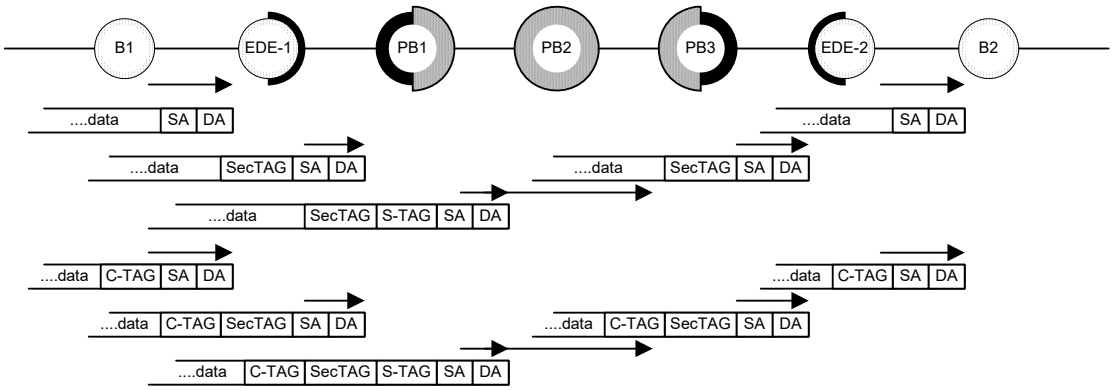


Figure 15-3—MACsec protected frame traversing a PBN

Since the PBN service provider (operating PB1) sees frames received from the customer as untagged, Figure 15-3 does not provide a way for EDE-1 to communicate each frame's priority to PB1. This can be done by priority tagging the frame after protecting it with MACsec. This requirement to communicate priority is so common as to be a required EDE-M capability (15.4, Figure 15-4). Other types of EDE, including the EDE-CS (15.5, Figure 15-6), add a full VLAN tag that depends on the protected VLAN tag so that the service provider can support service selection without requiring access to the protected data.

15.4 Securing PBN connectivity with an EDE-M

The point-to-point connectivity between the EDEs shown in Figure 15-1 could equally be provided by a Provider Bridged Network (PBN). Figure 15-4 illustrates the use of a pair of EDE-Ms to secure connectivity between Customer Bridges attached to port-based service interfaces provided by a PBN. (Numeric references in the figure are to 8.5.1, 6.9, and 6.13 in IEEE Std 802.1Q-2018.)

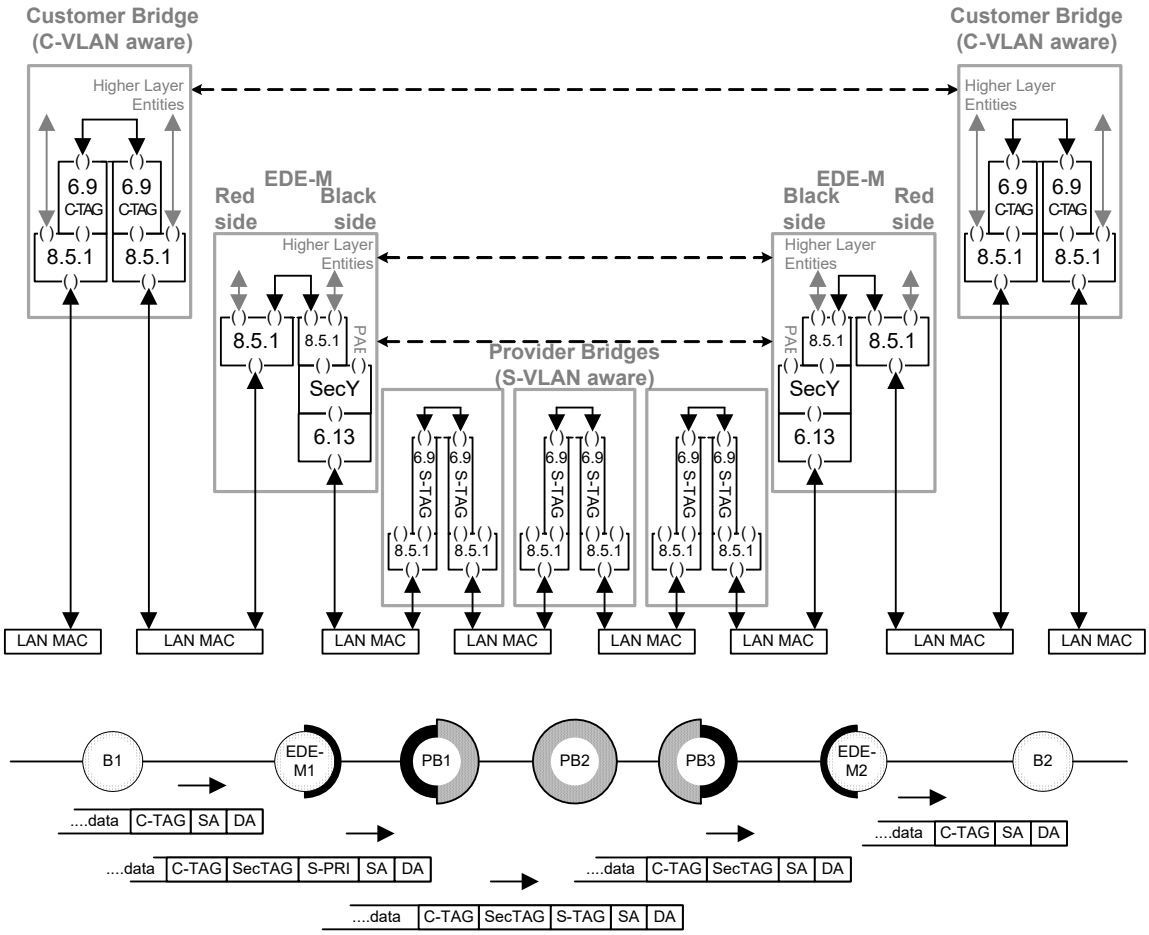


Figure 15-4—EDE-Ms securing point-to-point LAN connectivity across a PBN

As shown in the figure, EDE-M1 (on the left) may recover the signaled priority from the C-VLAN tag (if present) of the received frame as specified by IEEE Std 802.1Q [see 6.20 (Support of the ISS with signaled priority) of IEEE Std 802.1Q-2018] before protecting the frame using MACsec. If the EDE-M1 is capable of recovering signaled priority, it shall also be capable of being configured to priority tag [see 6.13 (Support of the ISS for attachment to a PBN) of IEEE Std 802.1Q-2018] or not priority tag frames transmitted by the black-side port. EDE-M2 (on the left) receives the frame and (if it is capable of recognizing signaled priority and priority tagging frames) removes any S-TAG immediately following the MAC SA and DA (not shown in the figure, which assumes the service provider interface has been configured to deliver frames that are not S-tagged). It validates the frame using MACsec before recovering the priority originally signaled by B1 in the C-VLAN tag (if present, and if capable of recovering signaled priority) and forwarding the frame.

In this scenario, each EDE-M's PAE uses the Bridge Group Address (01-80-C2-00-00-00) as the destination address of EAPOL PDUs transmitted to its peer PAE. The use of this address is specified in IEEE Std 802.1X, and it is also identified as the Nearest Customer Bridge group address by

IEEE Std 802.1Q. Each EDE-M filters any frame with a destination address that is one of the Reserved Addresses specified by IEEE Std 802.1Q as filtered by MAC Bridge and C-VLAN components (Table 8-1 of IEEE Std 802.1Q-2018) with the exception of the Nearest Customer Bridge group address.

NOTE—A frame that is received on the red-side port and not forwarded by the MAC Relay Entity will not be received by the PAE for the black-side port or transmitted on the black-side port. A frame that is received on the black-side port and not forwarded will not be transmitted on the red-side port but can be received by the PAE for the black-side port.

The PBN service is not necessarily limited to point-to-point connectivity, Figure 15-5 illustrates the secure use of a multi-point service to connect three customer bridges.

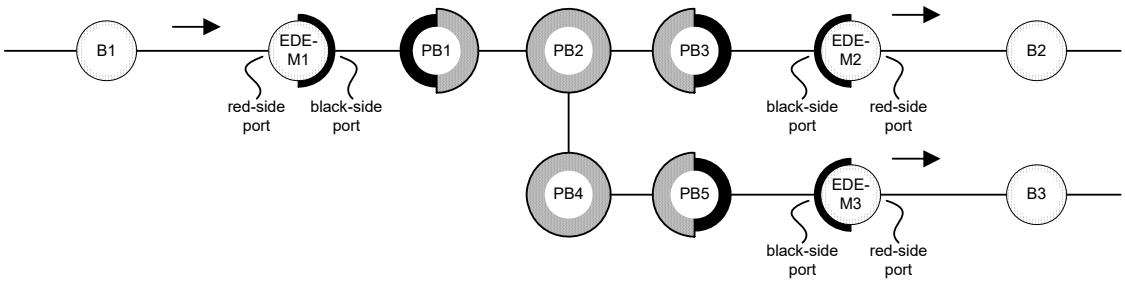


Figure 15-5—EDE-Ms securing multi-point PBN connectivity

In the figure the provider bridge ports attached to the black-side ports of each of the EDEs are assumed to be configured to provide connectivity between each of the latter.

15.5 Securing PBN connectivity with an EDE-CS

An EDE-CS provides a red-side C-tagged service interface and uses a black-side S-tagged service interface. It allows an attached customer network to use C-VIDs to select between provider service instances, protects each of those service instances with a separate CA, and identifies frames for each with a single S-VID. An EDE-CS comprises both a C-VLAN and an S-VLAN component, just as in a Provider Edge Bridge (PEB), with the following additions and restrictions. A single C-VLAN component provides a single red-side Customer Edge Port and one or more Provider Edge Ports, each supported by a SecY. Each of the Provider Edge Ports is attached to one of the Customer Network Ports of the S-VLAN component, which supports a single black-side Provider Network Port.

NOTE—The terms *customer* and *provider* applied to the external and internal ports of an EDE-CS are those used by IEEE Std 802.1Q in its description of PBs, PEBs, and BEBs and reflect the role of those ports in the layered network architecture. They do not indicate control or ownership of the equipment.

Figure 15-6 depicts an example network, with a single provider operated bridge (PB) that provides S-tagged service interfaces to two customer operated EDE-CSs and a customer operated PEB, and a port-based interface to a customer owned EDE-M. Consider EDE-CS1, on the left of the figure. This has a C-VLAN component, with a single Customer Edge Port connected to bridge B1 and three Provider Edge Ports, and an S-VLAN component, with a Provider Network Port attached to the provider's S-tagged service interface and three Customer Network Ports. Each of the three Provider Edge Ports has a SecY that protects transmitted and received frames and connects to one of the Customer Network Ports. EDE-CS1's C-VLAN component is constrained, as required by IEEE Std 802.1Q's specification of PEB operation, to forward frames received on its Customer Edge Port for a given VLAN to at most one of its Provider Edge Ports, and hence to at most one Customer Network Port. Each of the Customer Network Ports does not have a SecY, and thus treats each received frame as untagged and assigns it to the S-VLAN identified by the port's PVID as follows:

— The upper port's PVID is configured with the S-VID used by the point-to-point service instance that provides connectivity to EDE-M2 (thus protecting communication between B1 and B2).

— The middle port's PVID provides connectivity to the upper Provider Edge Port in EDE-CS3 (protecting communication between B1 and B3).
— The lower port's PVID provides connectivity to the upper C-VLAN component in PEB4 (with a SecY on its Provider Edge Port, protecting communication between B1 and B4.1).

Not all the protected traffic has to pass through EDE-CS1 or be associated with one of its protected service instances; the figure also shows EDE-CS3 and PEB4 (with a SecY on the Provider Edge Port of its lower C-VLAN component) protecting traffic between B3 and B4.2.

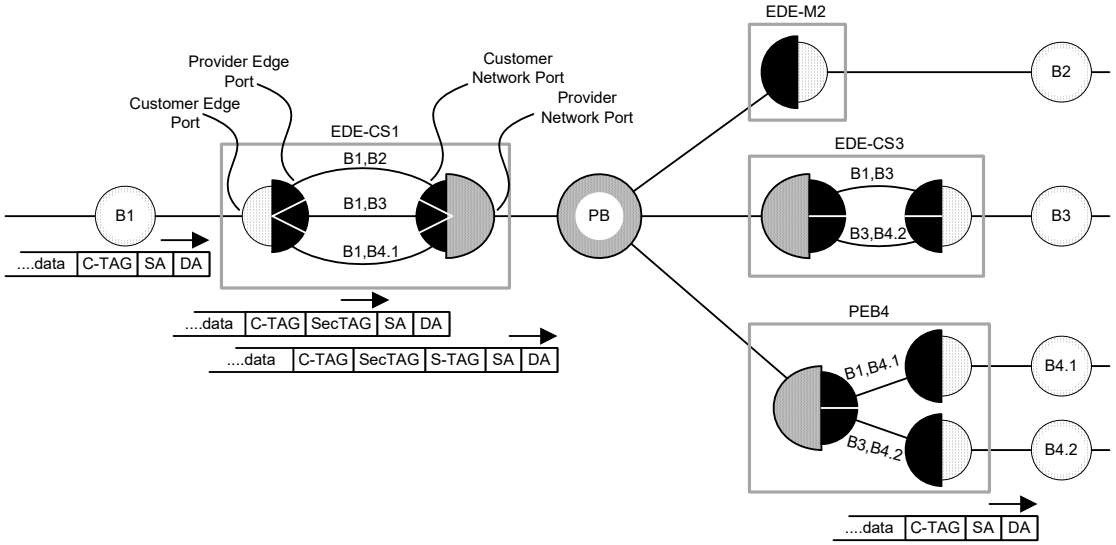


Figure 15-6—Example network with an EDE-CS

Figure 15-7 shows the internal architecture of an EDE-CS together with that of a Customer Bridge and a Provider Bridge (attached to the EDE-CS's Customer Edge Port and Provider Network Port respectively). This view of the interface stacks involved in the connection of the EDE-CS to an S-tagged interface depicts just one path through the network.

The PAE of each EDE-CS's Provider Edge Port shall be capable of being configured to use the Bridge Group Address (01-80-C2-00-00-00, also known as the Nearest Customer Bridge group address) as the destination address of EAPOL PDUs that it transmits and receives. The C-VLAN component of an EDE-CS filters any frame with a destination address that is one of the Reserved Addresses specified by IEEE Std 802.1Q as filtered by MAC Bridge and C-VLAN components (Table 8-1 of IEEE Std 802.1Q-2018), including the Nearest Customer Bridge group address. The S-VLAN component filters any frame with a destination address that is one of the Reserved Addresses specified by IEEE Std 802.1Q as filtered by S-VLAN components.

NOTE—Because the EDE-CS's C-VLAN component provides connectivity to multiple service instances, it does not offer the same level of transparency to protocols using the Nearest Customer Bridge group address as does an EDE-M.

In Figure 15-6 the PAE associated with the uppermost Provider Edge Port on EDE-CS1 is connected (over an S-VLAN supported by the Provider Bridge PB) to the PAE for the black-side port of EDE-M2, and exchanges EAPOL PDUs with that PAE. The path between the PAEs is supported by the S-VLAN components (the network component in EDE-CS1 and the Provider Bridge PB) that do not filter frames with the destination MAC address used by the EAPOL PDUs. Similarly EAPOL PDUs are exchanged between the PAE for the middle Provider Edge Port of EDE-CS1 exchanges EAPOL PDUs with the PAE for the upper Provider Edge Port of EDE-CS3 (over another S-VLAN), between the PAE for lower Provider Edge Port of EDE-CS1 and that for the upper Provider Edge Port of PEB4, and between the PAEs for the lower Provider Edge Ports for EDE-CS3 and PEB4.

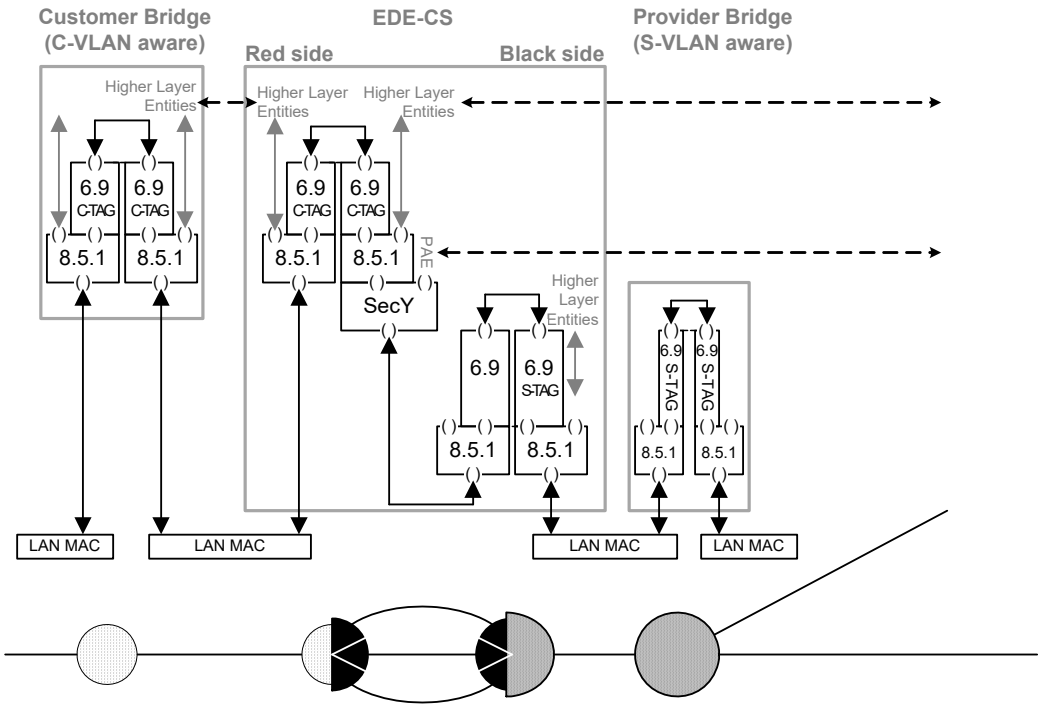


Figure 15-7—EDE-CS connected to a PBN S-tagged interface

15.6 Securing PBN connectivity with an EDE-CC

A service provider might offer C-tagged service interfaces but not S-tagged interfaces. Equally it might be desirable to secure existing network connectivity by adding an EDE between a Customer Bridge and the provider network while retaining the C-VLAN service selection capability. Figure 15-8 illustrates this before and after scenario.

The upper half of the figure depicts a Customer Bridge attached to a PEB that provides access to three point-to-point service instances, each selected by one or more C-VIDs or by the PVID used by the PEB's Customer Edge Port to classify frames received untagged. Frames with any given VID are forwarded through at most one of the PEB's Provider Edge Ports and hence through at most one Customer Network Port. The PEB's S-VLAN component sees each of these frames as untagged (as it does not recognize a C-TAG) and assigns each to the S-VLAN (and thus to the point-to-point service instance) identified by the Customer Network Port's PVID. The frame is S-tagged with the selected S-VID as it passes through the PEB's Provider Network Port into the PBN. In the other direction, each frame received from the PBN is directed by the PEB's S-VLAN component to Customer Network Port whose PVID matches the received S-VID. The S-TAG is removed on transmission through the Customer Network Port, revealing the C-tagged frame, which is then forwarded by the PEB's C-VLAN component. Within the PEB, one C-VLAN can be carried without a tag on each of the *internal LANs* that connects a Provider Edge Port with a Customer Network Port, which leads to the possibility of frames for each of these C-VLANs being carried without a C-TAG within the PBN.

In the lower half of Figure 15-8 an EDE-CC has been added between the Customer Bridge and the PEB. In the figure the PEB is shown as supporting three service provider instances, protecting communication between B1 and B2, B1 and B3, and between B1 and B4 (B2, B3, and B4 lying elsewhere in the network), and the EDE-CC's edge component has three Provider Edge Ports, each participating in a CA that protects one of the service instances. In this scenario, it is unnecessary to carry the additional C-TAG, added by the

EDE-CC's network component, over the PBN—it can be removed by the PEB's Provider Edge Port as the attached internal LAN suffices to identify the provider service instance within the PEB. The PBN frame format for each of the service instances is then as shown in the bottom right of the figure. If the PEB had been configured to map these CAs to one or two service instances, a C-TAG would be required to distinguish those carried over a common service instance.

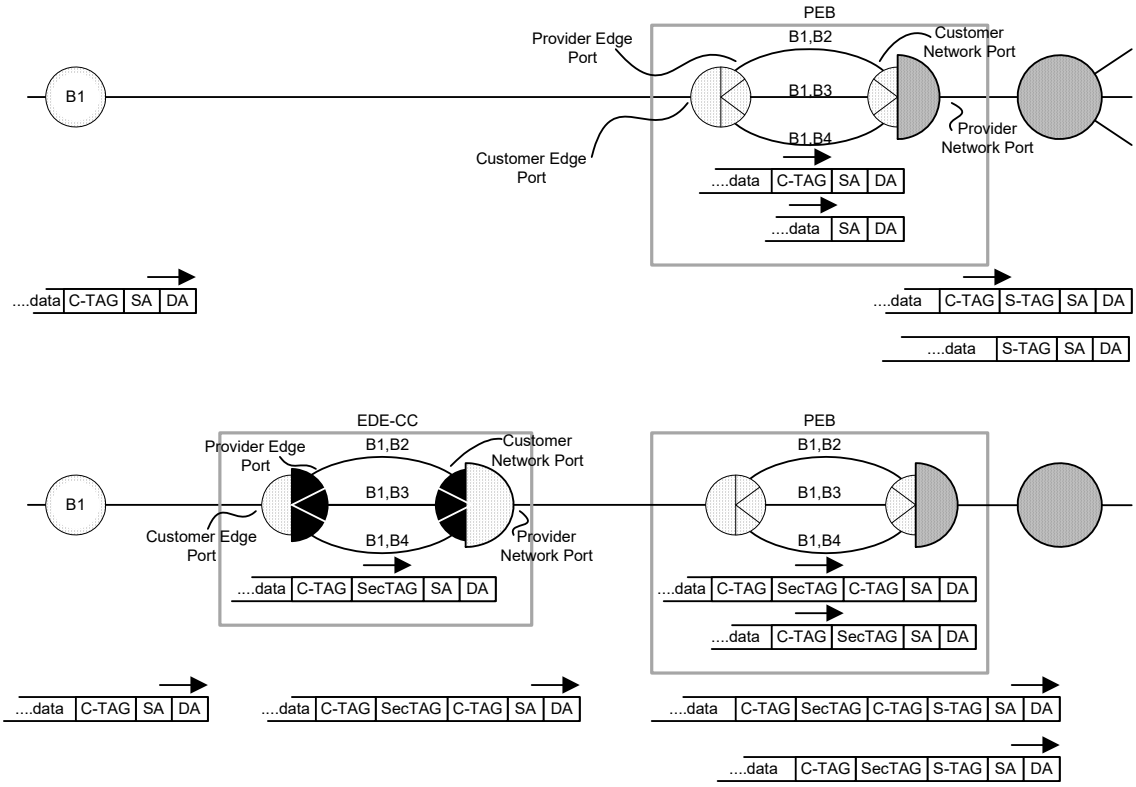


Figure 15-8—Using an EDE-CC with a C-tagged provider service interface

Figure 15-9 shows the architecture of the EDE-CC in more detail. This standard extends the use of the terms Customer Edge Port, Provider Edge Port, Customer Network Port, and Provider Network Port (initially defined in IEEE Std 802.1Q for Provider Edge Bridges) to identify ports that play similar roles in EDE-CSs, EDE-CCs, and EDE-SSs. However, if a SecY associated with a EDE-CC's Provider Edge Port is configured not to protect frames (as might be done to facilitate initial deployment), a SecTAG will not be added to transmit frames, and the EDE-CC's Customer Network Port component will see the received frames as already C-VLAN-tagged and will not add a further tag. The externally observable behavior of the EDE-CC would then resemble that of a single Customer Bridge, not a Provider Edge Bridge.

