MACsec Privacy
August 31, 2019
()

Individual contribution—

Media Access Control (MAC) Security

Amendment:
MAC Privacy protection

Mick Seaman

This document is an individual contribution to assist discussion of potential new work on privacy protecting enhancements to be used in conjunction with the existing MAC Security (MACsec) protocol. The initial proposal from Don Fedyk aims at enhancing privacy when both Integrity and Confidentiality protection are being provided by MACsec, by making it harder (potentially impossible) for an adversary to use observed MAC Addresses, frame lengths, frame transmission timing, and bandwidth as part of fingerprinting network users and their activities.

The principle purpose of this document is to examine how this work could fit as an amendment to IEEE Standard 802.1AE, with the important effects of teasing out the complete scope of the work to be undertaken (if a project is approved) and of reducing the risk of prematurely concluding that the work is technically complete prior to its integration with the standard and ensuring that conformance claim, interoperability (including use cases and addressing requirements), and management are properly addressed.

This document is an individual contribution, not a draft standard, even though it necessarily (to meet its goals) mimics the formalism of an amendment. The related work is not yet, and may not become, an approved project. It reflects my own opinions and not group discussion or agreement, the opinions of others will inevitably differ.

Abstract: The MAC Privacy-protection protocol specified in this amendment can be used in conjunction with the MAC security protocol (MACsec) to hide the ultimate or end user source and destination MAC addresses, and to protect against traffic analysis based on the sizes and timing of data frames, which an observer might otherwise correlate with user identities and communication purpose, application, and content. A YANG model supports management of MAC security and the enhanced privacy capabilities. Privacy considerations for bridged networks are reviewed.

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

Contents

Figures

Tables

Individual contribution—

Media Access Control (MAC) Security

Amendment:
MAC Privacy protection

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

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

The editing instructions are shown in ***bold italics***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Deletions and insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this note will not be carried over into future editions because the changes will be incorporated into the base standard.[1]

<<Editor's notes, intended to assist review and solicit comment, are set in angle brackets and use this font. All editor's notes are temporary, and will be removed before Sponsor Ballot (at the latest). Editor's notes that appear to be simple text for inclusion in the amendment paraphrase, state the purpose of text that needs to be written, or are a very early draft of that text. Ideas for what the text should actually say are needed.>>

<<All proposed text changes (change, delete, or insert) in this contribution should be considered a mere suggestion and a request/primpt for comment (not limited to changes to that text).>>

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

1. Overview

<I don't believe we need any changes to 1.1 Introduction. Please review and check.>

1.2 Scope

Change 1.2 as follows:

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

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

To this end it

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

b) Specifies the requirements for MAC Security 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 MAC Security 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 a YANG configuration and operational state model for SecY management.

n) Specifies the Management Information Base (MIB) module for managing the operation of MAC Security in TCP/IP networks.

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

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

q) Models support of a privacy-enhanced secure MAC Service in terms of the operation of Privacy-protecting Entities (PrYs) that allow MAC Security to hide the ultimate or end user source and destination MAC addresses of encapsulated data frames and reduce any correlation of their sizes and transmission timing with user identities and communication purposes, applications, or content.

r) Specifies the format of the Privacy-protecting Protocol Data Units (PPDUs) used by PrYs.

s) 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.

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

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

v) Specifies how a PrY collocated with a SecY can use an associated PAE to discover peer PrYs.

w) Specifies administrative configuration of the relationships between peer PrYs.

x) Identifies the managed objects and defines the management operations for a PrY.

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

<<Proposed new list items p) through y) deliberately follow SecY related items c) and f) through o). The objective is to make sure that we understand the full scope of what needs to be specified to make Privacy-enhancement/Privacy-preservation [need to decide between these two] deployment practical. I have rearranged the list a little [moving item v), because in this case it would be easier on the reader to be presented with u) first]. I have also separated items v) and w) beacuse it would appear (given MKA capabilities) that the former is a particularly easy case to deal with, while the use case for placing a PrY in a separate system from that with the SecY [peer SecYs in EDEs, while the PrY encapsulation is elesewhere] seems [to me] to be a less likely candidate for automated configuration.>>

<<It is not necessarily the case that individual list items map to separate top-level clauses, rather they represent topics that need to be covered in some clear organization.>>

<<The new item for a YANG model for basic MACsec is an inevitable consequence of the fact that there is no point in specifying a MIB for privacy enhancement operation. Given that management will have to be defined for the latter (a necessary part of an 802 project) and will be YANG, that necessitates definition of a MACsec YANG model.>>

2. Normative references

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

<<While it is possible that this amendment will not require changes to the normative references list, in any PAR "5.5 Need for the Project" project scope should include "It will also address errors and omissions in existing functionality" and these include updates to references.>>

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

IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.[2,3]

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.[4]

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++.[5]

NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.[6]

[2] IEEE publications are available from The Institute of Electrical and Electronics Engineers (https://www.standards.ieee.org).

[3] The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

[4] IETF RFCs are available from the Internet Engineering Task Force (https://www.ietf.org/rfc.html).

[5] 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/).