The PAE of each Provider Edge Port for an EDE-CC's edge component shall be capable of being configured to use the EDE-CC PEP Address (see Table 15-1) as the destination address of EAPOL PDUs that it transmits and receives. In Figure 15-9, the EDE-CC's C-VLAN network component (shown on the right in the figure, with a Customer Network Port and a Provider Network Port) is shown at a lower level than its accompanying edge component to emphasize the fact that it is transparent to the operation of the Provider Edge Port PAEs and other edge component protocol entities. It filters any frame with a destination address that is one of the Reserved Addresses specified by IEEE Std 802.1Q as filtered by S-VLAN components. The EDE-CC's C-VLAN edge component (shown on the left in the figure, with a Customer Edge Port and a Provider Edge Port) filters any frame with a destination address that is either one of the Reserved Addresses specified by IEEE Std 802.1Q as filtered by MAC Bridge and C-VLAN components (Table 8-1 of IEEE Std 802.1Q-2018) or the EDE-CC PEP Address.

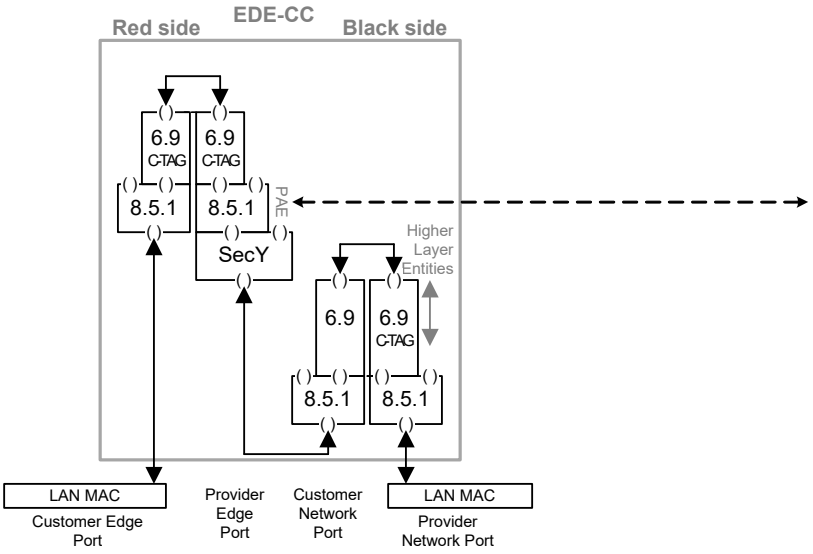


Figure 15-9—EDE-CC architecture

The configuration of an EDE-CC 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 match 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 (15.1) 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 secure Connectivity Association Keys (CAKs) and announcements—are beyond the scope of this specification (see IEEE Std 802.1X for detailed capabilities).

NOTE—The descriptive advantage of the two component architecture of PEBs and EDEs is not limited to VLAN multiplexing over service instances. It allows existing and developing port-based queue servicing specifications to be applied in the context of the resources available to each provider service instance, for example.

15.7 Securing PBN connectivity with an EDE-SS

An EDE-SS addresses a similar requirement to that for an EDE-CC—securing existing network connectivity with minimal change to existing systems, in this case retaining S-VLAN service selection capability, using the same architecture with two S-VLAN components instead of two C-VLAN components. The same configurations restrictions apply: the value of the outer S-VID added and removed by the EDE's network component matches that of the inner S-VID, and frames received untagged by the Customer Edge Port are transmitted untagged by the Provider Network Port and vice versa.

The PAE of each EDE-SS's Provider Edge Port shall be capable of being configured to use the EDE-SS PEP Address (see Table 15-1) as the destination address of EAPOL PDUs that it transmits and receives. The S-VLAN network component filters any frame with a destination address that is one of the Reserved Addresses specified by IEEE Std 802.1Q as filtered by S-VLAN components. The S-VLAN edge

component filters any frame with a destination address that is one of these addresses or the EDE-SS PEP Address.

NOTE—An EDE device with red-side recognition of S-TAGs and black-side addition and removal of I-TAGs and B-TAGs, used to secure connectivity across a Provider Bridged Backbone Network (PBBN), would not differ from a BEB with an EDE-SS on the customer side and is therefore not described in this standard.

15.8 EDE Interoperability

The PAEs specified for each EDE type (in 15.2, 15.4, 15.6, and 15.7) can be configured to use the group MAC address typically used by a potential peer MACsec capable system. Table 15-1 summarizes the group addresses specified by this standard and IEEE Std 802.1X, and their filtering by bridge components.

Table 15-1—PAE Group Addresses

Address assignment	Address value	Address filtering				
EDE-CC PEP Address	01-80-C2-00-00-1F	Y[a]				
Bridge Group Address, Nearest Customer Bridge group address	01-80-C2-00-00-00	Y	Y			
EDE-SS PEP Address	01-80-C2-00-00-0B	Y	Y	Y		
Nearest non-TPMR Bridge group address, IEEE Std 802.1X PAE address[b]	01-80-C2-00-00-03	Y	Y	Y	Y	
Individual LAN Scope group address, Nearest Bridge group address[c]	01-80-C2-00-00-0E	Y	Y	Y	Y	Y
			EDE-CC Edge components			
				MAC Bridge & C-VLAN components (Customer Bridges, PEB w/multiple PEPs[d])		
					PEB C-VLAN components w/ single PEP	
						S-VLAN components (Provider Bridges, Provider Backbone Bridges, PEBs)
						TPMR components

[a]Y indicates, Yes, this address is filtered by the component.
[b]Identified as the "IEEE Std 802.1X PAE address" in IEEE Std 802.1Q-2003 and IEEE Std 802.1Q-2005 and as the "PAE group address" in IEEE Std 802.1X-2010.
[c]It is intended that no IEEE 802.1 relay device will be defined that will forward frames that carry this destination address.
[d]A PEB's C-VLAN component with multiple PEPs supports more than one provider network service instance.

Table 15-2 summarizes the use of these addresses in various scenarios. In each case, the choice of address is constrained by the need for it to be forwarded (and not filtered) by intervening components. For example, a PAE for an EDE-M connected via a port-based interface providing access to a single provider service instance can use the Nearest Customer Bridge group address to communicate to a peer PAE in a similarly connected EDE-M or Customer Bridge, but cannot use that address to communicate to an EDE-CC connected to a C-tagged interface. Where connectivity is impossible or undesirable, an address is not given. For example, securing connectivity between a EDE-CC connected to a provider network and a TPMR connected to a distant LAN might be possible using the EDE-CC PAE address, but could render traffic relayed by the TPMR unintelligible to neighboring Customer Bridges, which operate at a higher sublayer in the network's connectivity. The network administrator should take care not to introduce a similar (sub)layering violation by configuring the PAE of an EDE-M that is not directly connected to a PBN with the EDE-CC PAE address, as that could interfere with the operation of configuration protocols between the EDE-M and its immediate neighbors.

In addition to agreeing on the group addresses to be used by their PAEs, MACsec-capable systems connected to a PBN can only interoperate if the use, addition, removal, or modification of VLAN tags by the provider network is appropriate. EDE-CCs (for example) rely on the presence of VLAN tags to distinguish

provider network service instances. EDE-Ms (on the other hand) need to receive frames from the attached single provider network service instance untagged. When EDE-Ms (or MACsec-capable Customer Bridges) are connected across a PBN in a hub-and-spoke configuration to an EDE-CC acting a hub, the PBN has to be configured to add a C-VLAN tag prior to EDE-CC reception so that the latter can separate frames from each spoke and has to remove the outer C-VLAN tag from frames transmitted by the EDE-CC before they are delivered to each spoke.

Table 15-2—PAE Group Address use

System	Connectivity	Address[a,b]								
EDE-CC	C-tagged PBN i/f	-1F	-1F	-1F						
EDE-M	Port-based PBN i/f	-1F	-00	-00	-00					
Customer Bridge	Port-based PBN i/f	-1F	-00	-00	-00					
EDE-CS	S-tagged PBN i/f		-00	-00	-00					
EDE-SS	S-tagged PBN i/f					-0B				
Provider Bridge	Individual LAN						-03	-03	-03	-0E
EDE-M	Individual LAN						-03	-03	-03	-0E
Customer Bridge	Individual LAN						-03	-03	-03	-0E
TPMR	Individual LAN						-0E	-0E	-0E	-0E

[a]The system connected as shown in the *i*th row of the table interoperates with that in the *j*th row (also connected as shown in that row) using the group address in *i*th row and *j*th column.
[b]For convenience the EDE-CC PEP Address is shown as -1F, the Nearest Customer Bridge group address as -00, the EDE-SS PEP Address as -0B, the Nearest non-TPMR Bridge group address as -03, and the Nearest Bridge group address as 0E.

NOTE—Table 15-2 is not intended to cover all MACsec interoperability scenarios.

Any outer C-VLAN (or S-VLAN in the case of an EDE-SS) tag only facilitates service selection and connectivity across the PBN. This tag is neither used or trusted by systems attached to red-side ports. The level of trust associated with connectivity between EDEs is based on contents of each MACsec-protected frame, including (where appropriate) the VLAN tag. For example, red-side traffic can be separated by VLAN according to the use made of each VLAN. If the PAE of an EDE-CC's Provider Edge Port establishes connectivity across a PBN and the level of authorization (based on authentication attributes) associated with the remote EDE is sufficient only to permit limited access (perhaps to a *guest VLAN*), then the EDE-CC's Edge Components VLAN ingress controls for the PEP can be configured to deny access to other VLANs.

15.9 EDEs, CFM, and UNI Access

Provider network operators typically define a User Network Interface (UNI) that enables autoconfiguration of devices attached to provider network services and provides service status information. One set of protocols that supports these UNI functions is the Ethernet Local Management Interface (E-LMI) specified by the MEF Forum (MEF 16 [B14]). Connectivity Fault Management (CFM) as specified by Clause 18 through Clause 22 of IEEE Std 802.1Q-2018 can be used to monitor provider service instances, detecting and isolating connectivity failures. CFM supports the hierarchical nesting of maintenance domains, naturally complementing the service (sub)layering enforced by the use of MACsec. CFM PDUs that are interpreted, modified, or generated by the provider network are not forwarded through EDE red-side ports, but are sourced and sunk by maintenance end points within an EDE. An EDE-CC PEP can be configured to relay unprotected traffic for a given VLAN so that a red-side attached system can access the provider network UNI directly, but the red-side network then depends on those red-side systems not to forward frames for those VLANs further, and those systems would themselves depend on filtering of unprotected frames within the EDE for their own protection. The specification of such filtering or the substitution of equivalent proxy functions is specific to UNI operations and beyond the scope of this specification.

16. Using MIB modules to manage EDEs

Ethernet Data Encryption devices (EDEs), specified in Clause 15, can be managed using the MIB modules specified in IEEE Std 802.1Q, IEEE Std 802.1X, and Clause 13 (MAC Security Entity MIB) of this standard. This clause specifies how EDEs are managed using these MIB modules, drawing particular attention to how constraints on EDE behavior are reflected in the MIBs and their use.

16.1 Security considerations

The user of this standard is encouraged to review the security considerations described by the defining standard for each of the MIB modules referenced by this clause. An EDE implementation can restrict remote management use of the management controls identified as sensitive. When used to manage EDE functionality, a greater weight might be placed on their sensitivity than in other uses.

16.2 EDE-M Management

An EDE-M is managed as a VLAN-unaware MAC Bridge (5.14 of IEEE Std 802.1Q-2018) using the MIB modules specified in IEEE Std 802.1Q, with the black-side port interface stack including a SecY managed by the MIB module specified in this standard and supported by a PAE managed by the MIB module specified in IEEE Std 802.1X. The Bridge Name (the sysDescr object in the SNMPv2-MIB) should identify the bridge as an EDE-M, but this standard does not constrain the syntax of that identification. Similarly the Port Name (the ieee8021BridgeBasePortName in the IEEE8021-BRIDGE-MIB) should identify the red-side and black-side ports, but this standard does not constrain the syntax of that identification.

16.3 EDE-CS Management

An EDE-CS is managed as a Provider Edge Bridge (5.10.2 of IEEE Std 802.1Q-2018) using the MIB modules specified in IEEE Std 802.1Q, with each Provider Edge Port (PEP) stack including a SecY managed by the MIB module specified in this standard and supported by a PAE managed by the MIB module specified in IEEE Std 802.1X. The Bridge Name (the sysDescr object in the SNMPv2-MIB) should identify the bridge as an EDE-CS, but this standard does not constrain the syntax of that identification. Similarly the Port Name (the ieee8021BridgeBasePortName in the IEEE8021-BRIDGE-MIB) should identify the red-side Customer Edge Port and the black-side Provider Network Port, but this standard does not constrain the syntax of that identification.

16.4 EDE-CC and EDE-SS Management

The sysDescr object in the SNMPv2-MIB should identify the system as an EDE-CC or EDE-SS as appropriate, but this standard does not constrain the syntax of that identification.

The edge component of an EDE-CC or EDE-SS, its Customer Edge Port, and its Provider Edge Ports may be managed using the MIB modules specified in IEEE Std 802.1Q. The edge component has a ComponentID of 1. The red-side port is the only Bridge Port for that component that is identified as a Customer Edge Port.

NOTE—The applicable MIB modules specified in IEEE Std 802.1Q include, at a minimum, the IEEE8021-TC-MIB, the IEEE8021-BRIDGE-MIB, and the IEEE8021-Q-BRIDGE-MIB. The ComponentId is used as the ieee8021BridgeBaseComponentId in the ieee8021BridgeBaseTable of the IEEE8021-BRIDGE-MIB, and the ieee802QBridgeComponentId in the ieee8021QBridgeTable of the IEEE8021-QBRIDGE MIB.

An EDE-CC or 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). The network component 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.

Annex A

(normative)

PICS proforma[17]

A.1 Introduction

The supplier of a protocol implementation which is claimed to conform to this standard 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.

A.2 Abbreviations and special symbols

A.2.1 Status symbols

M	mandatory
O	optional
O.n	optional, but support of at least one of the group of options labelled by the same numeral n is required
X	prohibited
pred:	conditional-item symbol, including predicate identification: see A.3.4
¬	logical negation, applied to a conditional item's predicate

A.2.2 General abbreviations

N/A	not applicable
PICS	Protocol Implementation Conformance Statement

[17] *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.

A.3 Instructions for completing the PICS proforma

A.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 A.3.4 below. 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 labelled *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.

A.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.

A.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.

A.3.4 Conditional status

A.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 A.3.4.2 below, and S is a status symbol, M or 0.

If the value of the predicate is True (see A.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.

A.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.

A.4 PICS proforma for IEEE Std 802.1AE

A.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).	

A.4.2 Protocol summary, IEEE Std 802.1AE

Identification of protocol specification	IEEE Std 802.1AE-2018, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security
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 A.3.3: the answer Yes means that the implementation does not conform to IEEE Std 802.1AE-2018.)	No [ ] Yes [ ]

Date of Statement	

A.5 Major capabilities

Item	Feature	Status	References	Support
SAP	Does the implementation of each MAC Security Entity support the Controlled and Uncontrolled Ports, and use a Common Port as specified in Clause 10?	M	5.3(a), Clause 10, A.6	Yes [ ]
STAT	Does the implementation support the MAC status and point-to-point parameters for the Controlled and Uncontrolled Ports as specified in 6.4, 6.5, and 10.7?	M	5.3(b), 6.4, 6.5, 10.7, A.7	Yes [ ]
GEN	Does the implementation process transmit requests from the Controlled Port as required by the specification of Secure Frame Generation (10.5)?	M	5.3(c), 10.5, A.8	Yes [ ]
VER	Does the implementation process receive indications from the Common Port as required by the specification of Secure Frame Verification (10.6), prior to causing receive indications at the Controlled Port?	M	5.3(d), 10.6, A.9	Yes[ ]
FMT	Does the implementation encode and decode MACsec PDUs as specified in Clause 9?	M	5.3(e), Clause 9, A.10	Yes [ ]
SCI	Does the implementation use a 48-bit MAC Address and a 16-bit Port Identifier unique within the scope of that address assignment to identify each transmit SCI, as specified in 8.2.1?	M	5.3(f), 8.2.1	Yes [ ]
PERF	Does the implementation satisfy the performance requirements specified in Table 10-3 and 8.2.2?	M	5.3(g), 10.1, Table 10-3, 8.2.2	Yes [ ]
FCS	Does the implementation introduce an undetected frame error rate greater than that achievable by preserving the original FCS, as required by 10.4?	X	5.3(n), 10.4, 6.10	No [ ]
KAY	Does the implementation support the LMI operations required by the Key Agreement Entity as specified in Clause 10?	M	5.3(h), Clause 10, A.11	Yes [ ]
MGT	Does the implementation provide the management functionality specified in 10.7?	M	5.3(i), 10.7, A.12.1	Yes [ ]
MIB	Does the implementation support access to MACsec parameters by a network management using SNMPv3 and the MIB module specified in Clause 13?	O	5.3(a), Clause 13	Yes [ ] No[ ]
SNMX	Does the implementation support access to MACsec parameters using any version of SNMP prior to SNMPv3?	X	5.3(p)	No[ ]
MSC	Does the implementation support more than one receive SC?	O	5.4(b)	Yes [ ] No[ ]
MSAK	Does the implementation support more than two receive SAKs?	O	5.4(c)	Yes [ ] No[ ]
CS	Does the implementation protect and validate MACsec PDUs by using implemented Cipher Suites as specified in 14.1?	M	5.3(j), 14.1	Yes [ ]
CSI	Does the implementation support Integrity Protection using the Default Cipher Suite specified in Clause 14?	M	5.3(k), Clause 14, 14.5	Yes [ ]
CSC	Does the implementation support Confidentiality Protection using the Default Cipher Suite without a Confidentiality Offset as specified in Clause 14?	¬CSO:O CSO:M	5.4(e), Clause 14, 14.5	Yes [ ]
CSO	Does the implementation support Confidentiality Protection using the Default Cipher Suite with a Confidentiality Offset as specified in Clause 14?	O	5.4(f), Clause 14, 14.5,	Yes [ ]

A.5 Major capabilities *(continued)*

Item	Feature	Status	References	Support
CSA	Does the implementation include Cipher Suites that are specified in Clause 14 in addition to the Default Cipher Suite? (This PICS requires the completion of a copy of Table A.13 for each such Cipher Suite implemented.)	O	5.4(g), A.13	Yes [ ] No[ ]
CSX	Does the implementation include any Cipher Suite that is additional to those specified in Clause 14 and does not meet all the criteria specified in 14.2, 14.3, 14.4.1?	X	5.3(o), 14.2, 14.3, 14.4.1	No[ ]
CSV	Does the implementation include Cipher Suites other than those specified in Clause 14, but meeting the criteria specified in 14.2, 14.3, 14.4.1? (This PICS requires the completion of a copy of Table A.14 for each such Cipher Suite implemented.)	O	5.4(i), A.14	Yes [ ] No[ ]
CSR	Does the implementation support a minimum of one receive SC, two receive SAKs, one transmit SC, and one of the two receive SAKs at a time for transmission as specified in 5.3(l), for each Cipher Suite implemented?	M	5.3(l), Clause 14	Yes [ ]
CSS	Does this completed PICS specify the maximum number of receive SCs, receive SAKS, and transmit SCs for each Cipher Suite implemented?	M	5.3(m), A.13, A.14	Yes [ ]
CSRC	What is the maximum number of receive SCs supported by the Default Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSRK	What is the maximum number of receive SAKs supported by the Default Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSTC	What is the maximum number of transmit SCs supported by the Default Cipher Suite implementation? _ _ _ _ _		5.3(m)	
TC	Does the implementation support more than one transmit SC for any Cipher Suite?	O	5.4(d)	Yes [ ] No [ ]
TCT	Is a Traffic Class Table implemented?	**TC**:M	5.4(h), 10.7.17	Yes [ ] N/A[ ]
TCAPT	Is an Access Priority Table implemented?	**TC**:M	5.4(h), 10.7.17,	Yes [ ] N/A[ ]
FULL	Is a claim for full conformance being made for the implementation?	**CSV**:X ¬**CSV**:O	5.3	Yes [ ] No[ ]
VAR	Is a claim for conformance with cipher suite variance being made for the implementation?		5.3	Yes [ ] No[ ]

A.6 Support and use of Service Access Points

Item	Feature	Status	References	Support
SAP-1	Does each transmit request from the Uncontrolled Port result in a single request to the Common Port with the same parameters?	M	10.4	Yes [ ]
SAP-2	Does each receive indication from the Common Port result in a single indication to the Uncontrolled Port with the same parameters if any of the users of the Common Port wishes to receive the indication?	M	10.4	Yes [ ]
SAP-3	Does each transmit request from the Controlled Port result in at most one request to the Common Port?	M	10.4	Yes [ ]
SAP-4	Does each receive indication from the Common Port result in at most one indication to the Controlled Port?	M	10.4	Yes [ ]
SAP-5	Are any transmit requests made to the Common Port that do not correspond to requests made at the Uncontrolled or Controlled Port?	X	10.4	No [ ]
SAP-6	Are any receive indications caused at the Uncontrolled or Controlled Port that do not correspond to indications from the Common Port?	X	10.4	No [ ]
SAP-7	Is the order of requests made at the Common Port unchanged from the order of corresponding requests from the Uncontrolled Port?	M	10.4	Yes [ ]
SAP-8	Is the order of requests made at the Common Port unchanged from the order of corresponding requests from the Controlled Port?	M	10.4	Yes [ ]
SAP-9	Is the order of receive indications caused at the Uncontrolled Port the same as the order of reception from the Common Port?	M	10.4	Yes [ ]
SAP-10	Is each transmit request from the Controlled Port processed in accordance with the specification of the Secure Frame Generation process, prior to discarding the request or making a corresponding request to the Common Port?	M	10.4, 10.5	Yes [ ]
SAP-11	Is each receive indication from the Common Port processed in accordance with the specification of the Secure Frame Verification process prior to causing a possible corresponding indication at the Controlled Port?	M	10.4, 10.6	Yes [ ]

A.7 MAC status and point-to-point parameters

Item	Feature	Status	References	Support
STAT-1	Are the values for MAC_Operational and operPointToPointMAC for the Uncontrolled Port identical to those for the Common Port?	M	6.4, 10.7.2	Yes [ ]
STAT-2	Is MAC_Operational False for the Controlled Port, and frames neither accepted or delivered on the port, if the SA identified by the encodingSA is not available for use and protectFrames is set?	M	6.4, 10.5.1, 7.1	Yes [ ]
STAT-3	Is MAC_Operational False for the Controlled Port and frames neither accepted nor delivered, if the nextPN for the encodingSA is zero or 2^{32}?	M	6.4, 10.5.2	Yes [ ]
STAT-4	Is MAC_Operational True only if MAC_Enabled is True and MAC_Operational for the Common Port is True?	M	6.4, 10.7.4	Yes [ ]
STAT-5	Is the value of operPointToPointMAC for the Controlled Port always as specified in 10.7.4.	M	6.5, 10.7.4	Yes [ ]

A.8 Secure Frame Generation

Item	Feature	Status	References	Support
GEN-1	Does each transmit request from the Controlled Port result in an identical transmit request at the Common Port if the management control protectFrames is False?	M	10.5	Yes [ ]
GEN-2	Does each transmit request at the Common Port resulting from a request at the Common Port convey request parameters, i.e., a frame, protected in accordance with Clause 10.5 if the management control protectFrames is True?	M	10.5	Yes[ ]
GEN-3	Is each protected frame assigned to the SA with AN corresponding to the current value of encodingSA as specified by the KaY?	M	10.5.1	Yes[ ]
GEN-4	Are frames to be protected discarded if the assigned SA cannot be used?	M	10.5.1	Yes[ ]
GEN-5	Is the PN value of zero used?	X	10.5.2	No [ ]
GEN-6	Following assignment of a PN to a protected frame, is the next frame to be protected for the same SA assigned the next higher value of PN?	M	10.5.2	Yes[ ]
GEN-7	Is the SecTAG encoded as specified in Clause 9?	M	10.5.3, Clause 9	
GEN-8	Is the ES bit set or clear as required by the management controls useES and alwaysIncludeSCI?	M	10.5.3	
GEN-9	Is the SC bit set or clear and the SCI explicitly encoded or not as required by the management controls useES, use SCB, alwaysIncludeSCI, and by the number of receive SCs?	M	10.5.3	
GEN-10	Is the SCB bit set or clear as required by the management controls useSCB and alwaysIncludeSCI?	M	10.5.3	
GEN-11	Is the E bit set if the frame is confidentiality protected, and clear otherwise?	M	9.5	
GEN-12	Is the C bit set if the octets of the Secure Data differ from those of the User Data or the ICV is not 16 octets, and clear otherwise?	M	9.5	
GEN-13	Is each frame transmitted from the Controlled Port protected using a Cipher Suite as specified in Clause 14 if protectFrames is set?	M	10.5	
GEN-14	Is OutOctetsEncrypted incremented by the number of octets in the User Data if confidentiality protections is provided, and OutOctetsProtected incremented otherwise?	M	10.5.4	
GEN-15	Is the protected frame transmitted if the MACsec PDU (SecTAG, Secure Data, and ICV) does not exceed the maximum data unit size supported by the Common Port and discarded otherwise?	M	10.5.5	

A.9 Secure Frame Verification

Item	Feature	Status	References	Support
VER-1	For each receive indication, does the Secure Frame Verification process examine the user data for a SecTAG and validate frames with a SecTAG as specified in 9.12, extracting and decoding the SecTAG as specified in 9.3 through 9.9, and extracting the User Data and ICV as specified in 9.10 and 9.11?	M	10.6, 9.3 through 9.9, 9.10, 9.11, 9.12	Yes[ ]
VER-2	Is a received frame without a SecTAG delivered to the Controlled Port if validateFrames is not Strict, and discarded otherwise?	M	10.6	Yes[ ]
VER-3	Is a received frame with the SecTAG E bit set and C bit clear discarded and not delivered to the Controlled Port?	M	10.6	Yes[ ]
VER-4	Is the received frame discarded if the SC is unknown and validateFrames is Strict or the C bit is set, and delivered to the Controlled Port otherwise?	M	10.6.1	Yes[ ]
VER-5	Is the received frame discarded if the SA is unused and validateFrames is Strict or the C bit is set, and delivered to the Controlled Port otherwise?	M	10.6.1	Yes[ ]
VER-6	Is the received frame discarded if the PN is less than the lowest acceptable packet number for the SA and replayProtect is enabled?	M	10.6.2, 10.6.4	Yes[ ]
VER-7	Is the InPktsOverrun counter incremented if a received frame is discarded for reasons not attributed to the data conveyed?	M	10.6.3	Yes[ ]
VER-8	If validateFrames is Disabled, is Cipher Suite validation omitted and a received frame delivered to the Controlled Port if the C bit is not set?	M	10.6.3, 10.6.5	Yes[ ]
VER-9	If validateFrames is not Disabled is the Cipher Suite used to validated the received frame?	M	10.6.3	Yes[ ]
VER-10	Are frames that are not successfully validated discarded if validateFrames is Strict or the C bit is set?	M	10.6.5	Yes[ ]
VER-11	Are the values for the next expected and lowest acceptable PN updated as specified in 10.6.5 following receipt of a MACsec PDU successfully validate by the Cipher Suite, and not modified by received frames otherwise?	M	10.6.5	Yes[ ]
VER-12	Are received frames not discarded by Secure Frame Verification delivered to the Controlled Port after removal of a SecTAG and ICV?	M	10.6	Yes[ ]
VER-13	Are all received frames delivered to Controlled Port unmodified if validateFrames is Null?	M	10.6	Yes[ ]
VER-14	Is protectFrames set False if validateFrames is set to Null?	M	10.6	Yes[ ]

A.10 MACsec PDU encoding and decoding

Item	Feature	Status	References	Support
FMT-1	Does each MACsec PDU transmitted contain an integral number of octets?	M	9.1	Yes[ ]
FMT-2	Does each MACsec PDU transmitted comprise a SecTAG, formatted as specified in Clause 9, one or more octets of Secure Data, and an ICV of the length specified by the Cipher Suite in use?	M	9.1, 9.2, 9.3, Figure 9-1, 10.5.3	Yes[ ]
FMT-3	Is the EtherType encoded in the SecTAG the value specified in Table 9-1?	M	9.3, 9.4	Yes[ ]
FMT-4	Is the version number in the SecTAG encoded as zero?	M	9.5	Yes[ ]
FMT-5	Is the SC bit clear and the SCI not explicitly encoded if the ES bit is set?	M	9.5	Yes[ ]
FMT-6	Is the SC bit set if an SCI is explicitly encoded and clear otherwise?	M	9.5	Yes[ ]
FMT-7	Is the SC bit clear if the SCB bit is set?	M	9.5	Yes[ ]
FMT-8	Are bits 7 and 8 of octet 4 of the SecTAG zero?	M	9.7	Yes[ ]
FMT-9	Is each received MACsec PDU validated as specified in 9.12.	M	9.5	Yes[ ]

A.11 Key Agreement Entity LMI

Item	Feature	Status	References	Support
KAY-1	Does the implementation allow the KaY to read the values of the MAC_Enabled, MAC_Operational, and operPointToPointMAC parameters?	M	10.7.2	Yes[ ]
KAY-2	Does the implementation allow the KaY to set and clear the ControlledPortEnabled parameter, acting on the parameter as specified?	M	10.7.4, 10.7.5	Yes[ ]
KAY-3	Does the implementation allow the KaY to discover which Cipher Suites are implemented and how many receive SCs each can support?	M	10.2, 10.7.7, 10.7.16, 10.7.25	Yes[ ]
KAY-4	Does the implementation allow the KaY to create a receive SC?	M	10.6.1, 10.7.11	Yes[ ]
KAY-5	Does the implementation allow the KaY to create receive SAs as specified in 10.7.13?	M	10.7.13	Yes[ ]
KAY-6	Does the implementation allow the KaY to control the use of each receive SA and to update the values of the next expected PN and lowest acceptable PN as specified in 10.7.15?	M	10.7.15	Yes[ ]
KAY-7	Does the implementation allow the KaY to create transmit SAs as specified in 10.7.22?	M	10.7.22, 10.5.2	Yes[ ]
KAY-8	Does the implementation allow the KaY to control the use of each transmit SA as specified in 10.7.24?	M	10.7.24, 10.5.1, 10.5.2	Yes[ ]
KAY-9	Does the implementation allow the KaY to monitor the nextPN associated with each transmit SA in order to create a new SA with a fresh SAK prior to PN exhaustion?	M	10.7.2	Yes[ ]
KAY-10	Does the implementation allow the KaY to select the Current Cipher Suite as specified in 10.7.27?	M	10.7.27	Yes[ ]
KAY-11	Does the implementation allow the KaY to create and control an SAK as specified in 10.7.26 and 10.7.28?	M	10.7.26, 10.7.28	Yes[ ]

A.12 Management

A.12.1 Management—control and status information

Item	Feature	Status	References	Support
Can each of the following parameter values be read by management?				
MGT1-1	The SCI for the SecY	M	10.7.1	Yes[ ]
MGT1-2	MAC_Enabled, MAC_Operational, and operPointToPointMAC for the Uncontrolled Port	M	10.7.2	Yes[ ]
MGT1-3	MAC_Enabled, MAC_Operational, and operPointToPointMAC for the Controlled Port	M	10.7.4	Yes[ ]
MGT1-4	The maximum number of receive SCs and SAKs that can be in simultaneous use	M	10.7.7	Yes[ ]
MGT1-5	validateFrames, replayProtect, and replayWindow	M	10.7.8	Yes[ ]
MGT1-6	The SCI, receiving, createdTime, startedTime, and stoppedTime for each receive SC	M	10.7.12	Yes[ ]
MGT1-7	inUse, nextPN, lowestPN, createdTime, startedTime, stoppedTime, and Key Identifier for each receive SA	M	10.7.14	Yes[ ]
MGT1-8	The maximum number of SAKs that can be in simultaneous use for transmission	M	10.7.16	Yes[ ]
MGT1-9	protectFrames, useES, useSCB, and alwaysIncludeSCI	M	10.7.17	Yes[ ]
MGT1-10	transmitting, createdTime, startedTime, and stoppedTime for the transmit SC	M	10.7.21	Yes[ ]
MGT1-11	inUse, nextPN, lowestPN, createdTime, startedTime, stoppedTime, and Key Identifier for each transmit SA	M	10.7.23	Yes[ ]
MGT1-12	The currentCipherSuite identifier and the confidentialityOffset for frames with confidentiality protection	M	10.7.27	Yes[ ]
MGT1-13	transmits, receives, and createdTime for each SAK	M	10.7.29	Yes[ ]
MGT1-14	Can the management information for each implemented Cipher Suite be read?	M	10.7.25	Yes[ ]

A.12.2 Management—basic controls

Item	Feature	Status	References	Support
Can the following parameters be written by management, independently for each Controlled Port?				
MGT2-1	validateFrames	O	10.7.8, 10.6	Yes[ ] No [ ]
MGT2-2	replayProtect	O	10.7.8, 10.6.2, 10.6.4	Yes[ ] No [ ]
MGT2-3	replayWindow	O	10.7.8, 10.6.5	Yes[ ] No [ ]
MGT2-4	protectFrames	O	10.7.17, 10.5	Yes[ ] No [ ]
MGT2-5	useES	O	10.7.17, 10.5.3	Yes[ ] No [ ]
MGT2-6	useSCB	O	10.7.17, 10.5.3	Yes[ ] No [ ]
MGT2-7	alwaysIncludeSCI	O	10.7.17, 10.5.3	Yes[ ] No [ ]
Can the following parameters be written by management, independently for each Controlled Port for each CipherSuite implemented ?				
MGT2-15	enableUse	O	10.7.26	Yes[ ] No [ ]
MGT2-16	requireConfidentiality	O	10.7.26	Yes[ ] No [ ]
Can write access by management to each of the following parameters be disabled individually?				
MGT2-8	validateFrames	MGT2-1:M	10.7.8	Yes[ ]
MGT2-9	replayProtect	MGT2-2:M	10.7.8	Yes[ ]
MGT2-10	replayWindow	MGT2-3:M	10.7.8	Yes[ ]
MGT2-11	protectFrames	MGT2-4:M	10.7.17	Yes[ ]
MGT2-12	useES	MGT2-5:M	10.7.17	Yes[ ]
MGT2-13	useSCB	MGT2-6:M	10.7.17	Yes[ ]
MGT2-14	alwaysIncludeSCI	MGT2-7:M	10.7.17	Yes[ ]
Can write access by management to each of the following CipherSuite use parameters be disabled individually for each Controlled Port?				
MGT2-17	enableUse	MGT2-15:M	10.7.26	Yes[ ]
MGT2-18	requireConfidentiality	MGT2-16:M	10.7.26	Yes[ ]

A.12.3 Management—control over secure communication

Item	Feature	Status	References	Support
Can the following be created, controlled, or selected by management?				
MGT3-1	Receive SCs and SAs	O	10.7.11, 10.7.13, 10.7.15	Yes[ ] No [ ]
MGT3-2	Transmit SAs	O	10.7.22, 10.7.24	Yes[ ] No [ ]
MGT3-3	The current CipherSuite	O	10.7.27	Yes[ ] No [ ]
MGT3-4	confidentialityOffset	O	10.7.27	Yes[ ] No [ ]
MGT3-5	SAKs	O	10.7.28, 10.7.29	Yes[ ] No [ ]
Can creation, control, or selection by management of the following be disabled individually?				
MGT3-1	Receive SCs and SAs	MGT3-1:M	10.7.11	Yes[ ]
MGT3-2	Transmit SAs	MGT3-2:M	10.7.22, 10.7.24	Yes[ ]
MGT3-3	The current CipherSuite	MGT3-3:M	10.7.27	Yes[ ]
MGT3-4	confidentialityOffset	MGT3-4:M	10.7.27	Yes[ ]
MGT3-5	SAKs	MGT3-5:M	10.7.27	Yes[ ]

A.12.4 Management—statistics

Item	Feature	Status	References	Support
Are each of the following interface statistics provided for the Controlled Port as specified in 10.7.6?				
MGT4-1	ifInOctets	M	10.7.6	Yes[ ]
MGT4-2	ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts	M	10.7.6	Yes[ ]
MGT4-3	ifInDiscards	M	10.7.6	Yes[ ]
MGT4-4	ifInErrors	M	10.7.6	Yes[ ]
MGT4-5	ifOutOctets	M	10.7.6	Yes[ ]
MGT4-6	ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts	M	10.7.6	Yes[ ]
MGT4-7	ifOutErrors	M	10.7.6	Yes[ ]
Are each of the following frame verification statistics recorded as specified in 10.6 and maintained for the frame verification process as a whole?				
MGT4-8	InPktsUntagged	M	10.7.9, 10.6 Figure 10-4	Yes[ ]
MGT4-9	InPktsNoTag	M	10.7.9, 10.6 Figure 10-4	Yes[ ]
MGT4-10	InPktsBadTag	M	10.7.9, 10.6 Figure 10-4	Yes[ ]
MGT4-11	InPktsNoSARcv	M	10.7.9, 10.6.1	Yes[ ]
MGT4-12	InPktsNoSADiscard	M	10.7.9, 10.6.1	Yes[ ]
MGT4-13	InPktsOverrun	M	10.7.9, 10.6.3	Yes[ ]

A.12.4 Management—statistics *(continued)*

Item	Feature	Status	References	Support
Are each of the following frame verification statistics recorded as specified in 10.6 and maintained for each receive SC?				
MGT4-14	InPktsUnchecked	M	10.7.9, 10.6.5	Yes[ ]
MGT4-15	InPktsDelayed	M	10.7.9, 10.6.5	Yes[ ]
MGT4-16	InPktsLate	M	10.7.9, 10.6.2, 10.6.4	Yes[ ]
MGT4-17	InPktsOK	M	10.7.9, 10.6.5	Yes[ ]
MGT4-18	InPktsInvalid	M	10.7.9, 10.6.5	Yes[ ]
MGT4-19	InPktsNotValid	M	10.7.9, 10.6.5	Yes[ ]
Are each of the following frame validation statistics recorded as specified in 10.7?				
MGT4-22	InOctetsValidated	M	10.7.10	Yes[ ]
MGT4-23	InOctetsDecrypted	M	10.7.10	Yes[ ]
Are each of the following frame generation statistics recorded as specified in 10.5 and maintained for the frame verification process as a whole?				
MGT4-24	OutPktsUntagged	M	10.7.18, 10.5	Yes[ ]
MGT4-25	OutPktsTooLong	M	10.7.18, 10.5.5, Figure 10-3	Yes[ ]
Are each of the following frame generation statistics recorded as specified in 10.5 and maintained for each transmit SC?				
MGT4-26	OutPktsProtected	M	10.7.18, 10.5.4	Yes[ ]
MGT4-27	OutPktsEncrypted	M	10.7.18, 10.5.4	Yes[ ]
Are each of the following frame protection statistics recorded as specified in 10.7?				
MGT4-28	OutOctetsProtected	M	10.7.19	Yes[ ]
MGT4-29	OutOctetsEncrypted	M	10.7.19	Yes[ ]

A.13 Additional fully conformant Cipher Suite capabilities

Item	Feature	Status	References	Support
CSA-1	Name of Cipher Suite as specified in Clause 14. _			
CSA-2	Does the Cipher Suite implementation provide integrity without confidentiality?	O	14.2(a)	Yes[ ] No [ ]
CSA-3	Does the Cipher Suite implementation provide confidentiality for all the octets of the User Data?	¬CSV-19: O CSV-19: M	14.2(d), 14.3(c)	Yes[ ] No [ ]
CSA-4	Does the Cipher Suite implementation provide offset confidentiality for the User Data?	O	14.2(e), 14.3(c)	Yes[ ] No [ ]
CSA-5	What is the maximum number of receive SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSA-6	What is the maximum number of receive SAKs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSA-7	What is the maximum number of transmit SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	

A.14 Additional variant Cipher Suite capabilities

Item	Feature	Status	References	Support
CSV-1	Name of Cipher Suite or other commonly used identification (to be supplied) _			
CSV-2	Identify the specification(s) of the Cipher Suite, including any additional information necessary to acquire the specification(s) (supply items of Additional Information if necessary) _	M	14.3	
CSV-3	Does the specification include interoperable protection and verification procedures specified in terms of the parameters of 14.1?	M	14.3, 14.1	Yes [ ]
CSV-4	Does the specification state: Whether confidentiality of the User Data is provided? The maximum difference in the lengths of the User Data and Secure Data? The length of the ICV? The length and properties of the keys required, including assumptions of the scope and uniqueness?	M	 14.3(a) 14.3(b) 14.3(c) 14.3(d)	 Yes [ ] Yes [ ] Yes [ ] Yes [ ]
CSV-5	Do the Cipher Suite algorithms have an effective key length of at least 128 bits, and does any block cipher used have a block width of at least 128 bits?	M	14.4.1(a)	Yes[ ]

A.14 Additional variant Cipher Suite capabilities *(continued)*

Item	Feature	Status	References	Support
CSV-6	If serviced by separate algorithms, the properties of the authentication and confidentiality mechanisms are combinable in accordance with well-established security results?	M	14.4.1(b)	Yes[ ]
CSV-7a	Is the underlying cryptographic cipher approved by either a national or international standards body or a government agency?	O.1	14.4.1(c)(1)	Yes[ ] No[ ]
CSV-7b	Does the additional Cipher Suite meet the conditions expressed in 14.4.1(c)(2)?	O.1	14.4.1(c)(2)	Yes[ ] No[ ]
CSV-8	Does the Cipher Suite satisfy the message authentication requirements of 14.4.1? Identify the proof of security, including any additional information necessary to acquire the proof _	CSV-7b:M	14.4.1(c)(2)(i)	Yes [ ]
CSV-9	Does the Cipher Suite satisfy the confidentiality requirements of 14.4.1? Identify the proof of security, including any additional information necessary to acquire the proof _	CSV-7b:M	14.4.1(c)(2)(ii)	Yes [ ]
CSV-10	Does the Cipher Suite use mechanisms for confidentiality and authentication in a way that is consistent with the proofs of security?	CSV-7b:M	14.4.1(c)(2)(iii), 14.4.1(c)(2)(iv),	Yes[ ]
CSV-11	Does the Cipher Suite provide integrity protection for the SCI, PN, Source Address, Destination Address, SecTAG, and User Data?	M	14.2(a)	Yes[ ]
CSV-12	Does the Cipher Suite provide protection for at least 2^{32}–1 invocations without requiring a fresh SAK?	M	14.2(b)	Yes[ ]
CSV-13	Does the Cipher Suite generate a predictable number of octets of Secure Data and ICV given any specific number of octets of User Data?	M	14.2(c)	Yes[ ]
CSV-14	Does the maximum difference in length of the User Data and the Secure Data plus ICV exceed 896 octets?	X	14.2(f)	Yes[ ]
CSV-15	What is the maximum difference in length of the User Data and the Secure Data? _ _ _ _ _ octets		14.3(b)	
CSV-16	What is the length of the ICV _ _ _ _ _ octets		14.3(e)	
CSV-17	Does the specification specify the length and properties of the keys required, including assumptions of the scope of uniqueness?	M	14.3(f)	Yes[ ]
CSV-18	Does the Cipher Suite implementation provide confidentiality for all the octets of the User Data?	¬CSV-19: O CSV-19:M	14.2(d), 14.3(c)	Yes[ ] No [ ]

A.14 Additional variant Cipher Suite capabilities *(continued)*

Item	Feature	Status	References	Support
CSV-19	Does the Cipher Suite implementation provide offset confidentiality for the User Data?	O	14.2(e), 14.3(c)	Yes[ ] No [ ]
CSV-20	Does the Cipher Suite modify or constrain the values of the SCI, PN, Source Address, Destination Address, or SecTAG fields other than as specified in Clause 14?	X	14.2(g)	No [ ]
CSV-21	Does the Cipher Suite require an SAK exceeding 1024 bits long?	X	14.2(h)	No [ ]
CSV-22	Does the Cipher Suite require different keys for the protect and validate operations?	X	14.2(i)	No [ ]
CSV-23	What is the maximum number of receive SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSV-24	What is the maximum number of receive SAKs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSV-25	What is the maximum number of transmit SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	

Annex B

(informative)

Bibliography

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 802.3br™-2016, IEEE Standard for Ethernet—Amendment 5: Specification and Management Parameters for Interspersing Express Traffic.