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

3. Definitions

Change the following definitions in Clause 3 as shown:

<<No definitions that need changing have been identified at present. Updates were included in the IEEE Std 802.1AE-2018 revision. If there are no changes, remove the preceding editing instruction when this editor's note is remove.>>

Insert the following term and definition in Clause 3 in alphabetical order:

YANG: A data modeling language, published as IETF RFC 7950.

4. Abbreviations and acronyms

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

<<Note: The following abbreviations and acronyms (and others) are already defined in Clause 4:
DA, EDE and EDEs of various types, ES, MKA, MKPDU, PAE, PDU, SA.
Check 802.1AE-2018 before adding further entries.
>>

MPPDU	MAC Privacy-protecting Protocol Data Unit
P-TAG	Privacy TAG
PrY	Privacy-protecting Entity

5. Conformance

Change the introductory text of Clause 5 as follows:

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

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

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

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

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

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

5.1 Requirements terminology

<<It is not intended that this amendment make any changes to Requirements terminology, however this clause (5.1) has been retained in this draft because deviation from the need to adhere to precise conformance terminology has historically been a problem with amendments in general.>>

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***.

Change 5.2 as follows:

5.2 Protocol Implementation Conformance Statements (PICS)

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

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

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

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

<<No changes to 5.3 MAC Security Entity requirements, 5.4 MAC Security Entity options, and 5.5-5.9 EDE requirements and options, are anticipated.>>

Insert the following text (clauses 5.10 and 5.11) after clause 5.9:

5.10 Privacy-protecting Entity requirements

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

a) Support the Client and 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 PDUs 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 MACsec PDUs 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 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.

5.11 Privacy-protecting 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, 14.4.1.

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

17. MAC Privacy protection

This clause provides an overview of MAC Privacy protection. It provides the context necessary to understand the detailed operation of the MAC Privacy-protection protocol (Clause X) and individual MAC Privacy-protecting Entities (PrYs, Clause), and describes the following:

a) Why privacy exposure exists even when MAC Service user data frames are integrity and confidentiality protected.

b) How MAC Privacy protection removes or reduces that exposure.

c) The potential impact of privacy protection on Quality of Service parameters, and the management controls provided to balance the protection provide with the effect on those parameters.

d) Privacy protection deployment, interoperability with existing systems, network configuration, and use case scenarios.

17.1 Privacy protection overview

MACsec secures communication while minimizing its impact on the MAC Service's Quality of Service (QoS) parameters (6.10). Individual frames are cryptographically protected and transmitted with minimal delay, with the addition of only those octets required to support cryptographic integrity and confidentiality protection. Each frame's source and destination MAC Addresses remain unmodified. This deliberately limited impact on the transmission and reception of frames can allow a potentially adversarial observer to correlate those addresses and the pattern of frame sizes, transmission timing, and transmission frequency with the identities of communicating users, the reason they are communicating, and even (in some cases) the content of confidentiality protected communication.[7]

When protecting privacy is paramount, QoS and simplicity of network configuration can be less important. MAC Privacy-protecting Entities (PrYs) encapsulate user data frames within MAC Privacy-protecting Data Units (MPPDUs), provide control over MPPDU transmission timing, and allow MPPDUS to be padded to fixed sizes (17.3). The MAC Source Address of each MPPDU identifies its encapsulating PrY, its MAC Destination Address can be a unicast or multicast address associated with its decapsulating PrY(s). When MPPDUs are confidentiality and integrity protected by MACsec, the source and destination addresses and sizes of the encapsulated user data frames are hidden from an observer who lacks the protecting secret key.

Figure 17-1 shows the addition of both a PrY and a SecY to each of three interface stacks, each providing the privacy protected secure MAC Service to a Bridge Port or end station.

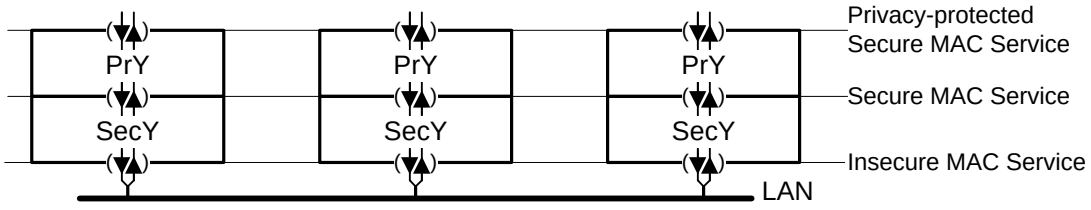


Figure 17-1—Privacy-protected communication between three stations

[7] <<Replace this footnote with a Bibliography reference, ideally to papers (else to newspaper articles - FT?) describing how frame sizes in financial applications could allow observers to deduce approximate balances, whether the account was overdrawn, and whether money was being added to or removrd from the account. Also provide a forward reference to 17.2 Correlation and finger-printing.>>

Figure 17-2 shows an MPPDU that has been protected by MACsec.