[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 4303, IP Encapsulating Security Payload (ESP), Kent, S., Dec. 2005.

[B9] IETF RFC 5116, An Interface and Algorithms for Authenticated Encryption, McGrew, D., Jan. 2008.

[B10] 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]

[B11] McGrew, D., “Generation of Deterministic Initialization Vectors (IVs) and Nonces,” Oct. 2013.[22]

[B12] McGrew, D. A., and J. Viega, “The Galois/Counter Mode of Operation (GCM),” 31 May 2005.[23]

[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).

[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.

[B13] McGrew, D., and J. Viega, "The Security and Performance of the Galois/Counter Mode (GCM) of Operation," *Proceedings of INDOCRYPT '04*, Springer-Verlag, 2004.[24]

[B14] MEF 16, Ethernet Local Management Interface (E-LMI).[25]

[B15] Seaman, M., "The XPN recovery algorithm," June 2012.[26]

[24] Available from the IACR Cryptology ePrint Archive: Report 2004/193, https://eprint.iacr.org/2004/193.

[25] MEF technical specifications are available from the Metro Ethernet Forum (https://www.mef.net).

[26] Available at https://www.ieee802.org/1/files/public/docs2012/aebw-seaman-xpn-recovery-0612-v02.pdf.

Annex C

(informative)

MACsec test vectors

This annex provides test case examples of the use of MACsec. Each example shows an unprotected frame that could be transmitted as a result of a MAC Service request (with a given set of parameters) and the corresponding MACsec protected frame (with a given set of MACsec SecY parameters). Test cases include the use of integrity protection without confidentiality (authenticated, but unencrypted) and the use of both integrity protection and confidentiality (authenticated and encrypted).

The test cases use a number of different unprotected frame sizes. Two correspond to common sizes of internet packets, 54 octets and 60 octets—two common representations of a TCP/IP SYN packet. A TCP SYN comprises 40 octets plus 14 octets of MAC DA+SA+EtherType. The frame could be padded to 60 octets to meet minimum Ethernet frame length requirements prior to MACsec processing. The remaining frame sizes represent “corner cases” of the GCM padding algorithm. A 61-octet frame, when encrypted, has a 49-octet payload, which results in the maximum 15 octets of padding for ICV calculation. When integrity protection is provided but confidentiality is not (i.e., when the user data is not encrypted) a 65-octet frame also requires that maximum padding. A 75-octet frame has a 63 octet payload, requiring 1 octet of padding for ICV calculation, as does a 79-octet frame that is integrity protected without confidentiality. The zero-octet padding case is covered by the 60-octet frame, above. MACsec processing is performed above the media-dependent functions of media access control, so all frame sizes given are prior to the addition of the 32-bit CRC or other media dependent fields.

Test cases are provided for the Default Cipher Suite (GCM-AES-128, 14.5), GCM-AES-256 (14.6), GCM-AES-XPN-256 (14.7), and GCM-AES-XPN-256 (14.8). The notation used in this annex is that specified in Clause 14 (Cipher Suites) and NIST SP 800-38D. Fields in the MACsec header are specified in Clause 9. Summaries of the computation and intermediate outputs are provided.

C.1 Integrity protection (54-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-1. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-1—Unprotected frame (example)

Field	Value
MAC DA	D6 09 B1 F0 56 63
MAC SA	7A 0D 46 DF 99 8D
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 00 01

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, the PN (32 least significant bits for Cipher Suites using extended packet numbering), and the (optional) SCI. The PN differs for each protected frame transmitted with any given SAK (*K*) and has been arbitrarily chosen (for this and in other examples) as have the other parameter values. The fields of the protected frame are shown (in the order transmitted) in Table C-2.

Table C-2—Integrity protected frame (example)

Field	Value
MAC DA	D6 09 B1 F0 56 63
MAC SA	7A 0D 46 DF 99 8D
MACsec EtherType	88 E5
TCI and AN	22
SL	2A
PN	B2 C2 84 65
SCI	12 15 35 24 C0 89 5E 81
Secure Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 00 01
ICV	Cipher Suite and Key (SAK) dependent (see Table C-3, Table C-4, Table C-5, and Table C-6)

C.1.1 GCM-AES-128 (54-octet frame integrity protection)

Table C-3 specifies an arbitrary 128-bit key (SAK) and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-2. The GCM parameter A, the additional data to be authenticated, is formed by concatenating the MAC DA, the MAC SA, the SecTAG, and the User Data. This input is then processed through the authentication only operation of the GCM module. The SCI and the PN are concatenated (in that order) to form the 96-bit IV used by GCM. The computed GCM parameter T is the ICV. Details of the computation follow the table.

Table C-3—GCM-AES-128 Key and calculated ICV (example)

Field	Value
Key (SAK)	AD7A2BD03EAC835A6F620FDCB506B345
ICV	F0 94 78 A9 B0 90 07 D0 6F 46 E9 B6 A1 DA 25 DD

```
key size = 128 bits
P:     0 bits
A:     560 bits
IV:    96 bits
ICV:   128 bits
K:     AD7A2BD03EAC835A6F620FDCB506B345
P:
A:     D609B1F056637A0D46DF998D88E5222A
       B2C2846512153524C0895E8108000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       313233340001
IV:    12153524C0895E81B2C28465
GCM-AES Authentication
H:     73A23D80121DE2D5A850253FCF43120E
Y[0]:  12153524C0895E81B2C2846500000001
E(K,Y[0]): EB4E051CB548A6B5490F6F11A27CB7D0
X[1]:  6B0BE68D67C6EE03EF7998E399C01CA4
X[2]:  5AABADF6D7806EC0CCCB028441197B22
X[3]:  FE072BFE2811A68AD7FDB0687192D293
X[4]:  A47252D1A7E09B49FB356E435DBB4CD0
X[5]:  18EBF4C65CE89BF69EFB4981CEE13DB9
GHASH(H,A,C): 1BDA7DB505D8A165264986A703A6920D
C:
T:     F09478A9B09007D06F46E9B6A1DA25DD
```

C.1.2 GCM-AES-256 (54-octet frame integrity protection)

Table C-4 specifies an arbitrary 256-bit key (SAK) and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-2. The GCM parameter A, the additional data to be authenticated, is formed by concatenating the MAC DA, the MAC SA, the SecTAG, and the User Data. This input is then processed through the authentication only operation of the GCM module. The SCI and the PN are concatenated (in that order) to form the 96-bit IV used by GCM. The computed GCM parameter T is the ICV. Details of the computation follow the table.

Table C-4—GCM-AES-256 Key and calculated ICV (example)

Field	Value
Key (SAK)	E3C08A8F06C6E3AD95A70557B23F7548 3CE33021A9C72B7025666204C69C0B72
ICV	2F 0B C5 AF 40 9E 06 D6 09 EA 8B 7D 0F A5 EA 50

```
key size = 256 bits
P:     0 bits
A:     560 bits
IV:    96 bits
ICV:   128 bits
K:     E3C08A8F06C6E3AD95A70557B23F7548
       3CE33021A9C72B7025666204C69C0B72
P:
A:     D609B1F056637A0D46DF998D88E5222A
       B2C2846512153524C0895E8108000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       313233340001
IV:    12153524C0895E81B2C28465
GCM-AES Authentication
H:     286D73994EA0BA3CFD1F52BF06A8ACF2
Y[0]:  12153524C0895E81B2C2846500000001
E(K,Y[0]): 714D54FDCFCEE37D5729CDDAB383A016
X[1]:  BA7C26F578254853CF321281A48317CA
X[2]:  2D0DF59AE78E84ED64C3F85068CD9863
X[3]:  702DE0382ABF4D42DD62B8F115124219
X[4]: DAED65979342F0D155BFDFE362132078
X[5]: 9AB4AFD6344654B2CD23977E41AA18B3
GHASH(H,A,C): 5E4691528F50E5AB5EC346A7BC264A46
C:
T: 2F0BC5AF409E06D609EA8B7D0FA5EA50
```

C.1.3 GCM-AES-XPN-128 (54-octet frame integrity protection)

Table C-5 specifies an arbitrary value for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), a 96-bit Salt, and 128-bit key (SAK), with the ICV generated by the GCM-AES-XPN-128 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-2. The GCM parameter *A*, the additional data to be authenticated, is formed by concatenating the MAC DA, the MAC SA, the SecTAG, and the User Data. This input is then processed through the authentication only operation of the GCM module. The 32 most significant bits of the 96-bit *IV* are the octets of the SSCI, encoded as a binary number (9.1) and exclusive-or'd with the 32 most significant bits of the Salt. The 64 least significant bits of the 96-bit *IV* are the octets of the PN, encoded as a binary number (9.1) and exclusive-or'd with the 64 least significant bits of the Salt. The computed GCM parameter *T* is the ICV. Details of the computation follow the table.

Table C-5—GCM-AES-XPN-128 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	AD7A2BD03EAC835A6F620FDCB506B345
ICV	17 FE 19 81 EB DD 4A FC 50 62 69 7E 8B AA 0C 23

```
key size = 128 bits
P:     0 bits
A:     560 bits
IV:    96 bits
ICV:   128 bits
K:     AD7A2BD03EAC835A6F620FDCB506B345
P:
A:     D609B1F056637A0D46DF998D88E5222A
       B2C2846512153524C0895E8108000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       313233340001
IV:    9C002902F801C33EAEA47E08
GCM-AES Authentication
H:     73A23D80121DE2D5A850253FCF43120E
Y[0]:  9C002902F801C33EAEA47E0800000001
E(K,Y[0]): 0C246434EE05EB99762BEFD9880C9E2E
X[1]:  6B0BE68D67C6EE03EF7998E399C01CA4
X[2]:  5AABADF6D7806EC0CCCB028441197B22
X[3]:  FE072BFE2811A68AD7FDB0687192D293
X[4]:  A47252D1A7E09B49FB356E435DBB4CD0
X[5]:  18EBF4C65CE89BF69EFB4981CEE13DB9
GHASH(H,A,C): 1BDA7DB505D8A165264986A703A6920D
C:
T:     17FE1981EBDD4AFC5062697E8BAA0C23
```

C.1.4 GCM-AES-XPN-256 (54-octet frame integrity protection)

Table C-6 specifies an arbitrary value for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), a 96-bit Salt, and 256-bit key (SAK), with the ICV generated by the GCM-AES-XPN-256 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-2. The GCM parameter *A*, the additional data to be authenticated, is formed by concatenating the MAC DA, the MAC SA, the SecTAG, and the User Data. This input is then processed through the authentication only operation of the GCM module. The 32 most significant bits of the 96-bit *IV* are the octets of the SSCI, encoded as a binary number (9.1) and exclusive-or'd with the 32 most significant bits of the Salt. The 64 least significant bits of the 96-bit *IV* are the octets of the PN, encoded as a binary number (9.1) and exclusive-or'd with the 64 least significant bits of the Salt. The computed GCM parameter *T* is the ICV. Details of the computation follow the table.

Table C-6—GCM-AES-XPN-256 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	E3C08A8F06C6E3AD95A70557B23F7548 3CE33021A9C72B7025666204C69C0B72
ICV	4D BD 2F 6A 75 4A 6C F7 28 CC 12 9B A6 93 15 77

```
key size = 256 bits
P:     0 bits
A:     560 bits
IV:    96 bits
ICV:   128 bits
K:     E3C08A8F06C6E3AD95A70557B23F7548
       3CE33021A9C72B7025666204C69C0B72
P:
A:     D609B1F056637A0D46DF998D88E5222A
       B2C2846512153524C0895E8108000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       313233340001
IV:    9C002902F801C33EAEA47E08
GCM-AES Authentication
H:     286D73994EA0BA3CFD1F52BF06A8ACF2
Y[0]:  9C002902F801C33EAEA47E0800000001
E(K,Y[0]): 13FBBE38FA1A895C760F543C1AB55F31
X[1]:  BA7C26F578254853CF321281A48317CA
X[2]:  2D0DF59AE78E84ED64C3F85068CD9863
X[3]:  702DE0382ABF4D42DD62B8F115124219
X[4]:  DAED65979342F0D155BFDFE362132078
X[5]:  9AB4AFD6344654B2CD23977E41AA18B3
GHASH(H,A,C): 5E4691528F50E5AB5EC346A7BC264A46
C:
T:     4DBD2F6A754A6CF728CC129BA6931577
```

C.2 Integrity protection (60-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-7. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-7—Unprotected frame (example)

Field	Value
MAC DA	E2 01 06 D7 CD 0D
MAC SA	F0 76 1E 8D CD 3D
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 00 03

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, and the PN. In this example the optional SCI has been omitted. The fields of the protected frame are shown (in the order transmitted) in Table C-8.

Table C-8—Integrity protected frame (example)

Field	Value
MAC DA	E2 01 06 D7 CD 0D
MAC SA	F0 76 1E 8D CD 3D
MACsec EtherType	88 E5
TCI and AN	40
SL	00
PN	76 D4 57 ED
Secure Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 00 03
ICV	Cipher Suite and Key (SAK) dependent (see Table C-9, Table C-10, Table C-11, and Table C-12)

C.2.1 GCM-AES-128 (60-octet frame integrity protection)

Table C-9 specifies an arbitrary 128-bit key (SAK) and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-8. Details of the computation follow the table.

Table C-9—GCM-AES-128 Key and calculated ICV (example)

Field	Value
Key (SAK)	071B113B0CA743FECCCF3D051F737382
ICV	0C 01 7B C7 3B 22 7D FC C9 BA FA 1C 41 AC C3 53

```
key size = 128 bits
P:     0 bits
A:     544 bits
IV:    96 bits
ICV:   128 bits
K:     071B113B0CA743FECCCF3D051F737382
P:
A:     E20106D7CD0DF0761E8DCD3D88E54000
       76D457ED08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A0003
IV:    F0761E8DCD3D000176D457ED
GCM-AES Authentication
H:     E4E01725D724C1215C7309AD34539257
Y[0]:  F0761E8DCD3D000176D457ED00000001
E(K,Y[0]): FC25539100959B80FE3ABED435E54CAB
X[1]:  8DAD4981E33493018BB8482F69E4478C
X[2]:  5B0BFA3E67A3E080CB60EA3D523C734A
X[3]:  051F8D267A68CF88748E56C5F64EF503
X[4]:  4187F1240DB1887F2A92DDAB8903A0F6
X[5]:  C7D64941A90F02FA9FCDECC083B4B276
GHASH(H,A,C): F02428563BB7E67C378044C874498FF8
C:
T:     0C017BC73B227DFCC9BAFA1C41ACC353
```

C.2.2 GCM-AES-256 (60-octet frame integrity protection)

Table C-10 specifies an arbitrary 256-bit key (SAK) and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-8. Details of the computation follow the table.

Table C-10—GCM-AES-256 Key and calculated ICV (example)

Field	Value
Key (SAK)	691D3EE909D7F54167FD1CA0B5D76908 1F2BDE1AEE655FDBAB80BD5295AE6BE7
ICV	35 21 7C 77 4B BC 31 B6 31 66 BC F9 D4 AB ED 07

```
key size = 256 bits
P:     0 bits
A:     544 bits
IV:    96 bits
ICV:   128 bits
K:     691D3EE909D7F54167FD1CA0B5D76908
       1F2BDE1AEE655FDBAB80BD5295AE6BE7
P:
A:     E20106D7CD0DF0761E8DCD3D88E54000
       76D457ED08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A0003
IV:    F0761E8DCD3D000176D457ED
GCM-AES Authentication
H:     1E693C484AB894B26669BC12E6D5D776
Y[0]:  F0761E8DCD3D000176D457ED00000001
E(K,Y[0]): 87E183649AE3E7DBF725659152C39A22
X[1]:  20107B262134C35B60499E905C532004
X[2]:  D7A468F455F09F947884E35A2C80CD7F
X[3]:  A82D607070F2E4470FD94C0EECA9FCC1
X[4]: 03C3C8725883EB355963BD53B515C82D
X[5]: 8FF6F0311DDE274FFA936965C0C905B4
GHASH(H,A,C): B2C0FF13D15FD66DC643D96886687725
C:
T: 35217C774BBC31B63166BCF9D4ABED07
```

C.2.3 GCM-AES-XPN-128 (60-octet frame integrity protection)

Table C-11 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), 96-bit Salt, and 128-bit key (SAK), with the ICV generated by the GCM-AES-XPN-128 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-8.

Table C-11—GCM-AES-XPN-128 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	071B113B0CA743FECCCF3D051F737382
ICV	AB C4 06 85 A3 CF 91 1D 37 87 E4 9D B6 A7 26 5E

```
key size = 128 bits
P:     0 bits
A:     544 bits
IV:    96 bits
ICV:   128 bits
K:     071B113B0CA743FECCCF3D051F737382
P:
A:     E20106D7CD0DF0761E8DCD3D88E54000
       76D457ED08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A0003
IV:    9C002902F801C33E6AB2AD80
GCM-AES Authentication
H:     E4E01725D724C1215C7309AD34539257
Y[0]:  9C002902F801C33E6AB2AD8000000001
E(K,Y[0]): 5BE02ED3987877610007A055C2EEA9A6
X[1]:  8DAD4981E33493018BB8482F69E4478C
X[2]:  5B0BFA3E67A3E080CB60EA3D523C734A
X[3]:  051F8D267A68CF88748E56C5F64EF503
X[4]:  4187F1240DB1887F2A92DDAB8903A0F6
X[5]:  C7D64941A90F02FA9FCDECC083B4B276
GHASH(H,A,C): F02428563BB7E67C378044C874498FF8
C:
T:     ABC40685A3CF911D3787E49DB6A7265E
```

C.2.4 GCM-AES-XPN-256 (60-octet frame integrity protection)

Table C-12 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), a 96-bit Salt, and 256-bit key (SAK), with the ICV generated by the GCM-AES-XPN-256 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-8.

Table C-12—GCM-AES-XPN-256 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	691D3EE909D7F54167FD1CA0B5D76908 1F2BDE1AEE655FDBAB80BD5295AE6BE7
ICV	AC 21 95 7B 83 12 AB 3C 99 AB 46 84 98 79 C3 F3

```
key size = 256 bits
P:     0 bits
A:     544 bits
IV:    96 bits
ICV:   128 bits
K:     691D3EE909D7F54167FD1CA0B5D76908
       1F2BDE1AEE655FDBAB80BD5295AE6BE7
P:
A:     E20106D7CD0DF0761E8DCD3D88E54000
       76D457ED08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A0003
IV:    9C002902F801C33E6AB2AD80
GCM-AES Authentication
H:     1E693C484AB894B26669BC12E6D5D776
Y[0]:  9C002902F801C33E6AB2AD8000000001
E(K,Y[0]): 1EE16A68524D7D515FE89FEC1E11B4D6
X[1]:  20107B262134C35B60499E905C532004
X[2]:  D7A468F455F09F947884E35A2C80CD7F
X[3]:  A82D607070F2E4470FD94C0EECA9FCC1
X[4]:  03C3C8725883EB355963BD53B515C82D
X[5]:  8FF6F0311DDE274FFA936965C0C905B4
GHASH(H,A,C): B2C0FF13D15FD66DC643D96886687725
C:
T: AC21957B8312AB3C99AB46849879C3F3
```

C.3 Integrity protection (65-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-13. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-13—Unprotected frame (example)

Field	Value
MAC DA	84 C5 D5 13 D2 AA
MAC SA	F6 E5 BB D2 72 77
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 00 05

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, the PN, and the (optional) SCI. The fields of the protected frame are shown (in the order transmitted) in Table C-14.

Table C-14—Integrity protected frame (example)

Field	Value
MAC DA	84 C5 D5 13 D2 AA
MAC SA	F6 E5 BB D2 72 77
MACsec EtherType	88 E5
TCI and AN	23
SL	00
PN	89 32 D6 12
SCI	7C FD E9 F9 E3 37 24 C6
Secure Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 00 05
ICV	(see Table C-15, Table C-16, Table C-17, and Table C-18)

C.3.1 GCM-AES-128 (65-octet frame integrity protection)

Table C-15 specifies an arbitrary 128-bit key (SAK) and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-14. Details of the computation follow the table.

Table C-15—GCM-AES-128 Key and calculated ICV (example)

Field	Value
Key (SAK)	013FE00B5F11BE7F866D0CBBC55A7A90
ICV	21 78 67 E5 0C 2D AD 74 C2 8C 3B 50 AB DF 69 5A

```
key size = 128 bits
P:     0 bits
A:     648 bits
IV:    96 bits
ICV:   128 bits
K:     013FE00B5F11BE7F866D0CBBC55A7A90
P:
A:     84C5D513D2AAF6E5BBD2727788E52300
       8932D6127CFDE9F9E33724C608000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       3132333435363738393A3B3C3D3E3F00
       05
IV:    7CFDE9F9E33724C68932D612
GCM-AES Authentication
H:     EB28DCB361EE1110F98CA0C9A07C88F7
Y[0]:  7CFDE9F9E33724C68932D61200000001
E(K,Y[0]): 4EAAF8E4DF948ACAC7F3349C1006A91F
X[1]:  279344E391DB8834EFA68FD3F1BA5CD8
X[2]:  DC35B123F4D387BBB076D0822BD60816
X[3]:  8AB3B52963CC15C9C2DB3E4C801CB65A
X[4]:  CAB6A261225F42578E6B86ABA9F0DD18
X[5]:  6ABDBB3ECAC0458F116A82AA0DAC563F
X[6]:  8F39EF45985C691E35814202B6BB6EF6
GHASH(H,A,C): 6FD29F01D3B927BE057F0FCCBBD9C045
C:
T:     217867E50C2DAD74C28C3B50ABDF695A
```

C.3.2 GCM-AES-256 (65-octet frame integrity protection)

Table C-16 specifies an arbitrary 256-bit key (SAK) and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-14. Details of the computation follow the table.

Table C-16—GCM-AES-256 Key and calculated ICV (example)

Field	Value
Key (SAK)	83C093B58DE7FFE1C0DA926AC43FB360 9AC1C80FEE1B624497EF942E2F79A823
ICV	6E E1 60 E8 FA EC A4 B3 6C 86 B2 34 92 0C A9 75

```
key size = 256 bits
P:     0 bits
A:     648 bits
IV:    96 bits
ICV:   128 bits
K:     83C093B58DE7FFE1C0DA926AC43FB360
       9AC1C80FEE1B624497EF942E2F79A823
P:
A:     84C5D513D2AAF6E5BBD2727788E52300
       8932D6127CFDE9F9E33724C608000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       3132333435363738393A3B3C3D3E3F00
       05
IV:    7CFDE9F9E33724C68932D612
GCM-AES Authentication
H:     D03D3B51FDF2AACB3A165D7DC362D929
Y[0]:  7CFDE9F9E33724C68932D61200000001
E(K,Y[0]): E97EA8EE4455AE79EC4225CAC340E326
X[1]:  22C28F4DF8D09267EA3E11F019F5932C
X[2]:  3D02CFE5FC6A8A9E65B8FFD63E525083
X[3]:  78466AE4A3490819A08645DDC95B143B
X[4]: 6FE4921A6F0A1D5DD90A100A40206142
X[5]: C880DEC2FF2C44F8AD611692AF6D1069
X[6]:  CF4D709A4D020BA876F4371BAA788444
GHASH(H,A,C): 879FC806BEB90ACA80C497FE514C4A53
C:
T: 6EE160E8FAECA4B36C86B234920CA975
```

C.3.3 GCM-AES-XPN-128 (65-octet frame integrity protection)

Table C-17 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), 96-bit Salt, and 128-bit key (SAK), with the ICV generated by the GCM-AES-XPN-128 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-14.

Table C-17—GCM-AES-XPN-128 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	013FE00B5F11BE7F866D0CBBC55A7A90
ICV	67 85 59 B7 E5 2D B0 06 82 E3 B8 30 34 CE BE 59

```
key size = 128 bits
P:     0 bits
A:     648 bits
IV:    96 bits
ICV:   128 bits
K:     013FE00B5F11BE7F866D0CBBC55A7A90
P:
A:     84C5D513D2AAF6E5BBD2727788E52300
       8932D6127CFDE9F9E33724C608000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       3132333435363738393A3B3C3D3E3F00
       05
IV:    9C002902F801C33E95542C7F
GCM-AES Authentication
H:     EB28DCB361EE1110F98CA0C9A07C88F7
Y[0]:  9C002902F801C33E95542C7F00000001
E(K,Y[0]): 0857C6B6369497B8879CB7FC8F177E1C
X[1]:  279344E391DB8834EFA68FD3F1BA5CD8
X[2]:  DC35B123F4D387BBB076D0822BD60816
X[3]:  8AB3B52963CC15C9C2DB3E4C801CB65A
X[4]:  CAB6A261225F42578E6B86ABA9F0DD18
X[5]:  6ABDBB3ECAC0458F116A82AA0DAC563F
X[6]:  8F39EF45985C691E35814202B6BB6EF6
GHASH(H,A,C): 6FD29F01D3B927BE057F0FCCBBD9C045
C:
T:     678559B7E52DB00682E3B83034CEBE59
```

C.3.4 GCM-AES-XPN-256 (65-octet frame integrity protection)

Table C-18 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), a 96-bit Salt, and 256-bit key (SAK), with the ICV generated by the GCM-AES-XPN-256 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-14.

Table C-18—GCM-AES-XPN-256 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	83C093B58DE7FFE1C0DA926AC43FB360 9AC1C80FEE1B624497EF942E2F79A823
ICV	84 BA C8 E5 3D 1E A3 55 A5 C7 D3 34 84 0A E9 62

```
key size = 256 bits
P:     0 bits
A:     648 bits
IV:    96 bits
ICV:   128 bits
K:     83C093B58DE7FFE1C0DA926AC43FB360
       9AC1C80FEE1B624497EF942E2F79A823
P:
A:     84C5D513D2AAF6E5BBD2727788E52300
       8932D6127CFDE9F9E33724C608000F10
       1112131415161718191A1B1C1D1E1F20
       2122232425262728292A2B2C2D2E2F30
       3132333435363738393A3B3C3D3E3F00
       05
IV:    9C002902F801C33E95542C7F
GCM-AES Authentication
H:     D03D3B51FDF2AACB3A165D7DC362D929
Y[0]:  9C002902F801C33E95542C7F00000001
E(K,Y[0]): 032500E383A7A99F250344CAD546A331
X[1]:  22C28F4DF8D09267EA3E11F019F5932C
X[2]:  3D02CFE5FC6A8A9E65B8FFD63E525083
X[3]:  78466AE4A3490819A08645DDC95B143B
X[4]:  6FE4921A6F0A1D5DD90A100A40206142
X[5]:  C880DEC2FF2C44F8AD611692AF6D1069
X[6]:  CF4D709A4D020BA876F4371BAA788444
GHASH(H,A,C): 879FC806BEB90ACA80C497FE514C4A53
C:
T: 84BAC8E53D1EA355A5C7D334840AE962
```

C.4 Integrity protection (79-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-19. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-19—Unprotected frame (example)

Field	Value
MAC DA	68 F2 E7 76 96 CE
MAC SA	7A E8 E2 CA 4E C5
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 00 07

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, and the PN. In this example the optional SCI has been omitted. The fields of the protected frame are shown (in the order transmitted) in Table C-20.

Table C-20—Integrity protected frame (example)

Field	Value
MAC DA	68 F2 E7 76 96 CE
MAC SA	7A E8 E2 CA 4E C5
MACsec EtherType	88 E5
TCI and AN	41
SL	00
PN	2E 58 49 5C
Secure Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 00 07
ICV	(see Table C-21, Table C-22, Table C-23, and Table C-24)

C.4.1 GCM-AES-128 (79-octet frame integrity protection)

Table C-21 specifies an arbitrary 128-bit key (SAK) and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-20. Details of the computation follow the table.

Table C-21—GCM-AES-128 Key and calculated ICV (example)

Field	Value
Key (SAK)	88EE087FD95DA9FBF6725AA9D757B0CD
ICV	07 92 2B 8E BC F1 0B B2 29 75 88 CA 4C 61 45 23

```
key size = 128 bits
P:     0 bits
A:     696 bits
IV:    96 bits
ICV:   128 bits
K:     88EE087FD95DA9FBF6725AA9D757B0CD
P:
A:     68F2E77696CE7AE8E2CA4EC588E54100
       2E58495C08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A3B3C3D3E3F404142434445464748
       494A4B4C4D0007
IV:    7AE8E2CA4EC500012E58495C
GCM-AES Authentication
H:     AE19118C3B704FCE42AE0D15D2C15C7A
Y[0]:  7AE8E2CA4EC500012E58495C00000001
E(K,Y[0]): D2521AABC48C06033E112424D4A6DF74
X[1]:  CA0CAE2BEE8F19845DCB7FE3C5E713AB
X[2]:  5D3F9C7A3BC869457EA5FDFD404A415F
X[3]:  760E6A2873ACC0515D4901B5AC1C85E4
X[4]:  5A40A8425165E3D1978484F07AFC70D8
X[5]:  D9687630FC4436EE582A90A8E4AFC504
X[6]:  311CE361065F86403CDA5DB00798B961
GHASH(H,A,C): D5C03125787D0DB11764ACEE98C79A57
C:
T:     07922B8EBCF10BB2297588CA4C614523
```

C.4.2 GCM-AES-256 (79-octet frame integrity protection)

Table C-22 specifies an arbitrary 256-bit key (SAK) and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-20. Details of the computation follow the table.

Table C-22—GCM-AES-256 Key and calculated ICV (example)

Field	Value
Key (SAK)	4C973DBC7364621674F8B5B89E5C1551 1FCED9216490FB1C1A2CAA0FFE0407E5
ICV	00 BD A1 B7 E8 76 08 BC BF 47 0F 12 15 7F 4C 07

```
key size = 256 bits
P:     0 bits
A:     696 bits
IV:    96 bits
ICV:   128 bits
K:     4C973DBC7364621674F8B5B89E5C1551
       1FCED9216490FB1C1A2CAA0FFE0407E5
P:
A:     68F2E77696CE7AE8E2CA4EC588E54100
       2E58495C08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A3B3C3D3E3F404142434445464748
       494A4B4C4D0007
IV:    7AE8E2CA4EC500012E58495C
GCM-AES Authentication
H:     9A5E559A96459C21E43C0DFF0FA426F3
Y[0]:  7AE8E2CA4EC500012E58495C00000001
E(K,Y[0]): 316F5EDB0829AC9271A6AFF79F3600BF
X[1]:  06A9019B44B76FFEC18978E8B21513E2
X[2]:  89A6401E39EAB6EE5B8159570139F54D
X[3]:  0A5E22BA54F282CE464C334D1AF598EF
X[4]: 4514D8A5C15E15CABC3D2A0E24FC758E
X[5]: 6F98DE3369B88F25AACBF3A993003E78
X[6]:  8183B21C0A932A2D5F598E1B2967564B
GHASH(H,A,C): 31D2FF6CE05FA42ECEE1A0E58A494CB8
C:
T: 00BDA1B7E87608BCBF470F12157F4C07
```

C.4.3 GCM-AES-XPN-128 (79-octet frame integrity protection)

Table C-23 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), 96-bit Salt, and 128-bit key (SAK), with the ICV generated by the GCM-AES-XPN-128 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-20.

Table C-23—GCM-AES-XPN-128 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	88EE087FD95DA9FBF6725AA9D757B0CD
ICV	D0 DC 89 6D C8 37 98 A7 9F 3C 5A 95 BA 3C DF 9A

```
key size = 128 bits
P:     0 bits
A:     696 bits
IV:    96 bits
ICV:   128 bits
K:     88EE087FD95DA9FBF6725AA9D757B0CD
P:
A:     68F2E77696CE7AE8E2CA4EC588E54100
       2E58495C08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A3B3C3D3E3F404142434445464748
       494A4B4C4D0007
IV:    9C002902F801C33E323EB331
GCM-AES Authentication
H:     AE19118C3B704FCE42AE0D15D2C15C7A
Y[0]:  9C002902F801C33E323EB33100000001
E(K,Y[0]): 051CB848B04A95168858F67B22FB45CD
X[1]:  CA0CAE2BEE8F19845DCB7FE3C5E713AB
X[2]:  5D3F9C7A3BC869457EA5FDFD404A415F
X[3]:  760E6A2873ACC0515D4901B5AC1C85E4
X[4]:  5A40A8425165E3D1978484F07AFC70D8
X[5]:  D9687630FC4436EE582A90A8E4AFC504
X[6]:  311CE361065F86403CDA5DB00798B961
GHASH(H,A,C): D5C03125787D0DB11764ACEE98C79A57
C:
T:     D0DC896DC83798A79F3C5A95BA3CDF9A
```

C.4.4 GCM-AES-XPN-256 (79-octet frame integrity protection)

Table C-24 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), a 96-bit Salt, and 256-bit key (SAK), with the ICV generated by the GCM-AES-XPN-256 Cipher Suite when that key is used in conjunction with the foregoing and the frame field data of Table C-20.

Table C-24—GCM-AES-XPN-256 Key and calculated ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	4C973DBC7364621674F8B5B89E5C1551 1FCED9216490FB1C1A2CAA0FFE0407E5
ICV	04 24 9A 20 8A 65 B9 6B 3F 32 63 00 4C FD 86 7D

```
key size = 256 bits
P:     0 bits
A:     696 bits
IV:    96 bits
ICV:   128 bits
K:     4C973DBC7364621674F8B5B89E5C1551
       1FCED9216490FB1C1A2CAA0FFE0407E5
P:
A:     68F2E77696CE7AE8E2CA4EC588E54100
       2E58495C08000F101112131415161718
       191A1B1C1D1E1F202122232425262728
       292A2B2C2D2E2F303132333435363738
       393A3B3C3D3E3F404142434445464748
       494A4B4C4D0007
IV:    9C002902F801C33E323EB331
GCM-AES Authentication
H:     9A5E559A96459C21E43C0DFF0FA426F3
Y[0]:  9C002902F801C33E323EB33100000001
E(K,Y[0]): 35F6654C6A3A1D45F1D3C3E5C6B4CAC5
X[1]:  06A9019B44B76FFEC18978E8B21513E2
X[2]:  89A6401E39EAB6EE5B8159570139F54D
X[3]:  0A5E22BA54F282CE464C334D1AF598EF
X[4]:  4514D8A5C15E15CABC3D2A0E24FC758E
X[5]:  6F98DE3369B88F25AACBF3A993003E78
X[6]:  8183B21C0A932A2D5F598E1B2967564B
GHASH(H,A,C): 31D2FF6CE05FA42ECEE1A0E58A494CB8
C:
T: 04249A208A65B96B3F3263004CFD867D
```

C.5 Confidentiality protection (54-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-25. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-25—Unprotected frame (example)

Field	Value
MAC DA	E2 01 06 D7 CD 0D
MAC SA	F0 76 1E 8D CD 3D
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 00 04

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, and the PN. In this example the optional SCI has been omitted. The fields of the protected frame are shown (in the order transmitted) in Table C-26.

Table C-26—Confidentiality protected frame (example)

Field	Value
MAC DA	E2 01 06 D7 CD 0D
MAC SA	F0 76 1E 8D CD 3D
MACsec EtherType	88 E5
TCI and AN	4C
SL	2A
PN	76 D4 57 ED
Secure Data	Cipher Suite and Key (SAK) dependent (see Table C-27, Table C-28, Table C-29, and Table C-30)
ICV	Cipher Suite and Key (SAK) dependent (see Table C-27, Table C-28, Table C-29, and Table C-30)

C.5.1 GCM-AES-128 (54-octet frame confidentiality protection)

Table C-27 specifies an arbitrary 128-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-25 and Table C-26. The GCM parameter P, the data to be encrypted, is the User Data. The additional data A to be authenticated is formed by concatenating the MAC DA, the MAC SA, and the SecTAG. The SCI and the PN are concatenated (in that order) to form the 96-bit IV used by GCM. The computed GCM parameter T is the ICV. Details of the computation follow the table.

Table C-27—GCM-AES-128 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	071B113B0CA743FECCCF3D051F737382
Secure Data	13 B4 C7 2B 38 9D C5 01 8E 72 A1 71 DD 85 A5 D3 75 22 74 D3 A0 19 FB CA ED 09 A4 25 CD 9B 2E 1C 9B 72 EE E7 C9 DE 7D 52 B3 F3
ICV	D6 A5 28 4F 4A 6D 3F E2 2A 5D 6C 2B 96 04 94 C3

```
key size = 128 bits
P:     336 bits
A:     160 bits
IV:    96 bits
ICV:   128 bits
K:     071B113B0CA743FECCCF3D051F737382
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F30313233340004
A:     E20106D7CD0DF0761E8DCD3D88E54C2A
       76D457ED
IV:    F0761E8DCD3D000176D457ED
GCM-AES Encryption
H:            E4E01725D724C1215C7309AD34539257
Y[0]:         F0761E8DCD3D000176D457ED00000001
E(K,Y[0]):    FC25539100959B80FE3ABED435E54CAB
Y[1]:         F0761E8DCD3D000176D457ED00000002
E(K,Y[1]):    1BB4C83B298FD6159B64B669C49FBECF
C[1]:         13B4C72B389DC5018E72A171DD85A5D3
Y[2]:         F0761E8DCD3D000176D457ED00000003
E(K,Y[2]):    683C6BF3813BD8EEC82F830DE4B10530
C[2]:         752274D3A019FBCAED09A425CD9B2E1C
Y[3]:         F0761E8DCD3D000176D457ED00000004
E(K,Y[3]):    B65CC1D7F8EC4E66B3F7182C2E358591
C[3]:         9B72EEE7C9DE7D52B3F3
X[1]:         A0AE6DFAE25C0AE80E9A1AAC0D5123D3
X[2]:         EAEA2A767986B7D5B9E6ED37A3CBC63B
X[3]:         8809F1263C02DC9BD09FDF0F34575BA6
X[4]:         A173C5A2C03DE08C025C93945B2E74B7
X[5]:         65D113682551614E556BFAA80AA2FA7A
GHASH(H,A,C): 2A807BDE4AF8A462D467D2FFA3E1D868
C:     13B4C72B389DC5018E72A171DD85A5D3
       752274D3A019FBCAED09A425CD9B2E1C
       9B72EEE7C9DE7D52B3F3
T:     D6A5284F4A6D3FE22A5D6C2B960494C3
```

C.5.2 GCM-AES-256 (54-octet frame confidentiality protection)

Table C-28 specifies an arbitrary 256-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-25 and Table C-26. The GCM parameter *P*, the data to be encrypted, is the User Data. The additional data *A* to be authenticated is formed by concatenating the MAC DA, the MAC SA, and the SecTAG. The SCI and the PN are concatenated (in that order) to form the 96-bit *IV* used by GCM. Details of the computation follow the table.

Table C-28—GCM-AES-256 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	691D3EE909D7F54167FD1CA0B5D76908 1F2BDE1AEE655FDBAB80BD5295AE6BE7
Secure Data	C1 62 3F 55 73 0C 93 53 30 97 AD DA D2 56 64 96 61 25 35 2B 43 AD AC BD 61 C5 EF 3A C9 0B 5B EE 92 9C E4 63 0E A7 9F 6C E5 19
ICV	12 AF 39 C2 D1 FD C2 05 1F 8B 7B 3C 9D 39 7E F2

```
key size = 128 bits
P:     336 bits
A:     160 bits
IV:    96 bits
ICV:   128 bits
K:     691D3EE909D7F54167FD1CA0B5D76908
       1F2BDE1AEE655FDBAB80BD5295AE6BE7
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F30313233340004
A:     E20106D7CD0DF0761E8DCD3D88E54C2A
       76D457ED
IV:    F0761E8DCD3D000176D457ED
GCM-AES Encryption
H:            1E693C484AB894B26669BC12E6D5D776
Y[0]:         F0761E8DCD3D000176D457ED00000001
E(K,Y[0]):    87E183649AE3E7DBF725659152C39A22
Y[1]:         F0761E8DCD3D000176D457ED00000002
E(K,Y[1]):    C9623045621E80472581BAC2CB4C7F8A
C[1]:         C1623F55730C93533097ADDAD2566496
Y[2]:         F0761E8DCD3D000176D457ED00000003
E(K,Y[2]):    7C3B2A0B628F8F9944E3C812E02170C2
C[2]:         6125352B43ADACBD61C5EF3AC90B5BEE
Y[3]:         F0761E8DCD3D000176D457ED00000004
E(K,Y[3]):    BFB2CB533F95AC58E51D6608DBEBDBC2
C[3]:         929CE4630EA79F6CE519
X[1]:         F268EF5B38A96261A139D06CD7F43A33
X[2]:         9AE3BF42A20F4FB773EEFD5B5C5DBDD3
X[3]:         22A7FA0F7E5FC49715374D6B72EC7FBB
X[4]:         2FE103C6651C845A71217C1C7E80D559
X[5]:         FA94D93A0A7D235AEED7891F5E381A17
GHASH(H,A,C): 954EBAA64B1E25DEE8AE1EADCFFAE4D0
C:     C1623F55730C93533097ADDAD2566496
       6125352B43ADACBD61C5EF3AC90B5BEE
       929CE4630EA79F6CE519
T:     12AF39C2D1FDC2051F8B7B3C9D397EF2
```

C.5.3 GCM-AES-XPN-128 (54-octet frame confidentiality protection)

Table C-29 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), 96-bit Salt, and 128-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-128 Cipher Suite when used in conjunction with the foregoing and the frame field data of Table C-25 and Table C-26. The GCM parameter *P*, the data to be encrypted, is the User Data. The additional data *A* to be authenticated is formed by concatenating the MAC DA, the MAC SA, and the SecTAG. The computed GCM parameter *T* is the ICV.

Table C-29—GCM-AES-XPN-128 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	071B113B0CA743FECCCF3D051F737382
Secure Data	9C A4 69 84 43 02 03 ED 41 6E BD C2 FE 26 22 BA 3E 5E AB 69 61 C3 63 83 00 9E 18 7E 9B 0C 88 56 46 53 B9 AB D2 16 44 1C 6A B6
ICV	F0 A2 32 E9 E4 4C 97 8C F7 CD 84 D4 34 84 D1 01

```
key size = 128 bits
P:     336 bits
A:     160 bits
IV:    96 bits
ICV:   128 bits
K:     071B113B0CA743FECCCF3D051F737382
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F30313233340004
A:     E20106D7CD0DF0761E8DCD3D88E54C2A
       76D457ED
IV:    9C002902F801C33E6AB2AD80
GCM-AES Encryption
H:            E4E01725D724C1215C7309AD34539257
Y[0]:         9C002902F801C33E6AB2AD8000000001
E(K,Y[0]):    5BE02ED3987877610007A055C2EEA9A6
Y[1]:         9C002902F801C33E6AB2AD8000000002
E(K,Y[1]):    94A46694521010F95478AADAE73C39A6
C[1]:         9CA46984430203ED416EBDC2FE2622BA
Y[2]:         9C002902F801C33E6AB2AD8000000003
E(K,Y[2]):    2340B44940E140A725B83F56B226A37A
C[2]:         3E5EAB6961C36383009E187E9B0C8856
Y[3]:         9C002902F801C33E6AB2AD8000000004
E(K,Y[3]):    6B7D969BE32477286AB2194F5E91341E
C[3]:         4653B9ABD216441C6AB6
X[1]:         A0AE6DFAE25C0AE80E9A1AAC0D5123D3
X[2]:         EAEA2A767986B7D5B9E6ED37A3CBC63B
X[3]:         2B263EA98B4A3CDAC1039172AD286472
X[4]:         7A8F8EDACACADACB31FC58F3C1750828
X[5]:         3B9CA06E32AF0F5A1F6A49B5F38EC6C0
GHASH(H,A,C): AB421C3A7C34E0EDF7CA2481F66A78A7
C:     9CA46984430203ED416EBDC2FE2622BA
       3E5EAB6961C36383009E187E9B0C8856
       4653B9ABD216441C6AB6
T:     F0A232E9E44C978CF7CD84D43484D101
```

C.5.4 GCM-AES-XPN-256 (54-octet frame confidentiality protection)

Table C-30 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN (the 32 least significant bits are those of the PN field in the SecTAG), a 96-bit Salt, and 256-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-256 Cipher Suite when used in conjunction with the foregoing and the frame field data of Table C-25 and Table C-26. The GCM parameter *P*, the data to be encrypted, is the User Data. The additional data *A* to be authenticated is formed by concatenating the MAC DA, the MAC SA, and the SecTAG. The computed GCM parameter *T* is the ICV.