<<Figure to show an MACSec-protected MPPDU, using 802.1AE Figure 6-2 (included here) as a starting point. Show both pre- and post-user frame padding, but stick to the simple case of a single encapsulated frame here, the more complex cases can be shown in Clause 18. I have already revised the NOTE that follows the Figure.>>

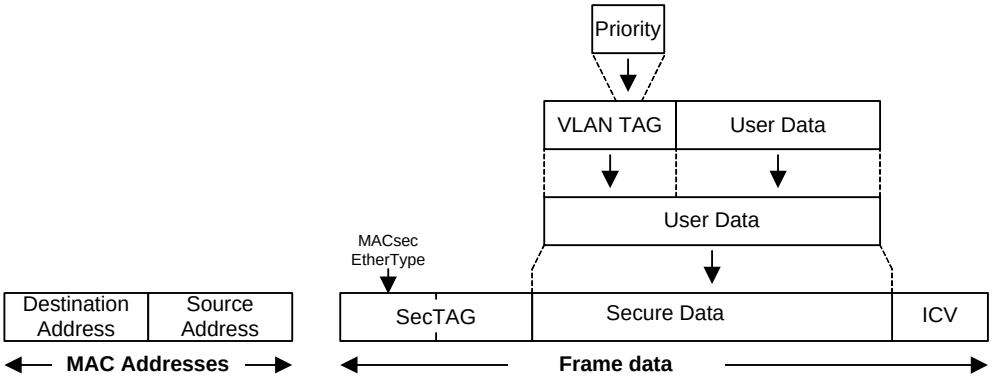


Figure 17-2—A privacy-protected frame

NOTE 2—The MPDUs Destination Address and Source Address are shown as separate from the accompanying data in Figure 17-2, as they are separate parameters of each ISS service request. The supporting service encodes these parameters into a frame and could add octets between those addresses and the MSDU. Strictly this standard specifies parameters of service primitives, not frames. However, it is often convenient to talk of these parameters as a frame.

A single MPPDU can convey more than one user data frame, reducing bandwidth loss when using of large padded MPPDUs sizes. It can also convey no user data at all, but only padding octets, to prevent an observer from determining the level of user activity. A user data frame may be split between two successive MPPDUs, utilizing the remaining octets of an MPPDU that would otherwise be padded (see Clause X).

<<I have included a user data frame fragmentation option (necessitating reassembly by a receiving PrY) since it has been mentioned, but have considerable doubts as to its merits and the likelihood of its deployment as part of MAC Privacy-protection (as opposed to deployment as part of IETF Traffic Flow Security). The potential for frame size fingerprinting might be significantly reduced by padding to a nearest frame size boundary, e.g. to multiples of 68 octets, or even 512 octets, prior to MACsec protection.>>

Two communicating PrYs, encapsulating and decapsulating provide control over transmission timing, and allow frames to be padded to fixed sizes. User data frames are encapsulated within the MAC Privacy-protecting Data Units (MPPDUs) transmitted between PrYs. More than one user data frame can be transmitted in a single MPPDU, reducing bandwidth loss from the use of large padded MPPDUs sizes.

and patternwho does not possess the secure association keys (SAKs) to use the information on frames are transmitted and received, exposes

17.2 Correlation and finger-printing

<<Description of what we are trying to protect against. Use concepts from P802E, but do not rely on the reader reading that document. Similarly can refer to Annex but do not assume reader has read it first, or will read it before the remainder of this clause. Include sufficient examples to point out the need for the address hiding, time transmission, packing, and padding provided. Don't forget to mention the case where address hiding is not the main point - as the MAC addresses of the frames to be protected are those of routers providing a service and may well be constant, not associated with individual activities or activity patterns. However point out that no protection variant is offered that does not encapsulate the addresses - don't believe the bandwidth saving that would represent is worth the complexity (??).>>

17.3 Privacy-protection and Quality of Service

<<The degree of privacy protection provided can be balanced with its impact on Quality of Service. Describe the important use cases where that impact is insignificant - transmission over long distances - and where the risk of adversarial observers is high - public networks, networks with accessible components, networks administered by others..>>

<<Describe the impact on Quality of Service of packet padding, packing, delaying transmission to keep a constant transmission

<<Describe what we are going to do with priority. Valid use cases for packing packets of multiple priorities into a single MPPDU. Are these cases (or at least the important ones) already handled by EDEs, does that have consequences for when and how we introduce EDEs in this discussion. It does have important consequences for the complexity of PrYs, particularly with the modelling (and reality) of the timing of delivering decapsulated frames. To what extent do we acknowledge the fact that delivering multiple frames from a decapsulated frame is a process that might take time. It cannot begin until the whole of the MPPDU has been received, which means that the decapsulated frames are buffered if only for that reception time.>>

<<MSDU size issues. Basic approach, don't worry about these unduly. Not been found to be an obstacle for other tagging schemes. Other quality of service items, go over 6.n clauses again hunting for things that have been missed.>>

17.4 Interoperability and deployment

<<Privacy protection supports point-to-point, multipoint, and point-to-multipoint transmission between perr PrYs.>>

<<Encapsulation can be turned off. Decapsulation can still operate, can receive both MPPDU and other frames (subject to management control). Walk through the trivial deployment steps. PrY cannot encapsulate for some destinations, and not for others (separation functionality rapidly approaches that of a bridge, forward reference to multi-component model).>>