Table C-30—GCM-AES-XPN-256 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	691D3EE909D7F54167FD1CA0B5D76908 1F2BDE1AEE655FDBAB80BD5295AE6BE7
Secure Data	88 D9 F7 D1 F1 57 8E E3 4B A7 B1 AB C8 98 93 EF 1D 33 98 C9 F1 DD 3E 47 FB D8 55 3E 0F F7 86 EF 56 99 EB 01 EA 10 42 0D 0E BD
ICV	39 A0 E2 73 C4 C7 F9 5E D8 43 20 7D 7A 49 7D FA

```
key size = 256 bits        P:     336 bits      A:     160 bits
IV:    96 bits             ICV:   128 bits
K:     691D3EE909D7F54167FD1CA0B5D76908
       1F2BDE1AEE655FDBAB80BD5295AE6BE7
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F30313233340004
A:     E20106D7CD0DF0761E8DCD3D88E54C2A
       76D457ED
IV:    9C002902F801C33E6AB2AD80
GCM-AES Encryption
H:             1E693C484AB894B26669BC12E6D5D776
Y[0]:          9C002902F801C33E6AB2AD8000000001
E(K,Y[0]):     1EE16A68524D7D515FE89FEC1E11B4D6
Y[1]:          9C002902F801C33E6AB2AD8000000002
E(K,Y[1]):     80D9F8C1E0459DF75EB1A6B3D18288F3
C[1]:          88D9F7D1F1578EE34BA7B1ABC89893EF
Y[2]:          9C002902F801C33E6AB2AD8000000003
E(K,Y[2]):     002D87E9D0FF1D63DEFE721626DDADC3
C[2]:          1D3398C9F1DD3E47FBD8553E0FF786EF
Y[3]:          9C002902F801C33E6AB2AD8000000004
E(K,Y[3]):     7BB7C431DB2271390EB9457F85679B03
C[3]:          5699EB01EA10420D0EBD
X[1]:          F268EF5B38A96261A139D06CD7F43A33
X[2]:          9AE3BF42A20F4FB773EEFD5B5C5DBDD3
X[3]:          A4C4D446D50875C031E49F1039DA6E86
X[4]:          CA6D1BC70BFCAF8B7D1D2279E57D6B45
X[5]:          A76805EBCFDBB50AB9D9C198701753C0
GHASH(H,A,C): 2741881B968A840F87ABBF916458C92C
C:     88D9F7D1F1578EE34BA7B1ABC89893EF
       1D3398C9F1DD3E47FBD8553E0FF786EF
       5699EB01EA10420D0EBD
T:     39A0E273C4C7F95ED843207D7A497DFA
```

C.6 Confidentiality protection (60-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-31. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-31—Unprotected frame (example)

Field	Value
MAC DA	D6 09 B1 F0 56 63
MAC SA	7A 0D 46 DF 99 8D
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 00 02

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, the PN, and the (optional) SCI. The fields of the protected frame are shown (in the order transmitted) in Table C-32.

Table C-32—Confidentiality protected frame (example)

Field	Value
MAC DA	D6 09 B1 F0 56 63
MAC SA	7A 0D 46 DF 99 8D
MACsec EtherType	88 E5
TCI and AN	2E
SL	00
PN	B2 C2 84 65
SCI	12 15 35 24 C0 89 5E 81
Secure Data	Cipher Suite and Key (SAK) dependent (see Table C-33, Table C-34, Table C-35, and Table C-36)
ICV	Cipher Suite and Key (SAK) dependent (see Table C-33, Table C-34, Table C-35, and Table C-36)

C.6.1 GCM-AES-128 (60-octet frame confidentiality protection)

Table C-33 specifies an arbitrary 128-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-31 and Table C-32. Details of the computation follow the table.

Table C-33—GCM-AES-128 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	AD7A2BD03EAC835A6F620FDCB506B345
Secure Data	70 1A FA 1C C0 39 C0 D7 65 12 8A 66 5D AB 69 24 38 99 BF 73 18 CC DC 81 C9 93 1D A1 7F BE 8E DD 7D 17 CB 8B 4C 26 FC 81 E3 28 4F 2B 7F BA 71 3D
ICV	4F 8D 55 E7 D3 F0 6F D5 A1 3C 0C 29 B9 D5 B8 80

```
key size = 128 bits
P:     384 bits
A:     224 bits
IV:    96 bits
ICV:   128 bits
K:     AD7A2BD03EAC835A6F620FDCB506B345
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A0002
A:     D609B1F056637A0D46DF998D88E52E00
       B2C2846512153524C0895E81
IV:    12153524C0895E81B2C28465
GCM-AES Encryption
H:            73A23D80121DE2D5A850253FCF43120E
Y[0]:         12153524C0895E81B2C2846500000001
E(K,Y[0]):    EB4E051CB548A6B5490F6F11A27CB7D0
Y[1]:         12153524C0895E81B2C2846500000002
E(K,Y[1]):    781AF50CD12BD3C370049D7E44B17238
C[1]:         701AFA1CC039C0D765128A665DAB6924
Y[2]:         12153524C0895E81B2C2846500000003
E(K,Y[2]):    2587A05339EEFFA5ECB53A895694A5F1
C[2]:         3899BF7318CCDC81C9931DA17FBE8EDD
Y[3]:         12153524C0895E81B2C2846500000004
E(K,Y[3]):    5039E4BB7D14CFB5D61E78134680713F
C[3]:         7D17CB8B4C26FC81E3284F2B7FBA713D
X[1]:         9CABBD91899C1413AA7AD629C1DF12CD
X[2]:         B99ABF6BDBD18B8E148F8030F0686F28
X[3]:         8B5BD74B9A65A459150392C3872BCE7F
X[4]:         934E9D58C59230EE652675D0FF4FB255
X[5]:         4738D208B10FAFF24D6DFBDDC916DC44
GHASH(H,A,C): A4C350FB66B8C960E83363381BA90F50
C:     701AFA1CC039C0D765128A665DAB6924
       3899BF7318CCDC81C9931DA17FBE8EDD
       7D17CB8B4C26FC81E3284F2B7FBA713D
T:     4F8D55E7D3F06FD5A13C0C29B9D5B880
```

C.6.2 GCM-AES-256 (60-octet frame confidentiality protection)

Table C-34 specifies an arbitrary 256-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-31 and Table C-32. Details of the computation follow the table.

Table C-34—GCM-AES-256 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	E3C08A8F06C6E3AD95A70557B23F7548 3CE33021A9C72B7025666204C69C0B72
Secure Data	E2 00 6E B4 2F 52 77 02 2D 9B 19 92 5B C4 19 D7 A5 92 66 6C 92 5F E2 EF 71 8E B4 E3 08 EF EA A7 C5 27 3B 39 41 18 86 0A 5B E2 A9 7F 56 AB 78 36
ICV	5C A5 97 CD BB 3E DB 8D 1A 11 51 EA 0A F7 B4 36

```
key size = 256 bits
P:     384 bits
A:     224 bits
IV:    96 bits
ICV:   128 bits
K:     E3C08A8F06C6E3AD95A70557B23F7548
       3CE33021A9C72B7025666204C69C0B72
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A0002
A:     D609B1F056637A0D46DF998D88E52E00
       B2C2846512153524C0895E81
IV:    12153524C0895E81B2C28465
GCM-AES Encryption
H:            286D73994EA0BA3CFD1F52BF06A8ACF2
Y[0]:         12153524C0895E81B2C2846500000001
E(K,Y[0]):    714D54FDCFCEE37D5729CDDAB383A016
Y[1]:         12153524C0895E81B2C2846500000002
E(K,Y[1]):    EA0061A43E406416388D0E8A42DE02CB
C[1]:         E2006EB42F5277022D9B19925BC419D7
Y[2]:         12153524C0895E81B2C2846500000003
E(K,Y[2]):    B88C794CB37DC1CB54A893CB21C5C18B
C[2]:         A592666C925FE2EF718EB4E308EFEAA7
Y[3]:         12153524C0895E81B2C2846500000004
E(K,Y[3]):    E8091409702AB53E6ED49E476F917834
C[3]:         C5273B394118860A5BE2A97F56AB7836
X[1]:         D62D2B0792C282A27B82C3731ABCB7A1
X[2]:         841068CDEDA878030E644F03743927D0
X[3]:         224CE5247BE62FB2AC5932EFAC5D1991
X[4]:         EB66718E589AB6472880D1A2C908CB72
X[5]:         6D109A3C7F34085754FDDFF0EB5D4595
GHASH(H,A,C): 2DE8C33074F038F04D389C30B9741420
C:     E2006EB42F5277022D9B19925BC419D7
       A592666C925FE2EF718EB4E308EFEAA7
       C5273B394118860A5BE2A97F56AB7836
T:     5CA597CDBB3EDB8D1A1151EA0AF7B436
```

C.6.3 GCM-AES-XPN-128 (60-octet frame confidentiality protection)

Table C-35 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN, 96-bit Salt, and 128-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-128 Cipher Suite when used with the frame field data of Table C-31 and Table C-32.

Table C-35—GCM-AES-XPN-128 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	AD7A2BD03EAC835A6F620FDCB506B345
Secure Data	07 12 D9 80 CA 50 BB ED 35 A0 FA 56 63 38 72 9F FA 16 D1 9F FC F0 7B 3A 1E 79 19 B3 77 6A AC EC 8A 59 37 20 8B 48 3A 76 91 98 4D 38 07 92 E0 7F
ICV	C2 C3 C7 9F 26 3F A6 BF F8 E7 58 1E 2C E4 5A F8

```
key size = 128 bits
P:     384 bits
A:     224 bits
IV:    96 bits
ICV:   128 bits
K:     AD7A2BD03EAC835A6F620FDCB506B345
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A0002
A:     D609B1F056637A0D46DF998D88E52E00
       B2C2846512153524C0895E81
IV:    9C002902F801C33EAEA47E08
GCM-AES Encryption
H:            73A23D80121DE2D5A850253FCF43120E
Y[0]:         9C002902F801C33EAEA47E0800000001
E(K,Y[0]):    0C246434EE05EB99762BEFD9880C9E2E
Y[1]:         9C002902F801C33EAEA47E0800000002
E(K,Y[1]):    0F12D690DB42A8F920B6ED4E7A226983
C[1]:         0712D980CA50BBED35A0FA566338729F
Y[2]:         9C002902F801C33EAEA47E0800000003
E(K,Y[2]):    E708CEBFDDD2581E3B5F3E9B5E4087C0
C[2]:         FA16D19FFCF07B3A1E7919B3776AACEC
Y[3]:         9C002902F801C33EAEA47E0800000004
E(K,Y[3]):    A7771810BA7A0942A4AE7A003EA8E07D
C[3]:         8A5937208B483A7691984D380792E07F
X[1]:         9CABBD91899C1413AA7AD629C1DF12CD
X[2]:         B99ABF6BDBD18B8E148F8030F0686F28
X[3]:         D51D27D562BEC296CAA7989F501D7438
X[4]:         ED6EBBA57C47B9A98F94037EAA603CF7
X[5]:         09D09142DBDE8105AB991E09076D6399
GHASH(H,A,C): CEE7A3ABC83A4D268ECCB7C7A4E8C4D6
C:     0712D980CA50BBED35A0FA566338729F
       FA16D19FFCF07B3A1E7919B3776AACEC
       8A5937208B483A7691984D380792E07F
T:     C2C3C79F263FA6BFF8E7581E2CE45AF8
```

C.6.4 GCM-AES-XPN-256 (60-octet frame confidentiality protection)

Table C-36 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN, 96-bit Salt, and 256-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-256 Cipher Suite when used with the frame field data of Table C-31 and Table C-32.

Table C-36—GCM-AES-XPN-256 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	E3C08A8F06C6E3AD95A70557B23F7548 3CE33021A9C72B7025666204C69C0B72
Secure Data	3E B0 4A 4B BF 54 C6 EB 12 22 A9 AE A0 0C 38 68 7F 6C 35 20 D9 76 A3 B6 94 80 06 50 CE 65 85 E6 20 A4 19 19 17 D2 A6 05 D8 70 C7 8D 27 52 CE 49
ICV	3B 44 2A C0 C8 16 D7 AB D7 0A D6 5C 25 D4 64 13

```
key size = 256 bits
P:     384 bits
A:     224 bits
IV:    96 bits
ICV:   128 bits
K:     E3C08A8F06C6E3AD95A70557B23F7548
       3CE33021A9C72B7025666204C69C0B72
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A0002
A:     D609B1F056637A0D46DF998D88E52E00
       B2C2846512153524C0895E81
IV:    9C002902F801C33EAEA47E08
GCM-AES Encryption
H:            286D73994EA0BA3CFD1F52BF06A8ACF2
Y[0]:         9C002902F801C33EAEA47E0800000001
E(K,Y[0]):    13FBBE38FA1A895C760F543C1AB55F31
Y[1]:         9C002902F801C33EAEA47E0800000002
E(K,Y[1]):    36B0455BAE46D5FF0734BEB6B9162374
C[1]:         3EB04A4BBF54C6EB1222A9AEA00C3868
Y[2]:         9C002902F801C33EAEA47E0800000003
E(K,Y[2]):    62722A00F8548092B1A62178E74FAECA
C[2]:         7F6C3520D976A3B694800650CE6585E6
Y[3]:         9C002902F801C33EAEA47E0800000004
E(K,Y[3]):    0D8A362926E09531ED46F0B51E68CE4B
C[3]:         20A4191917D2A605D870C78D2752CE49
X[1]:         D62D2B0792C282A27B82C3731ABCB7A1
X[2]:         841068CDEDA878030E644F03743927D0
X[3]:         EF4AD8D95E4309D95F8E5F73533BAED7
X[4]:         152F572D6A56D8F0F70E77BE99BAC80D
X[5]:         A9BE13F28CD8F4CCFF1870E5EBB5A9D6
GHASH(H,A,C): 28BF94F8320C5EF7A10582603F613B22
C:     3EB04A4BBF54C6EB1222A9AEA00C3868
       7F6C3520D976A3B694800650CE6585E6
       20A4191917D2A605D870C78D2752CE49
T:     3B442AC0C816D7ABD70AD65C25D46413
```

C.7 Confidentiality protection (61-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-37. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-37—Unprotected frame (example)

Field	Value
MAC DA	84 C5 D5 13 D2 AA
MAC SA	F6 E5 BB D2 72 77
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 00 06

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, the PN, and the (optional) SCI. The fields of the protected frame are shown (in the order transmitted) in Table C-38.

Table C-38—Confidentiality protected frame (example)

Field	Value
MAC DA	84 C5 D5 13 D2 AA
MAC SA	F6 E5 BB D2 72 77
MACsec EtherType	88 E5
TCI and AN	2F
SL	00
PN	89 32 D6 12
SCI	7C FD E9 F9 E3 37 24 C6
Secure Data	Cipher Suite and Key (SAK) dependent (see Table C-39, Table C-40, Table C-41, and Table C-42)
ICV	Cipher Suite and Key (SAK) dependent (see Table C-39, Table C-40, Table C-41, and Table C-42)

C.7.1 GCM-AES-128 (61-octet frame confidentiality protection)

Table C-39 specifies an arbitrary 128-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-37 and Table C-38. Details of the computation follow the table.

Table C-39—GCM-AES-128 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	013FE00B5F11BE7F866D0CBBC55A7A90
Secure Data	3A 4D E6 FA 32 19 10 14 DB B3 03 D9 2E E3 A9 E8 A1 B5 99 C1 4D 22 FB 08 00 96 E1 38 11 81 6A 3C 9C 9B CF 7C 1B 9B 96 DA 80 92 04 E2 9D 0E 2A 76 42
ICV	BF D3 10 A4 83 7C 81 6C CF A5 AC 23 AB 00 39 88

```
key size = 128 bits
P:     392 bits
A:     224 bits
IV:    96 bits
ICV:   128 bits
K:     013FE00B5F11BE7F866D0CBBC55A7A90
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B00
       06
A:     84C5D513D2AAF6E5BBD2727788E52F00
       8932D6127CFDE9F9E33724C6
IV:    7CFDE9F9E33724C68932D612
GCM-AES Encryption
H:            EB28DCB361EE1110F98CA0C9A07C88F7
Y[0]:         7CFDE9F9E33724C68932D61200000001
E(K,Y[0]):    4EAAF8E4DF948ACAC7F3349C1006A91F
Y[1]:         7CFDE9F9E33724C68932D61200000002
E(K,Y[1]):    324DE9EA230B0300CEA514C137F9B2F4
C[1]:         3A4DE6FA32191014DBB303D92EE3A9E8
Y[2]:         7CFDE9F9E33724C68932D61200000003
E(K,Y[2]):    BCAB86E16C00D82C25B0C61038AB4110
C[2]:         A1B599C14D22FB080096E13811816A3C
Y[3]:         7CFDE9F9E33724C68932D61200000004
E(K,Y[3]):    B1B5E04C2AA9A5EEB5A433DAA4341176
C[3]:         9C9BCF7C1B9B96DA809204E29D0E2A76
Y[4]:         7CFDE9F9E33724C68932D61200000005
E(K,Y[4]):    44491285F0FCF957EB73F79AC5D4E273
C[4]:         42
X[1]:         BA7749648FCB954F95B5933AC87D5AA3
X[2]:         A78C78463850956BF8939E6D8314DED1
X[3]:         18EB5A2C2541C14DD668468C26D2CD8A
X[4]:         32C49AA9AD2B7025767B14F37740A2E8
X[5]:         59CEE3A487F7ACAA9531883B31B11561
X[6]:         3FC125EEEC404708A0D8B9998FE0DE9B
GHASH(H,A,C): F179E8405CE80BA6085698BFBB069097
C:     3A4DE6FA32191014DBB303D92EE3A9E8
       A1B599C14D22FB080096E13811816A3C
       9C9BCF7C1B9B96DA809204E29D0E2A76
       42
T:     BFD310A4837C816CCFA5AC23AB003988
```

C.7.2 GCM-AES-256 (61-octet frame confidentiality protection)

Table C-40 specifies an arbitrary 256-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-37 and Table C-38. Details of the computation follow the table.

Table C-40—GCM-AES-256 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	83C093B58DE7FFE1C0DA926AC43FB360 9AC1C80FEE1B624497EF942E2F79A823
Secure Data	11 02 22 FF 80 50 CB EC E6 6A 81 3A D0 9A 73 ED 7A 9A 08 9C 10 6B 95 93 89 16 8E D6 E8 69 8E A9 02 EB 12 77 DB EC 2E 68 E4 73 15 5A 15 A7 DA EE D4
ICV	A1 0F 4E 05 13 9C 23 DF 00 B3 AA DC 71 F0 59 6A

```
key size = 256 bits
P:     392 bits
A:     224 bits
IV:    96 bits
ICV:   128 bits
K:     83C093B58DE7FFE1C0DA926AC43FB360
       9AC1C80FEE1B624497EF942E2F79A823
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B00
       06
A:     84C5D513D2AAF6E5BBD2727788E52F00
       8932D6127CFDE9F9E33724C6
IV:    7CFDE9F9E33724C68932D612
GCM-AES Encryption
H:             D03D3B51FDF2AACB3A165D7DC362D929
Y[0]:          7CFDE9F9E33724C68932D61200000001
E(K,Y[0]):     E97EA8EE4455AE79EC4225CAC340E326
Y[1]:          7CFDE9F9E33724C68932D61200000002
E(K,Y[1]):     19022DEF9142D8F8F37C9622C98068F1
C[1]:          110222FF8050CBECE66A813AD09A73ED
Y[2]:          7CFDE9F9E33724C68932D61200000003
E(K,Y[2]):     678417BC3149B6B7AC30A9FEC143A585
C[2]:          7A9A089C106B959389168ED6E8698EA9
Y[3]:          7CFDE9F9E33724C68932D61200000004
E(K,Y[3]):     2FC53D47EADE1D5CD14522622C9DE1EE
C[3]:          02EB1277DBEC2E68E473155A15A7DAEE
Y[4]:          7CFDE9F9E33724C68932D61200000005
E(K,Y[4]):     D2541F9E6E5ABAB19C0341912287646B
C[4]:          D4
X[1]:          0B75EC495656426640FD4E24ABA3ED1E
X[2]:          4BC3618F5864A86E9F4EE84504DE347C
X[3]:          F67E393EC69D2D6FFD54C4EFA6F5FF88
X[4]:          C7FE302C946CC29D1EFAAA22B7F587DD
X[5]:          87FCCA374A2EAFC6FD08FE08F919FB8E
X[6]:          0A648461F8E051A0B03165459D5E6F59
GHASH(H,A,C): 4871E6EB57C98DA6ECF18F16B2B0BA4C
C:     110222FF8050CBECE66A813AD09A73ED
       7A9A089C106B959389168ED6E8698EA9
       02EB1277DBEC2E68E473155A15A7DAEE
       D4
T:     A10F4E05139C23DF00B3AADC71F0596A
```

C.7.3 GCM-AES-XPN-128 (61-octet frame confidentiality protection)

Table C-41 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN, 96-bit Salt, and 128-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-128 Cipher Suite when used with the frame field data of Table C-37 and Table C-38.