<<Introduce the important use case over provider networks, keeping it simple to begin with (one port).>>

<<Extend that use case to the EDE model. PrY in the same component as the SecY.>>

<<PrY can be in a separate system from the SecY, deployment when EDEs are already in place, EDEs can be separately administered.>>

<<Encapsulation for some destinations and not for others revisited, how do local control protocols work? Do we need a separate 'Uncontrolled Port' as for the SecY, or is the one for the SecY enough. How does this work when the PrY is in a separate system from the SecY - may well need Controlled and Uncontrolled there.>>

17.5 Network configuration

<<Relationship between PrYs constitutes a CA, as in 7.1. How do PrYs find each other, what (if anything) do they need to know about the other PrYs in the CA. MKA can carry information when PrYs are collocated with SecYs, specify necessary additional elements for this. Use manual configuration for other cases.>

17.6 Security considerations

<<Although PrYs protect privacy for the data they carry, their own identities can be revealed through their MAC Addresses and/or through characteristics of the MPPDUs (where the timing and padding algorithms are in anyway distinctive). This may facilitate attacks against known vulnerabilities of particular implementations.>>

<<By providing privacy for user data, MPPDUs can hide the characteristics of traffic from devices (such as firewalls) that use those characteristics to identify malicious traffic. Care needs to be taken to ensure that a PrY does not become part of the attack surface.>>>

18. MAC Privacy-protecting Protocol

MAC Privacy-protecting Protocol Data Units[8] (MPPDUs) are used with MACsec, as described in Clause 17, to enhance the privacy of communication using the MAC Service.

This clause describes protocol design (18.1) and support (18.2) requirements, and how they are supported. The encoding of MPPDUs is specified in Clause X, and their use by Privacy-protecting enties in Clause Y.

<<In developing this clause we have to watch for excessive overlap with Clause 17 and in particular with 17.3. Some of what has already been said in can be assumed as context, e.g. we don't have to start from a blank slate when explaining the requirement. However initial examination of what needs to be said overall indicates that there are points to be covered here which do not naturally arise in Clause 17, and if relocated there would lead to a lack of focus in that clause, plus a lack of clarity as to what is not conveyed in the protocol. Notably the protocol only conveys user data frames, and padding (and optionally, fragments of user data frames) and not timing information, though it is important to note that timing can play an important part in a PrYs decision as to what padding to include in an MPPDU. >>

<<This clause should not dive into MPPDU encoding specifics, that is left for Clause. The overview of operation should also not specify the details of the management controls used by the PrY, that should be left to Clause 17.3>>

18.1 Protocol design requirements

The structure of each MPPDU is self-describing, allowing a recipient of an MPPDU to recover user data frames from the MPPDU without the need to share additional parameters with its source. MPPDUs and their use meet requirements for the following aspects of operation:

a) Applicability (18.1.1)
b) Privacy protection (18.1.2)
c) Priority and traffic class support (18.1.3)
d) Coexistence, interoperability and deployment (18.1.4)

18.1.1 Applicability

MPPDUs can encapsulate user data frames in all networks where MACsec can operate and does not compromise MACsec's ability to meeting the requirements described in 8.1.

<<Minimal additions to frame size, as with VLAN-tagging allowing the complete MPPDU to be transmitted

18.1.2 Privacy protection

MPPDUs provide the following:

e) User data frame address privacy
f) User data frame size privacy ()
g) User data frame timing privacy ()

Tradeoffs between the degree to which frame size and frame timing privacy is provided are discussed in 17.3 <<possibly with additional discussion in this clause>>. <<The source of the MPPDU is free to make those tradeoffs, without negotiation, with the exception of noting whether the intended decapsulators can reassemble fragments.>

[8] I am attempting to avoid defining an acronym for 'MAC Privacy-protection Protocol'. I don't think it warrants new alphabet soup. Just referring to MPPDUs where a short form is required should also discourage growth of the protocol in unwanted directions.

18.1.3 Priority and traffic class support

<<MPPDUS need to be able to be used in a way that provides priority/traffic class support. Answers: Any individual MPPDU is handled as a unit when forwarded through a network, so is naturally handled (at each step) with one level of priority. The MPPDU itself carries no indication of priority, but can be VLAN-tagged, just as any other frame can [though the places in the network where this is likely to be applied are few and very limited in scope - see security considerations/attack surface for explanation]. Similarly when the MPPDU is MACsec'd it can be assigned to an SCI that reflects its priority and a subsequent VLAN tag applied that include the priority. This has consequenes not on the MPPDU, but on the packing of user data frames into MPPDUs. A higher priority user data frame may be packed into a different MPPDU than has already been started for a lower priority frame, and then repacking into the lower priority frame resumed. There is some sensitivity to the MACsec implementation here (whether it is streaming directly into transmission) or not. The whole is going to need careful specification in the PrY clause, though not right here in this clause.>>

18.1.4 Coexistence, interoperability, and deployment

<<Various topics/aspects to be covered here. Explicit indication of data length (no minimum frame size/802.3 dependence. Minimum fragment size (possibly a topic for elsewhere - premeption rules may be useful here. Explicit length indication pre-empts simple cut-through, although back-to-back fragments in a single MPPDU might be permitted [there are Cipher Suite consideration here that should be taken into account before spending much time on that]. Normal coexistence/protocol identifications rules. Controlled/Uncontrolled ports (relabel as Protected/Unprotected in this case) allows PrY to transmit user data frames without packing into MPPDU. Simultaneous reception/decapsulation of MPPDUs and single user data frames (subject to management control). Depoyment one LAN at a time/Incremental deployment.>>

18.2 Protocol support requirements

The support of MAC Privacy protection places requirements on the secure system of which each PrY forms part, and on the functionality of authentication, authorization, and key agreement protocols supporting the SecY that protects its MPPDUs SecY, for the following:

a) PrY identification (18.2.1)

b) Peer PrY authentication and authorization (18.2.2)

and when the PrY supports fragmentation of user data frames across MPPDUs

c) User data fragment identification (18.2.3).

When the PrY is colocated with its associated SecY , MACsec Key Agreement (MKA) can be used to discover peer PrYs and reduce the requirement for administrative configuration (18.2.4).

When the PrY and its SecY are not colocated with the SecY that protects its MPPDUS, the connectivity between the PrY and its Secy is constrained, and additional administrative configuration is required (18.2.5).

18.2.1 PrY identification

<<start by modifying the following text>>

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.

18.2.2 Peer PrY authentication and authorization

<<when colocated, can leverage SecY authentication and authorization>>

18.2.3 User data fragment identification

<<Need to deal with a rebooted PrY here. Various approaches. Require colocation (same system) for fragmentation, in which case SecY labelling/reboot might help. Otherwise very long labels might be required.>>

18.2.4 Peer PrY discovery

<<MKA can include the additional parameter(s) as necessary. Can also communicate whether PrY can reassemble user data frame fragments. The KaY accepts indications of PrY capability via the LMI, and vice versa, see AE Clause 8 text for a model.>>

18.2.5 PrY and SecY not colocated

<tbs>

19. Encoding of MAC Privacy-protection Protocol Data Units

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

a) Specifies rules for the structure, representation, and encoding of protocol fields (19.1)
b) Specifies the MPPDU components (19.1, 19.5–19.9)
c) Documents the allocation of the MAC Privacy-protection EtherType that identifies MPPDUs (19.3)
d) Describes provisions that allow specification of additional MPDU component types, if required, protocol version interoperability and version numbering (19.4)
e) Specifies rules for the generation of MPPDUs using the specified components (19.10)
f) Specifies validation and extraction of MPPDU components on reception (19.11)

NOTE—The MPPDU validation checks specified do not overlap with the specification of PrY operation (Clause 10).

19.1 Structure, representation, and encoding

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

19.2 MMPDU components

Each MPPDU comprises a MAC Privacy-protection EtherType (MPP Ethertype, 19.3) followed by one or MPPDU components. Each component comprises a component identifier (MPPCI, 19.3) followed by or more octets. The MPPCI identifies the type and length of the component.

This standard defines the following MPPDU components:

a) Encapsulated Frame (19.5)
b) Trailing Pad (19.6)
c) Explicit Pad (19.7)
d) Encapsulated Frame Fragment ()

Figure 19-1 illustrates the general format of MPPDU components, and the Encapsulated Frame, Trailing Pad, and Explicit Pad components.

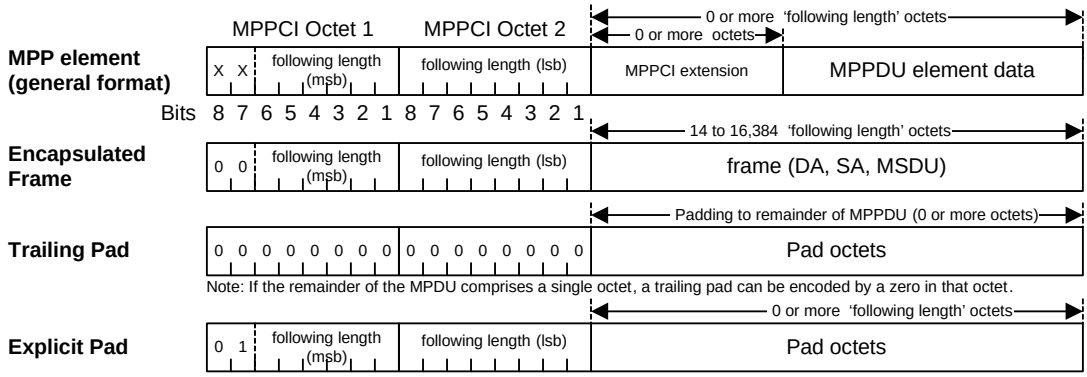


Figure 19-1—MPPDU components

NOTE 1—The MPDU does not include the source and destination MAC addresses of the communicating PrYs, as these are separate parameters of the service requests and indications that supports its transmission and reception.

NOTE 2—The encoding of Privacy-protection components allows conformant implementations to discard unrecognized components without discarding the remainder of a received MPPDU.