Table C-41—GCM-AES-XPN-128 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	013FE00B5F11BE7F866D0CBBC55A7A90
Secure Data	14 C1 76 93 BC 82 97 EE 6C 47 C5 65 CB E0 67 9E 80 F0 0F CA F5 92 C9 AA 04 73 92 8E 7F 2F 21 6F F5 A0 33 DE C7 51 3F 45 D3 4C BB 98 1C 5B D6 4E 8B
ICV	D8 4B 8E 2A 78 E7 4D AF EA A0 38 46 FE 93 0C 0E

```
key size = 128 bits        P:     392 bits      A:     224 bits
IV:    96 bits             ICV:   128 bits
K:     013FE00B5F11BE7F866D0CBBC55A7A90
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B00
       06
A:     84C5D513D2AAF6E5BBD2727788E52F00
       8932D6127CFDE9F9E33724C6
IV:    9C002902F801C33E95542C7F
GCM-AES Encryption
H:            EB28DCB361EE1110F98CA0C9A07C88F7
Y[0]:         9C002902F801C33E95542C7F00000001
E(K,Y[0]):    0857C6B6369497B8879CB7FC8F177E1C
Y[1]:         9C002902F801C33E95542C7F00000002
E(K,Y[1]):    1CC17983AD9084FA7951D27DD2FA7C82
C[1]:         14C17693BC8297EE6C47C565CBE0679E
Y[2]:         9C002902F801C33E95542C7F00000003
E(K,Y[2]):    9DEE10EAD4B0EA8E2155B5A656050A43
C[2]:         80F00FCAF592C9AA0473928E7F2F216F
Y[3]:         9C002902F801C33E95542C7F00000004
E(K,Y[3]):    D88E1CEEF6630C71E67A8CA02561ED4E
C[3]:         F5A033DEC7513F45D34CBB981C5BD64E
Y[4]:         9C002902F801C33E95542C7F00000005
E(K,Y[4]):    8D2F0332C7B929F6A40244B1750EDD0A
C[4]:         8B
X[1]:         BA7749648FCB954F95B5933AC87D5AA3
X[2]:         A78C78463850956BF8939E6D8314DED1
X[3]:         A751317726E0F1E84315A9C743DF0C4F
X[4]:         767D6E085166E75CAEAB804D781C2415
X[5]:         5047BCF3D97EADA35994813E3373B800
X[6]:         FA31D67AA9192FB24E5491D4FE366987
GHASH(H,A,C): D01C489C4E73DA176D3C8FBA71847212
C:     14C17693BC8297EE6C47C565CBE0679E
       80F00FCAF592C9AA0473928E7F2F216F
       F5A033DEC7513F45D34CBB981C5BD64E
       8B
T:     D84B8E2A78E74DAFEAA03846FE930C0E
```

C.7.4 GCM-AES-XPN-256 (61-octet frame confidentiality protection)

Table C-42 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN, 96-bit Salt, and 256-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-256 Cipher Suite when used with the frame field data of Table C-37 and Table C-38.

Table C-42—GCM-AES-XPN-256 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	83C093B58DE7FFE1C0DA926AC43FB360 9AC1C80FEE1B624497EF942E2F79A823
Secure Data	09 96 E0 C9 A5 57 74 E0 A7 92 30 4E 7D C1 50 BD 67 FD 74 7D D1 B9 41 95 94 BF 37 3D 4A CE 8F 87 F5 C1 34 9A FA C4 91 AA 0A 40 D3 19 90 87 B2 9F DF
ICV	80 2F 05 0E 69 1F 11 A2 D9 B3 58 F6 99 41 84 F5

```
key size = 256 bits        P:     392 bits      A:     224 bits
IV:    96 bits             ICV:   128 bits
K:     83C093B58DE7FFE1C0DA926AC43FB360
       9AC1C80FEE1B624497EF942E2F79A823
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B00
       06
A:     84C5D513D2AAF6E5BBD2727788E52F00
       8932D6127CFDE9F9E33724C6
IV:    9C002902F801C33E95542C7F
GCM-AES Encryption
H:             D03D3B51FDF2AACB3A165D7DC362D929
Y[0]:          9C002902F801C33E95542C7F00000001
E(K,Y[0]):     032500E383A7A99F250344CAD546A331
Y[1]:          9C002902F801C33E95542C7F00000002
E(K,Y[1]):     0196EFD9B44567F4B284275664DB4BA1
C[1]:          0996E0C9A55774E0A792304E7DC150BD
Y[2]:          9C002902F801C33E95542C7F00000003
E(K,Y[2]):     7AE36B5DF09B62B1B199101563E4A4AB
C[2]:          67FD747DD1B9419594BF373D4ACE8F87
Y[3]:          9C002902F801C33E95542C7F00000004
E(K,Y[3]):     D8EF1BAACBF6A29E3F76E421A9BD899F
C[3]:          F5C1349AFAC491AA0A40D3199087B29F
Y[4]:          9C002902F801C33E95542C7F00000005
E(K,Y[4]):     D965B16B455F2E1F14A50977DF3CAB5E
C[4]:          DF
X[1]:          0B75EC495656426640FD4E24ABA3ED1E
X[2]:          4BC3618F5864A86E9F4EE84504DE347C
X[3]:          CBBE203CFB8356BE9E2454FFE5A1C9AA
X[4]:          78CD005E32EC7ECB0BDFF959C34FA917
X[5]:          C2B1847755117D0352CD4E6C77FF618C
X[6]:          8CA987756B41595D96458142BBBF57C3
GHASH(H,A,C): 830A05EDEAB8B83DFCB01C3C4C0727C4
C:     0996E0C9A55774E0A792304E7DC150BD
       67FD747DD1B9419594BF373D4ACE8F87
       F5C1349AFAC491AA0A40D3199087B29F
       DF
T:     802F050E691F11A2D9B358F6994184F5
```

C.8 Confidentiality protection (75-octet frame)

The MAC Destination Address, MAC Source Address, and MAC Service Data Unit (MSDU, User Data) of a MAC Service data request and a corresponding data indication are shown in Table C-43. These comprise the octets of an unprotected frame when concatenated in the order given (with the addition of any media dependent additional fields such as padding). The User Data shown includes the IP EtherType.

Table C-43—Unprotected frame (example)

Field	Value
MAC DA	68 F2 E7 76 96 CE
MAC SA	7A E8 E2 CA 4E C5
User Data	08 00 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 00 08

The MAC Security TAG (SecTAG) comprises the MACsec EtherType, the TCI, the AN, the SL, and the PN. The optional SCI has been omitted. The fields of the protected frame are shown (in the order transmitted) in Table C-44.

Table C-44—Confidentiality protected frame (example)

Field	Value
MAC DA	68 F2 E7 76 96 CE
MAC SA	7A E8 E2 CA 4E C5
MACsec EtherType	88 E5
TCI and AN	4D
SL	00
PN	2E 58 49 5C
Secure Data	Cipher Suite and Key (SAK) dependent (see Table C-45, Table C-46, Table C-47, and Table C-48)
ICV	Cipher Suite and Key (SAK) dependent (see Table C-45, Table C-46, Table C-47, and Table C-48)

C.8.1 GCM-AES-128 (75-octet frame confidentiality protection)

Table C-45 specifies an arbitrary 128-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-128 Cipher Suite when that key is used in conjunction with the frame field data of Table C-43 and Table C-44. Details of the computation follow the table.

Table C-45—GCM-AES-128 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	88EE087FD95DA9FBF6725AA9D757B0CD
Secure Data	C3 1F 53 D9 9E 56 87 F7 36 51 19 B8 32 D2 AA E7 07 41 D5 93 F1 F9 E2 AB 34 55 77 9B 07 8E B8 FE AC DF EC 1F 8E 3E 52 77 F8 18 0B 43 36 1F 65 12 AD B1 6D 2E 38 54 8A 2C 71 9D BA 72 28 D8 40
ICV	88 F8 75 7A DB 8A A7 88 D8 F6 5A D6 68 BE 70 E7

```
key size = 128 bits
P:     504 bits
A:     160 bits
IV:    96 bits
ICV:   128 bits
K:     88EE087FD95DA9FBF6725AA9D757B0CD
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B3C
       3D3E3F404142434445464748490008
A:     68F2E77696CE7AE8E2CA4EC588E54D00
       2E58495C
IV:    7AE8E2CA4EC500012E58495C
GCM-AES Encryption
H:            AE19118C3B704FCE42AE0D15D2C15C7A
Y[0]:         7AE8E2CA4EC500012E58495C00000001
E(K,Y[0]):    D2521AABC48C06033E112424D4A6DF74
Y[1]:         7AE8E2CA4EC500012E58495C00000002
E(K,Y[1]):    CB1F5CC98F4494E323470EA02BC8B1FB
C[1]:         C31F53D99E5687F7365119B832D2AAE7
Y[2]:         7AE8E2CA4EC500012E58495C00000003
E(K,Y[2]):    1A5FCAB3D0DBC18F117350B32EA493D2
C[2]:         0741D593F1F9E2AB3455779B078EB8FE
Y[3]:         7AE8E2CA4EC500012E58495C00000004
E(K,Y[3]):    81F1C32FBF0C6143CD2E3C7B0F255E2E
C[3]:         ACDFEC1F8E3E5277F8180B43361F6512
Y[4]:         7AE8E2CA4EC500012E58495C00000005
E(K,Y[4]):    908F526E7916C96834DBFD3A61D848B2
C[4]:         ADB16D2E38548A2C719DBA7228D840
X[1]:         A9845CAED3E164079E217A8D26A600DA
X[2]:         09410740B1204002F754119A976F31C8
X[3]:         CB897D3B71442B121E77CEA5416D3931
X[4]:         5F3A6A2D049FF2337096523ECAA1BD30
X[5]:         0C95908AEEBDAF1B1C279837AE498000
X[6]:         1ACA99E1E46D2395BC610D21BB4216A0
GHASH(H,A,C): 5AAA6FD11F06A18BE6E77EF2BC18AF93
C:     C31F53D99E5687F7365119B832D2AAE7
       0741D593F1F9E2AB3455779B078EB8FE
       ACDFEC1F8E3E5277F8180B43361F6512
       ADB16D2E38548A2C719DBA7228D840
T:     88F8757ADB8AA788D8F65AD668BE70E7
```

C.8.2 GCM-AES-256 (75-octet frame confidentiality protection)

Table C-46 specifies an arbitrary 256-bit key (SAK), the Secure Data, and the ICV generated by the GCM-AES-256 Cipher Suite when that key is used in conjunction with the frame field data of Table C-43 and Table C-44. Details of the computation follow the table.

Table C-46—GCM-AES-256 Key, Secure Data, and ICV (example)

Field	Value
Key (SAK)	4C973DBC7364621674F8B5B89E5C1551 1FCED9216490FB1C1A2CAA0FFE0407E5
Secure Data	BA 8A E3 1B C5 06 48 6D 68 73 E4 FC E4 60 E7 DC 57 59 1F F0 06 11 F3 1C 38 34 FE 1C 04 AD 80 B6 68 03 AF CF 5B 27 E6 33 3F A6 7C 99 DA 47 C2 F0 CE D6 8D 53 1B D7 41 A9 43 CF F7 A6 71 3B D0
ICV	26 11 CD 7D AA 01 D6 1C 5C 88 6D C1 A8 17 01 07

```
key size = 256 bits
P:     504 bits
A:     160 bits
IV:    96 bits
ICV:   128 bits
K:     4C973DBC7364621674F8B5B89E5C1551
       1FCED9216490FB1C1A2CAA0FFE0407E5
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B3C
       3D3E3F404142434445464748490008
A:     68F2E77696CE7AE8E2CA4EC588E54D00
       2E58495C
IV:    7AE8E2CA4EC500012E58495C
GCM-AES Encryption
H:            9A5E559A96459C21E43C0DFF0FA426F3
Y[0]:         7AE8E2CA4EC500012E58495C00000001
E(K,Y[0]):    316F5EDB0829AC9271A6AFF79F3600BF
Y[1]:         7AE8E2CA4EC500012E58495C00000002
E(K,Y[1]):    B28AEC0BD4145B797D65F3E4FD7AFCC0
C[1]:         BA8AE31BC506486D6873E4FCE460E7DC
Y[2]:         7AE8E2CA4EC500012E58495C00000003
E(K,Y[2]):    4A4700D02733D0381D12D9342D87AB9A
C[2]:         57591FF00611F31C3834FE1C04AD80B6
Y[3]:         7AE8E2CA4EC500012E58495C00000004
E(K,Y[3]):    452D80FF6A15D5070A904BA1E37DF9CC
C[3]:         6803AFCF5B27E6333FA67C99DA47C2F0
Y[4]:         7AE8E2CA4EC500012E58495C00000005
E(K,Y[4]):    F3E8B2135A9502ED0689B0EE383BD81D
C[4]:         CED68D531BD741A943CFF7A6713BD0
X[1]:         1F7477283AA77457BD0C161CB6F179C5
X[2]:         617F112B72DF67BC42218163B73AF025
X[3]:         20A91ADD33433324DBE7822A5BC98013
X[4]:         84D320FCB3B7AF10A66A48BADD00CFA1
X[5]:         52F52D34BC031431185DB9A617FCE98C
X[6]:         57E7CFDDBA0BA07415FD58BCEE906CAC
GHASH(H,A,C): 177E93A6A2287A8E2D2EC236372101B8
C:     BA8AE31BC506486D6873E4FCE460E7DC
       57591FF00611F31C3834FE1C04AD80B6
       6803AFCF5B27E6333FA67C99DA47C2F0
       CED68D531BD741A943CFF7A6713BD0
T:     2611CD7DAA01D61C5C886DC1A8170107
```

C.8.3 GCM-AES-XPN-128 (75-octet frame confidentiality protection)

Table C-47 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN, 96-bit Salt, and 128-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-128 Cipher Suite when used with the frame field data of Table C-43 and Table C-44.

Table C-47—GCM-AES-XPN-128 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	88EE087FD95DA9FBF6725AA9D757B0CD
Secure Data	EA EC C6 AF 65 12 FC 8B 6C 8C 43 BC 55 B1 90 B2 62 6D 07 D3 D2 18 FA F5 DA A7 D8 F8 00 A5 73 31 EB 43 B5 A1 7A 37 E5 B1 D6 0D 27 5C CA F7 AC D7 04 CC 9A CE 2B F8 BC 8B 9B 23 B9 AD F0 2F 87
ICV	34 6B 96 D1 13 6A 75 4D F0 A6 CD E1 26 C1 07 F8

```
key size = 128 bits
P:     504 bits
A:     160 bits
IV:    96 bits
ICV:   128 bits
K:     88EE087FD95DA9FBF6725AA9D757B0CD
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B3C
       3D3E3F404142434445464748490008
A:     68F2E77696CE7AE8E2CA4EC588E54D00
       2E58495C
IV:    9C002902F801C33E323EB331
GCM-AES Encryption
H:            AE19118C3B704FCE42AE0D15D2C15C7A
Y[0]:         9C002902F801C33E323EB33100000001
E(K,Y[0]):    051CB848B04A95168858F67B22FB45CD
Y[1]:         9C002902F801C33E323EB33100000002
E(K,Y[1]):    E2ECC9BF7400EF9F799A54A44CAB8BAE
C[1]:         EAECC6AF6512FC8B6C8C43BC55B190B2
Y[2]:         9C002902F801C33E323EB33100000003
E(K,Y[2]):    7F7318F3F33AD9D1FF81FFD0298F581D
C[2]:         626D07D3D218FAF5DAA7D8F800A57331
Y[3]:         9C002902F801C33E323EB33100000004
E(K,Y[3]):    C66D9A914B05D685E33B1064F3CD97EB
C[3]:         EB43B5A17A37E5B1D60D275CCAF7ACD7
Y[4]:         9C002902F801C33E323EB33100000005
E(K,Y[4]):    39F2A58E6ABAFFCFDE65FEE5B92F8F8E
C[4]:         04CC9ACE2BF8BC8B9B23B9ADF02F87
X[1]:         A9845CAED3E164079E217A8D26A600DA
X[2]:         09410740B1204002F754119A976F31C8
X[3]:         AEEAC0AE90A8F750E3328F7EC27BC7C9
X[4]:         0AC259846D1B384C53E945A6EFFFD3B3
X[5]:         05E38C36366E7137A9CAB89B45CDCE1A
X[6]:         868E036CF44752D418D368EA772C9CE4
GHASH(H,A,C): 31772E99A320E05B78FE3B9A043A4235
C:     EAECC6AF6512FC8B6C8C43BC55B190B2
       626D07D3D218FAF5DAA7D8F800A57331
       EB43B5A17A37E5B1D60D275CCAF7ACD7
       04CC9ACE2BF8BC8B9B23B9ADF02F87
T:     346B96D1136A754DF0A6CDE126C107F8
```

C.8.4 GCM-AES-XPN-256 (75-octet frame confidentiality protection)

Table C-48 specifies arbitrary values for the SSCI, the 32 most significant bits of the 64-bit PN, 96-bit Salt, and 256-bit key (SAK), with the Secure Data and ICV generated by the GCM-AES-XPN-256 Cipher Suite when used with the frame field data of Table C-43 and Table C-44.

Table C-48—GCM-AES-XPN-256 Key, Secure Data, and ICV (example)

Field	Value
SSCI	7A30C118
PN (ms 32 bits)	B0DF459C
Salt	E630E81A48DE86A21C66FA6D
Key (SAK)	4C973DBC7364621674F8B5B89E5C1551 1FCED9216490FB1C1A2CAA0FFE0407E5
Secure Data	B0 FE A3 63 18 B9 B3 64 66 C4 6E 9E 1B DA 1A 26 68 58 19 6E 7E 70 D8 82 AE 70 47 56 68 CD E4 EC 88 3F 6A C2 36 9F 28 4B ED 1F E3 2F 42 09 2F DF F5 86 8A 3C 64 E5 61 51 92 A7 A3 76 0B 34 BC
ICV	85 69 2C D8 15 B6 64 71 1A EF 91 1D F7 8D 7F 46

```
key size = 256 bits        P:     504 bits      A:     160 bits
IV:    96 bits             ICV:   128 bits
K:     4C973DBC7364621674F8B5B89E5C1551
       1FCED9216490FB1C1A2CAA0FFE0407E5
P:     08000F101112131415161718191A1B1C
       1D1E1F202122232425262728292A2B2C
       2D2E2F303132333435363738393A3B3C
       3D3E3F404142434445464748490008
A:     68F2E77696CE7AE8E2CA4EC588E54D00
       2E58495C
IV:    9C002902F801C33E323EB331
GCM-AES Encryption
H:             9A5E559A96459C21E43C0DFF0FA426F3
Y[0]:          9C002902F801C33E323EB33100000001
E(K,Y[0]):     35F6654C6A3A1D45F1D3C3E5C6B4CAC5
Y[1]:          9C002902F801C33E323EB33100000002
E(K,Y[1]):     B8FEAC7309ABA07073D2798602C0013A
C[1]:          B0FEA36318B9B36466C46E9E1BDA1A26
Y[2]:          9C002902F801C33E323EB33100000003
E(K,Y[2]):     7546064E5F52FBA68B56607E41E7CFC0
C[2]:          6858196E7E70D882AE70475668CDE4EC
Y[3]:          9C002902F801C33E323EB33100000004
E(K,Y[3]):     A51145F207AD1B7FD829D4177B3314E3
C[3]:          883F6AC2369F284BED1FE32F42092FDF
Y[4]:          9C002902F801C33E323EB33100000005
E(K,Y[4]):     C8B8B57C25A72215D7E1E43E4234B450
C[4]:          F5868A3C64E5615192A7A3760B34BC
X[1]:          1F7477283AA77457BD0C161CB6F179C5
X[2]:          617F112B72DF67BC42218163B73AF025
X[3]:          0ECB1CA029F4B30D352C800C284B6BAD
X[4]:          3E312F6336A81FEF782782F906EEBC0E
X[5]:          17322A4719E50FA1E2082E54CB12CB49
X[6]:          44D9DB0896C5D819DA6C9B40330B8BAC
GHASH(H,A,C): B09F49947F8C7934EB3C52F83139B583
C:     B0FEA36318B9B36466C46E9E1BDA1A26
       6858196E7E70D882AE70475668CDE4EC
       883F6AC2369F284BED1FE32F42092FDF
       F5868A3C64E5615192A7A3760B34BC
T:     85692CD815B664711AEF911DF78D7F46
```

Annex D

(normative)

PICS proforma for an Ethernet Data Encryption device[27]

D.1 Introduction

The supplier of a protocol implementation which is claimed to conform to this standard's provisions for an Ethernet Data Encryption device (EDE) 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.

D.2 Abbreviations and special symbols

D.2.1 Status symbols

M	mandatory
O	optional
O.n	optional, but support of at least one of the group of options labelled by the same numeral *n* is required
X	prohibited
pred:	conditional-item symbol, including predicate identification: see D.3.4
¬	logical negation, applied to a conditional item's predicate

D.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.

D.3 Instructions for completing the PICS proforma

D.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 D.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 labelled *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.

D.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.

D.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.

D.3.4 Conditional status

D.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 D.3.4.2, and S is a status symbol, M or 0.

If the value of the predicate is True (see D.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.

D.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.

D.4 PICS proforma for IEEE Std 802.1AE EDE

D.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).	

D.4.2 Protocol summary, IEEE Std 802.1AE EDE

Identification of protocol specification	IEEE Std 802.1AE-2018, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security
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 D.3.3: the answer Yes means that the implementation does not conform to IEEE Std 802.1AE-2018.)	No [ ] Yes [ ]

Date of Statement	

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	5.5, 5.6, 15.4, D.7	Yes [ ]
EDECS	Does the implementation conform to the specification for an EDE-CS?	O.1	5.5, 5.7, 15.5, D.7	Yes [ ]
EDECC	Does the implementation conform to the specification for an EDE-CC?	O.1	5.5, 5.8, 15.6, D.7	Yes[ ]
EDESS	Does the implementation conform to the specification for an EDE-SS?	O.1	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 [ ]
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 [ ]

D.6 EDE-M Configuration

Item	Feature	Status	References	Support
MC	Can the EDE-M be configured to secure connectivity within a customer or provider network? When so configured:	EDEM:M	5.6(c), 15.2	Yes [ ]
MCadd	Does the PAE supporting the black-side port SecY use the Nearest non-TPMR Bridge group address to transmit and receive EAPOL frames?		15.2	Yes [ ]
	Does the EDE's relay entity:			
MCftr	Filter frames whose destination MAC address is a TPMR component Reserved Address or the Nearest non-TPMR Bridge group address?		15.2	Yes [ ]
MCfwd	Forward frames whose destination MAC address is a Reserved Address that is not one of the above?		15.2	Yes [ ]
MB	Can the EDE-M be configured to secure connectivity across a provider network? When so configured:	EDEM:M	5.6(d), 15.4, 15.5	Yes [ ]
MBadd	Does the PAE supporting the black-side port SecY use the Nearest Customer Bridge group address to transmit and receive EAPOL frames?		15.4	Yes [ ]
	Does the EDE's relay entity:			
MBftr	Filter frames whose destination MAC address is a C-VLAN component Reserved Address with the exception of the Nearest Customer Bridge group address?		15.4	Yes [ ]
MBfwd	Forward frames whose destination MAC address is the Nearest Customer Bridge group address?		15.4	Yes [ ]
MM	Can the EDE-M be configured to secure connectivity across a provider network to an EDE-CC? When so configured:	EDEM:O	5.6(e), 15.6	Yes [ ]
MMadd	Does the PAE supporting the black-side port SecY use the EDE-CC PAE group address to transmit and receive EAPOL frames?		15.6	Yes [ ]
	Does the EDE's relay entity:			
MMftr	Filter frames whose destination MAC address is a C-VLAN component Reserved Address or the EDE-CC PAE group address?		15.6	Yes [ ]
Mpri	Can the EDE-M be configured to recover priority from C-VLAN tagged frames transmitted by the black-side port? When so configured:	EDEM:O	5.6(f), 15.4	Yes [ ]
Mprit	Can the black-side port be configured to priority tag or not priority tag transmitted frames?	MPri:M	5.6(f), 15.4	Yes [ ]

D.7 EDE-CS Configuration

Item	Feature	Status	References	Support
CS	Does the EDE-CS comprise a Provider Edge Bridge with a single C-VLAN component?	EDECS:M	5.7(a), 15.5	Yes [ ]
CSadd	Can the PAE supporting each PEP's SecY use the Nearest Customer Bridge group address to transmit and receive EAPOL frames?		5.7(c), 15.5	Yes [ ]

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?		5.8(e), 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?		5.8(e), 15.6	Yes [ ]

D.9 EDE-SS Configuration

Item	Feature	Status	References	Support
SS	Does the EDE-SS comprise two S-VLAN components internally connected as specified in 15.7?	EDESS:M	5.9(a), 15.7	Yes [ ]
SSadd	Can the PAE supporting each PEP's SecY use the EDE-SS PAE group address to transmit and receive EAPOL frames?		5.9(c), 15.7	Yes [ ]
SSftr	Does the EDE's edge component relay entity filter frames whose destination MAC address is an S-VLAN component Reserved Address or the EDE-SS PAE group address?		5.9(d), 15.7	Yes [ ]
SSrlyu	Are frames received untagged on the red-side port and relayed through the black-side port transmitted untagged?		5.8(e), 15.7	Yes [ ]
SSrlyt	Are frames received S-tagged on the red-side port and relayed through the black-side port transmitted S-tagged with the received S-VID?		5.8(e), 15.7	Yes [ ]

Annex E

(informative)

MKA operation for multiple transmit SCs

As specified in this standard (7.1.2) when a SecY uses multiple transmit SCs, each SCI is represented by a separate MKA participant that sends and receives MKPDUs to and from all the other participants just as if it were representing a separate SecY in the same group CA. This annex provides tutorial information that might prove useful when implementing a KaY that supports multiple participants. Conformance to this standard remains strictly in terms of externally observable behavior and does not depend on any implementation suggestion in this Annex. Refer to IEEE Std 802.1X for the specification of the normative externally observable behavior of an MKA participant and for the definition of additional terms and acronyms used in this Annex.