NOTE 3—The ability to encapsulate frames of between 14 and 16,384 octets in length does not imply that any specific media access control method can support frames with that entire range of lengths.

Encapsulated Frames and Explicit Pads can each be present zero or more times in an MPDU and their relative position(s) in the MPDU are not constrained. If a Trailing Pad is present, it is the last component in the MPPDU. Figure 19-2 shows some possible MPPDUs.

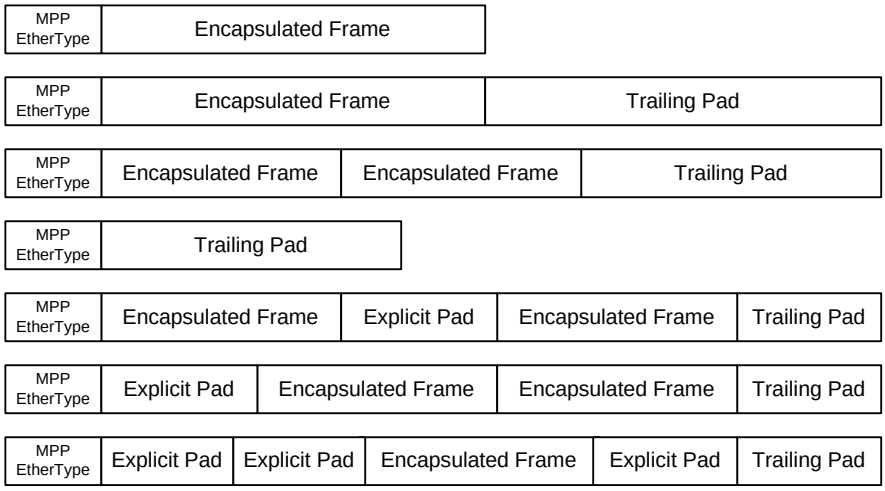


Figure 19-2—MPPDU examples

In the first MPPDU example (in Figure 19-2), a single frame is encapsulated to conceal (once the MPPDU has been MACsec confidentiality protected) it source and destination MAC addresses. The second example adds a trailing pad, reducing an adversary's ability to draw conclusions from the size of the encapsulated frame. If all MPPDUs are to be padded to a constant size, or some limited number of sizes, bandwidth can be used more efficiently by encapsulating a number of small frames in a single MPPDU, as shown in the third example. Maintaining an apparently constant, or near constant, level of activity can be desirable, so MPPDUs that contain only padding can be transmitted, as in the fourth example. The use of the Trailing Pad component allows the MPPDU to be terminated whenever there is a frame available for encapsulation in a following MPPDU. To reduce the delay imposed on an encapsulated frame that becomes available for transmission soon after transmission of a padding frame has begun, an MPDU can begin with one or more Explicit Pad components, followed by a combination of Encapsulated Frames and Explicit Pads.

NOTE 4—Not all PrYs and PrY configurations can transmit MPPDUs with all the formats shown in Figure 19-2, but all PrYs are capable of decoding these MPDUs and extracting the encapsulated frames.

Figure 19-2 shows an MPPDU that has been confidentiality protected by MACsec. The MACsec Secure Data includes the MPP Ethertype, so it is not necessarily clear to an observer that it includes privacy protected information.

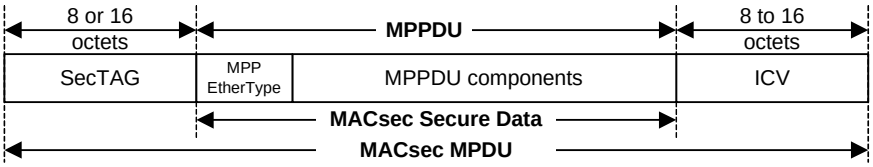


Figure 19-3—MACsec protected MPPDU components

19.3 MAC Privacy-protection EtherType

The MAC Privacy-protection EtherType (MPP Ethertype, Table 19-1) comprises octet 1 and octet 2 of each MPPDU. It is included to allow

a) Coexistence of MAC Privacy-protection capable systems in the same environment as other systems

b) Incremental deployment of MAC Privacy-protection capable systems

c) Transmission of MAC Privacy-protected information on the same media and between the same systems, using the same systems addresses

Table 19-1—MPP EtherType allocation

Name	Value
MAC privacy-protection EtherType	To be assigned at Standards Assocation ballot time

The encoding of the MAC Privacy-protection EtherType in the MPDU is illustrated in Figure 19-4.

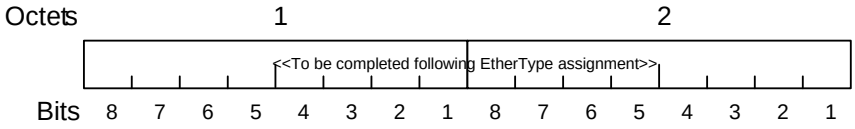


Figure 19-4—MPP EtherType encoding

19.4 Protocol versions

MPPDUs do not include an explicit version number field. The encoding of MPPDU components is extensible and permits the future definition of additional components, while allowing implementations that do not recognize those components to identify their length and discard them without discarding the remainder of a received MPPDU. For example, if bit 8 of octet 1 of the MPPCI is set, the MPPCI extends to octet 3 and possibly further octets of a component. Those further octets can include further component type and MPPCI extension information.The *following length* count encoded in bits 6 through 1 of octet 1 and bits 8 through 1 of of octet 2 includes octet 3 and any additional octets of MPPCI as well as other octets that are part of the component.

Protocol version numbers are assigned, for use by management and in related protocols, to identify implementation capabilities. Additional parameters can be used to further qualify capabilities. This edition of this standard specifies the following protocol implementation versions:

a) MAC Privacy-protection version 0 without fragmentation capabilities

   These implementations recognize and process, and can generate, Encapsulated Frames, Trailing Pads, and Explicit Pads.

   The following possible MPPCI encodings are not recognized

   1) Bit 8 of octet 1 set to 1.

b) MAC Privacy-protection version 0 with fragmentation capabilities

   These implementations recognize and process, and can generate, Encapsulated Frames, Trailing Pads, Explicit Pads, Initial Frame Fragments, and Final Frame Fragments.

   The following possible MPPCI encodings are not recognized

1) Bits 8 and 7 of octet 1 both set to 1
2) Bit 8 of octet 1 set to 1, with bit 8 of octet 3 set to 1.

19.5 Encapsulated Frames

An MPPDU component is identified as an MPP Encapsulated Frame if

a) Bit 8 and bit 7 of octet 1 of the component are both zero, and
b) Bits 6 through 1 of octet 1 (more significant) and bits 8 through 1 of octet 2 (less significant) encode a non-zero *following length* value

Octets 3 through 8 encode a 48-bit MAC destination address in the standard encoding, i.e. in the canonical format (Figure 10, 8.2, and Annex C of IEEE Std 802-2014), and octets 9 through 14 encode a 48-bit MAC source address in the standard encoding. Octets 15 and 16 are the initial octets of the MSDU (the Length/Type field of IEEE Std 802.3 frames), and subsequent octets are the remaining octets of the MSDU.

If the *following length* is less than 14, or exceeds the number of MPPDU octets remaining, the MPP Encapsulated Frame is incorrectly encoded.

19.6 Trailing Pads

An MPPDU component is an MPP Trailing Pad if

a) Octet 1 and octet 2 of the component are both zero, or
b) Octet 1 is zero, and there is no octet 2 (there are no more octets in the MPPDU).

Any octets in the MPDU following octet 1 of the Trailing Pad are part of that Trailing Pad.

19.7 Explicit Pads

An MPPDU component is an MPP Explicit Pad if

a) Bits 8 and 7 of octet 1 of the component are both zero.

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

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

NOTE—The specification of the Explicit Pad component accomodates the possibility that the MPPDU includes fewer octets than encoded in the length field in octets 1 and 2.

19.8 Initial Frame Fragments

An MPPDU component is identified as an Initial Frame Fragment if

a) Bit 8 of octet 1 of the component is one, and bit 7 of octet 1 of the component is zero, and
b) Bits 8 and 7 of octet 3 are both zero.

Figure 19-5 illustrates the format of Initial and Final Frame Fragments..

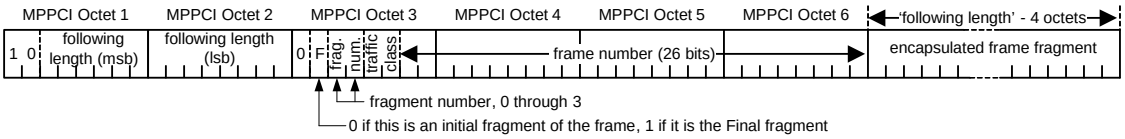


Figure 19-5—Initial and Final Frame Fragments

Bits 6 through 1 of octet 1 (more significant) and bits 8 through 1 of octet 2 (less significant) encode the *following length*, i.e. the number of octets following in the Encapsulated Frame Fragment after octet 2. If the *following length* is less than 68, or exceeds the number of MPPDU octets remaining, the component is incorrectly encoded.

Bits 7 through 5 of octet 3, carry the *traffic class* associated with transmission of the MPPDU. Bits 4 through 1 of octets 3, and bits 8 through 1 of octets 4, 5, and 6, carry the 28-bit *frame number*.

NOTE 1—Frames whose destination address, source address, and MDSDU comprise fewer than 128 octets are not fragmented. A frame is not fragmented into more than 2 components.

NOTE 2—The Initial Frame Fragment and the Final Frame Fragment components for any given fragmented frame convey the same *traffic class* value and *frame number*, and other MPPDUs of that priority are transmitted by a given PrY, or any other given PrY, between the initial and final fragments.

NOTE 3— A PrY transmitting MPPDUs at 1 Tb/s, with each MPPDU capable of conveying an unfragmented 1518 octet frame, in fact conveying a Final Frame Fragment followed by an Initial Frame Fragment will not wrap the *frame number* sequence space in less than 0.8 seconds.

The *following length – 4* octets beginning with octet 7 comprise the fragmented frame's destination address, source address, and the initial octets of the MSDU encoded as specified for an Encapsulated Frame component (19.5)].