All potential participants in the same CA necessarily possess the same secure Connectivity Association Key (CAK) identified by the same secure Connectivity Association Key Name (CKN). Whether this information is derived from an authentication protocol exchange conducted by the PAE or is pre-shared keying material it applies equally to a KaY's multiple traffic class SCI MKA participants just as it does for a participant representing a single SCI, and there are no additional authentication protocol requirements. Similarly there are no additional requirements for group CAK distribution. Although the use of multiple transmit SCIs by a SecY participating in a CA is in many respects similar to that of a group with multiple SecYs, each with a single transmit SC, there is no need to use or distribute an explicit group CAK if only two KaYs participate in the CA; those KaYs simply make the same keying material available to all their MKA participants.

If an implementation of a KaY operates by instantiating separate MKA participants, the implementer needs to be aware that each needs to receive MKPDUs transmitted by other KaYs and by each other. It is also possible to mimic two or more externally visible participants by extending the internal operation of a single participant's implementation, as in the following description. This has advantages including the following:

— Not complicating transmission and reception and the associated scheduling.
— Preserving the existing operation of the PAE's Logon Process[28] and its interaction with the PAE's CP state machine and the KaY.
— Using a single instance of the CP state machine, thus avoiding any need to make decisions about distributing the responsibility for managing receive SAs, avoiding introducing complications into SAK and CAK rollover resulting from dependencies between the participants.[29]

Each participant uses a separate MI and its own MN. The conditions under which either chooses a new MI remain unchanged, though each new MI should also be checked (and reselected if necessary) against those of its co-resident participants (representing the SCIs for other traffic classes).

The KaY maintains just one Live Peers List, and one Potential Peers List, using received MKPDUs to add to each in the usual way. When a participant transmits an MKPDU, its co-resident participants (representing the SCIs for other traffic classes) are added to the transmitted Live Peer List.

[28] See 12.5 and Clause 12 of IEEE Std 802.1X-2010.

[29] If the actors are implemented and scheduled separately and one of them is a potential key server, the best results will generally be obtained by scheduling that actor first on MKPDU reception.

Only one participant, that associated with the default traffic class SC, should advertise itself as a potential Key Server, with the others advertising a Key Server Priority of 0xFF.[30] Similarly only the default traffic class participant should distribute keys in MKPDUs (if selected).

In the CP state machine, RECEIVE state createSAs(lki) refers to the SAs used to receive from other members of the CA and all the transmit SAs used by the co-resident participants. Naturally there is no need to instantiate receive SAs for the latter. The 'electedSelf ...' transition from CP:RECEIVING is taken if satisfied by default traffic class participant.[31] In CP:TRANSMIT all the traffic class group transmit SAs are enabled.

The retireWhen timer in CP:TRANSMITTING takes care of the fact that a preempted frame may complete its arrival after a preempting frame has been received on a new SA from its transmitting port. This is legitimate when strict replay protection is not enforced.

[30] See 9.5 of IEEE Std 802.1X-2010. The default traffic class SC should be retained even if SCIs for other traffic class groups are added to or deleted from a running system, so this decision minimizes the potential disruption as well as avoiding the need for a management variable for the other SCs. IEEE Std 802.1Xbx-2014 clarifies a number of other issues related to Key Server selection that are equally applicable to the single and multiple participant cases.

[31] None of the other co-resident participants can be elected Key Server in preference.

Annex F

(informative)

EDE Interoperability and PAE addresses

This annex discusses interoperability between EDEs of the same and different types, and between EDEs and MACsec-capable bridges, providing some background and analysis that goes beyond the normative requirements of this standard.

Each of the types of EDE is naturally designed to interoperate with one or more EDEs of the same type, given suitable connectivity. For example, an EDE-M can interoperate with another EDE-M, or an EDE-CC with one or more EDE-CCs attached to the same PBN. Other interoperability scenarios are also possible, an EDE-M can interoperate with a MACsec-capable Customer Bridge, for example. In all scenarios, the secure Connectivity Associations (CAs) that they create need to have an appropriate scope. If, for example, two EDEs on either side of a Customer Bridge were to create a CA that passed through that bridge, it would then be unable to peer with its neighboring Customer Bridge. Configuration frames, such as RSTP BPDUs, originated by the bridge would be discarded by the EDEs, while frames destined for it would have an additional SecTAG and might be encrypted, rendering them indecipherable.

IEEE Std 802.1Q specifies Reserved Addresses, group MAC destination addresses used by MAC sublayer configuration and discovery protocols, that are selectively filtered by bridge components of various types. This filtering ensures that each protocol is confined to an appropriate scope. For example, when the Link Layer Discovery Protocol (LLDP, IEEE Std 802.1AB) is used to manage Power over Ethernet, it uses the Individual LAN (01-80-C2-00-00-0E) group address which is filtered by all bridges, including TPMRs. The LLDP frames thus reflect the power available from, and power drawn by, the devices attached to a single Ethernet LAN and do not erroneously include devices attached to other, bridged, LANs that do not affect the power budget. LLDP can also be used with different addresses. For example, the Provider Bridge Group Address (01-80-C2-00-00-08) is not filtered by TPMRs but is filtered by Provider Bridges, so it can be used by a Provider Bridge to discover its neighboring Provider Bridges as part of constructing a map of the provider network topology, without reaching the erroneous conclusion that distant bridges are locally connected. Similarly, a Provider Bridge Port that uses MACsec to secure connectivity can use the Nearest non-TPMR Bridge group address as the destination address of the EAPOL frames that it sends and receives to support peer authentication and key agreement—thus ensuring that the scope of any resulting CA does not extend beyond bridges that need to interpret the protected frames.

Note that it is the destination MAC address that determines the extent of the propagation (through various types of bridge) of protocol frames, and not the protocol type[32] or (when the frame is being transmitted by a bridge port) the type of bridge that originally transmits the frame. Separate instances of the same protocol can operate independently of each other, each within its own scope, without having to allocate additional protocol types for each possible application. Each bridge can enforce the desired scoping by correctly filtering a small number of group addresses without having to know every one of a larger (and increasing number) of protocol types, and without having to access the protocol type in (potentially encrypted) frames. A Customer Bridge Port can, for example, use LLDP frames with the Individual LAN group address to manage power (as described in the previous paragraph) and LLDP frames with the Nearest Customer Bridge group address to discover its peers. Similarly, the scope of one MACsec protected CA (between two EDE-Ms that form part of the connectivity between two neighboring Customer Bridges, for example) might be nested within the scope of another (between those bridges), with the organizations responsible for securing the two CAs being quite separate, not depending on or even necessarily being aware of each other.[33]

[32] Most commonly the EtherType of the frame.

[33] Clause 11 provides further examples of nested scopes (11.7).

The EAPOL destination address used by an EDE's Provider Edge Port PAE is naturally chosen to match the scope of the desired CA. The edge component of EDE-CCs, EDE-CSs, and EDE-SSs, and the MAC Bridge component of EDE-Ms, also enforce that scope by filtering their PAE addresses—precluding the choice of an address that supports an enclosing, more extensive, scope.

— The PAE of an EDE-M that is configured to secure simple LAN connectivity uses the Nearest non-TPMR group address (15.2).

— The PAE of an EDE-M that is configured to secure connectivity across a PBN uses the Nearest Customer Bridge group address (15.4). Frames with this destination address are forwarded by PBN port-based interfaces, S-tagged interfaces, and C-tagged interfaces that provide access to a single service instances, and the scope of the CA created can encompass the provider network service instance as the PBN has no need to interpret the protected frame. The PAE will be unable to establish secure connectivity if the EDE-M is attached to a PBN C-tagged interface that supports multiple service instances, as that interface will discard the PAE's EAPOL frames. The EDE will also not secure connectivity if the EAPOL frames received from the interface are tagged—it transmits EAPOL frames untagged or priority tagged.

  Frames destined to the Nearest Customer Bridge group address are forwarded by the EDE-M provided that the requirements for connectivity through the associated SecY's Controlled Port are met—its operation is transparent to that of Customer Bridges using that address for other protocols. In principle, two nested MACsec secured CAs could be created, both supported by EAPOL frames using the Nearest Customer Bridge address: an inner CA between a pair of EDE-Ms, and an outer CA between MACsec capable Customer Bridges attached to those EDEs. EAPOL frames associated with the outer CA would not be forwarded by the inner CA until the latter had established secure connectivity, when their initial octets would be a SecTAG so they would not be recognized by the EDE PAEs as EAPOL frames. If one of the EDE-Ms (M1, say) has been configured to allow insecure connectivity, the other EDE-M (M2) might establish a CA with M1's attached Customer Bridge (C1) if common PSKs have been (inadvisably) configured or authorization controls omitted from EAP authentication. This might or might not be viewed as a problem, though deliberate creation of this scenario is to be avoided. If the inner CA is subsequently established, the situation will eventually resolve into the two nested CAs.

— An EDE-CS's PAEs use an address that is forwarded by the S-VLAN components in Provider Bridges and is filtered (preferably) by C-VLAN components in Customer Bridges and Provider Edge Bridges, and by VLAN-unaware MAC Bridges. The Nearest Customer Bridge address can be used, allowing interoperability in a hub-and-spoke configuration with EDE-Ms or MACsec-capable Customer Bridges attached to PBN port-based interfaces.

— An EDE-CC's PAE's address has to be forwarded by PEBs that support multiple provider service instances, ruling out use of any of the Reserved Addresses specified by IEEE Std 802.1Q. This standard specifies the use of the EDE-CC PEP Address (01-80-C2-00-00-1F) to facilitate interoperability. It also recommends filtering of that address by bridges in the customer network, to reduce the chance of creating an inappropriately scoped CA that extends from an EDE-CC, through a PBN, across part of a customer network, through the same or another PBN, and back to another EDE-CC.

— An EDE-SS's PAEs have the same component filtering requirements as an EDE-CS with two differences.

  First, the address used should not be the same as the EDE-CS PAE Address. An EDE-SS PAE and EDE-CS PAE can communicate, the EAPOL frames that each transmits and receive are simply S-tagged. However, the purpose of the EDE-CS and that of the EDE-SS differ: the former accepts and delivers C-tagged frames to and from its Customer Edge Port; the latter accepts and delivers S-tagged frames. Just as there is little point in using either EDE to support a single untagged VLAN, so there is little point to both participating in the same CA, while preventing that participation will guard against the unintended S-tagged/C-tagged swap.

Second, a Reserved address that is not the Nearest Customer Bridge should be used, as the PAE address should be filtered by EDE-SSs and by all C-VLAN components, making it impossible to reach the EDE-SS PAE with an untagged EAPOL frame from a bridge or end station attached to a C-tagged PBN interface, even if frames from the external bridge or end station are not C-tagged when mapped to a provider service instance.

Table F-1 summarizes the scenarios described in this annex and others detailed in this standard, with recommended PAE addresses.

Table F-1—Interoperability scenarios and PAE Addresses

System(s)	Peer System(s)	Connectivity	PAE Address
MAC or Customer Bridge Port — EDE-M	MAC or Customer Bridge Port — EDE-M	Port i/f–PBN–Port i/f	Nearest Customer Bridge[a]
		LAN	Nearest non-TPMR Bridge[b]
	PEB Customer Edge Port — Provider Bridge Port		
	EDE-CS	Port i/f–PBN-S-tagged i/f[c]	Nearest Customer Bridge
	EDE-CC	Port i/f–PBN—C-tagged[c]	EDE-CC PEP Address[d]
EDE-CC	EDE-CC	C-tagged i/f–PBN–C-tagged i/f	EDE-CC PEP Address
EDE-CC	EDE-CS	C-tagged i/f-PBN-S-tagged i/f	EDE-CC PEP Address[e]
EDE-CS	EDE-CS	C-tagged i/f–PBN–S-tagged i/f	Nearest Customer Bridge
EDE-SS	EDE-SS	PBN or PBBN S-tagged i/f–PBN–PBN or PBBN S-tagged i/f — PBBN transparent i/f–PBBN transparent i/f	EDE-SS PEP Address[f]
Provider Bridge Port	Provider Bridge Port	LAN	Nearest non-TPMR

[a]01-80-C2-00-00-00

[b]01-80-C2-00-00-03

[c]Hub-and-spoke scenarios with an EDE hub, with the PBN configured to deliver a single service instance with no outer VLAN tag at the port based interface.

[d]01-80-C2-00-00-1F

[e]This configuration requires that the PBN service instances not deliver the service selecting C-tag used by the EDE-CC at the S-tagged interfaces used by the EDE-CS.

[f]01-80-C2-00-00-0B

Annex G

(informative)

Management and MIB revisions

Prior to IEEE Std 802.1AEcg-2017,[34] the SecY MIB module in this standard had not been revised since the initial version published as part of IEEE Std 802.1AE-2006 and identified as MIB-2006 in the following discussion. The revised MIB-2016 includes an updated compliance statement. Implementations that do not require MIB-2016's additional capabilities can continue to conform to the standard's (optional) requirement for remote management by continuing to support MIB-2006—subject to future revision as new implementations are expected to find MIB-2016 easier to support, whatever their functionality. Table 13-5, Table 13-6, Table 13-7, and Table 13-8 list both current and deprecated MIB objects. Where possible, MIB-2016 uses tables and objects already defined by MIB-2006. However, when the need to index for multiple transmit SCs required the creation of a new table, new objects were added only to that table and not to its single transmit SC predecessor.

IEEE Std 802.1AEcg-2017 added support for multiple, per traffic class, transmit SCs so that frames misordered within a traffic class can be discarded and strict replay protection provided even when the MAC service reorders frames of different classes or priorities. Such reordering services include Provider Bridged Networks and the use of IEEE 802.3 frame preemption on individual LANs. Management changes were necessary (for both control and reporting) to take full advantage of this new capability. Distinct traffic classes use distinct SCIs (secyIfSCI added to secyIfTable, as the SCI for the default traffic class) and distinct transmit SCs and SAs (secyTSCTable and secyTSATable derived from secyTxSCTable and secyTxSATable respectively, with an additional index of secyTSCI for the traffic class for each table entry). Control over the mapping of frames (with a given user priority) to traffic class SCs, and over the mapping of user Priority Code Point (PCP) (comprising priority bits and the discard eligible bit) to access PCP values, is provided a new Traffic Class Table (secyIfTCTable) and an Access Priority Table (secyIfAPTable).

The use of multiple transmit SCs multiplies the number of both receive and transmit per SC and SA counters. MIB-2006 was developed prior to the standardization of the MACsec Key Agreement protocol (MKA) in IEEE Std 802.1X-2010, and the desired counter capability extended to diagnosing manual or ad-hoc key distribution failures. MKA provides better management visibility without requiring counting at packet data rates, and MIB-2016 includes a text string Key Identifier (matching that originally called for in Clause 10 of IEEE Std 802.1AE-2006) that links MKA operation to the creation and use of SecY SAs. Any alternate key agreement or key distribution protocol should provide equivalent functionality, so a number of counters have been deprecated and reorganized. This annex describes the rationale for the changes (G.1). It also describes how the counts collected by an implementation supporting the 2006 MIB can be used to support the 2016 MIB.

Finally, since these changes involve a significant MIB revision and a new compliance statement, maintenance changes that by themselves would not have justified a MIB update have been included, rather than delay these to a separate later revision with a possible further change to compliance. The general industry move to YANG means that these might be the last changes to this MIB. The maintenance changes included making the full extended PN and the SSCI visible for XPN-capable Cipher Suites, and rectifying discrepancies between the MIB and normative text of the rest of the standard.

[34] This revision, IEEE Std 802.1AE-2018, incorporates the text of IEEE Std 802.1AE-2006 and amendments IEEE Std 802.1AEbn-2011, IEEE Std 802.1AEbw-2013, and IEEE Std 802.1AEcg-2017.

G.1 Counter changes

IEEE Std 802.1AEcg-2017 made changes to the specification of management counters with a primary goal of enabling cost reduction in real implementations, both by pointing out where implementation dependent cost reduction opportunities exist, and by reducing the number of formal counters.

Whether a verification counter can or cannot be incremented by a received frame depends on the setting of the management control validateFrames. While distinctly named counters are provided for clarity and can all accumulate counts over the long term (as validateFrames can be changed while an SC persists), the real time counting requirement can be reduced. NOTE 1 to Figure 10-4 is intended to make this clear.

Similarly real-time generation counts can be derived from the value of nextPN. See NOTE 1 to Figure 10-3. MIB-2016 does not explicitly include per SA OutPktsEncrypted and OutPktsProtected counts, removing the need for MIB-2006's secyTxSAStatsTable. If necessary this information can be retrieved from the secyTSATable.

MIB-2016's addition of the secyTSAKeyIdentifier and secyRxSAKeyIdentifier provide an explicit indication of whether or not two system's SAKs match, removing the need to use per SA receive counters to analyze synchronization issues. SA status (including SAK use) is also conveyed explicitly by MKA, so that MIB-2016 no longer requires per-SA receive counters (other than nextPN and lowestPN, which are required for basic operation). The use of XPN-capable Cipher Suites also makes it easier to identify SA and SAK rollover related issues, since SAK changes are now required only to meet key lifetime policy requirements (typically one week or longer) even on the fastest links, rather than being forced by PN exhaustion. MIB-2006's unknown SCI and per SC and SA unused SA counts have been combined into a single per SecY MIB-2016 count, as have the no SCI and per SC and SA not using SA counts. This combination also addresses the issue of attributing a precise error to a packet whose integrity cannot be validated.

MIB-2006 included per transmit SA and per receive SC octet counters (distinguishing protected/validated/encrypted/decrypted octets) not called for in Clause 10. These were deprecated in MIB-2016. Subclauses 10.7.10, 10.7.19, and Figure 10-5 describe using counts to investigate or validate cryptographic performance under the assumption that protection resources are dedicated to the SecY as a whole, not to individual SCs or SAs. MIB-2016 includes these per SecY counts, which are a simple sum of the MIB-2006 counts.

Packet counters required for MIB-2016 current compliance and not in MIB-2006 can be derived from those for the latter as follows:

— secyStatsRxNoSAPkts (per SecY) is the sum of
  secyStatsRxUnknownSCIPkts (per SecY),
  secyRxSCStatsUnusedSAPkts (per SC), and
  secyRxSAStatsUnusedSAPkts (per SA)
— secyStatsErrNoSAPkts is the sum of
  secyStatsRxNoSCIPkts (per SecY),
  secyRxSCStatsNotUsingSAPkts (per SC), and
  secyRxSAStatsNotUsingSAPkts (per SA)

Two of the IETF RFC 2863 Interfaces Group MIB counts do not have to be counted separately, but can be derived from MIB-2016 counts as follows:

— ifInErrors ("the number of inbound packets that contained errors preventing them from being deliverable") is the sum of
secyStatsRxNoTagPkts (per SecY),
secyStatsRxBadTagPkts (per SecY),
secyStatsRxNoSAErrorPkts (per SecY),
secyRxSCStatsLatePkts (per SC), and
secyRxSCStatsNotValidPkts (per SC)

— ifInDiscards ("the number of inbound packets which were chosen to be discarded even though no errors had been detected") is the same (provided that no other SecY resource limitation prevents frame delivery) as
secyStatsRxOverrunPkts (per Secy)

G.2 Available Cipher Suites

MIB-2006 provided a list of Cipher Suites for a system, with limited control over the use of each Cipher Suite, and a way for network management to add to the list. It did not provide a way of controlling or limiting Cipher Suite use per Controlled Port, nor did it support systems with different Cipher Suite capabilities on different ports (which might occur when cards for a chassis based system are procured over time). MIB-2016 added a per Controlled Port list that provides control over the use of each implemented Cipher Suite (secyIfCipherEnableUse, secyIfCipherRqConfidentiality) together with an index for the Cipher Suite entry in the secyCipherSuiteTable. The latter contains information that does not change per Controlled Port and was carried over from MIB-2006, though the use of the RowStatus object was deprecated as it is the system or subsystem that determines whether a Cipher Suite is implemented and potentially available for use, and not a network manager.

Consensus

WE BUILD IT.

Connect with us on:

Facebook: https://www.facebook.com/ieeesa

Twitter: @ieeesa

LinkedIn: http://www.linkedin.com/groups/IEEESA-Official-IEEE-Standards-Association-1791118

IEEE-SA Standards Insight blog: http://standardsinsight.com

YouTube: IEEE-SA Channel

IEEE
standards.ieee.org
Phone: +1 732 981 0060 Fax: +1 732 562 1571