An Initial Frame Fragment is the last MPPDU component in any MPPDU, any following octets are pad octets.

19.9 Final Frame Fragments

An MPPDU component is identified as a Final Frame Fragment if

a) Bit 8 of octet 1 of the component is one, and bit 7 of octet 1 of the component is zero, and

b) Bit 8 of octet 3 is zero, and bit 7 of octet 3 is one.

Bits 6 through 1 of octet 1 (more significant), and octets 2 through 6, encode the *following length*, the *traffic class*, and the *frame number* as specified for Initial Frame Fragments (19.8). If the *following length* is less than 68, or exceeds the number of MPPDU octets remaining, the component is incorrectly encoded.

The *following length – 4* octets beginning with octet 7 comprise the final octets of the fragmented frame's MSDU.

If a Final Frame Fragment is encoded in an MPPDU it shall be the the first component in that MPPDU.

19.10 MPPDU generation

The generation of MPPDUs is specified as part of the operation of a Privacy-protecting entity (PrY, Clause N). While not specifying desired behavior, the following rules constrain and permit aspects of the valid generation of MPPDUs:

a) All MPPDUs shall include the MPP EtherType as the first two octets.

b) An MPPDU can only contain components specified for the MAC Privacy-protection protocol version of the transmitting PrY.

c) Any Encapsulated Frame component included in the MPPDU shall be present in its entirety.

d) An Explicit Pad component can be encoded with a *following length* that exceeds the number of octets remaining in the MPPDU.

e) All pad octets in a Trailing Pad or an Explicit Pad shall have the value zero.

NOTE—This requirement guards against the accidental inclusion of data previously subject to privacy protection in a subsequent MPPDU, sent to the same or a different PrY. It is not checked as part of MPPDU validation, but is required for conformance.

19.11 MPPDU validation

The first two octets of each received MPPDU are the MPP EtherType. Each of the following possible MPPDU components is processed in the order they are encoded in the MPPDU as follows:

a) If zero or one MPPDU octets remain to be processed, validation of the MPPDU is terminated

b) If the initial octet(s) of the component identify it as an Encapsulated Frame (19.5)

1) If the *following length* is greater than that of the remaining octets of the MPPDU, the counter <EncapError> is incremented, and validation of the MPPDU is terminated; otherwise

2) If the *following length* is less than 14, the counter <EncapError> is incremented, the Encapsulated Frame is discarded, and validation proceeds with the next component; otherwise

3) The encapsulated frame is extracted, and validation proceeds with the next component

c) If the initial octet(s) of the component identify it as a Trailing Pad (19.6), the number of pad octets (the Trailing Pad MPPCI and the remaining octets in the MPPDU) is added to the <PadOctetsCount> and validation of the MPPDU is terminated.

d) If the initial octet(s) of the component identify it as an Explicit Pad (19.7)

1) If the *following length* is greater than the number of the remaining octets of the MPPDU, the latter number is added to the <PadOctetsCount> and validation of the MPPDU is terminated; otherwise

2) (the Explicit Pad MPPCI and the following pad octets) is added to the <PadOctetsCount> and validation proceeds with the next component.

e) If the initial octets of the component identify it as an Initial Frame Fragment (19.8)

1) If the *following length* is greater than the number of the remaining octets of the MPPDU, the counter <FragError> is incremented and validation of the MPPDU is terminated; otherwise

2) If the *following length* is less than 68, the counter <FragError> is incremented and validation proceeds with the next component; otherwise

3) The encapsulated initial frame fragment and its priority field are extracted, and validation proceeds with the next component.

NOTE—A PrY may lack the ability to reassemble frame fragments, and hence may not recognize an Initial or Final Fragment componnet (<ref>). The processing of subsequent MPPDU components does not depend on the recognition, or otherwise of an Initial Frame Fragment component.

f) If the initial octets of the component identify it as an Final Frame Fragment (19.8)

1) If the *following length* is greater than the number of the remaining octets of the MPPDU, the counter <FragError> is incremented and validation of the MPPDU is terminated; otherwise

2) If the *following length* is less than 68, the counter <FragError> is incremented and validation proceeds with the next component; otherwise

3) The encapsulated initial frame fragment and its priority field are extracted, and validation proceeds with the next component.

g) If the component type is not recognized

1) If the *following length* equals or exceeds the number of the remaining octets of the MPPDU, the counter <UnknownMPPCI> is incremented, and validation of the MPPDU is terminated; otherwise

2) If the *following length* is less than the number of the remaining octets of the MPPDU, the counter <UnknownMPPCI> is incremented and validation proceeds with the next component.

Once MPPDU validation is terminated, the MPPDU and any components not specifically identified as extracted are discarded. Any extracted components are retained for further processing.

20. MAC Privacy-protecting Entity (PrY) operation

<<Text below this point has not yet been worked on>>

This clause

a) Provides an overview of the PrY (20.1), the service that it provides, and its relationship to other entities in a secure system.

b) Describes the functionality of the PrY (20.2).

c) Provides a model of operation (20.3) comprising an architecture (20.4) and its constituent processes (20.5 through 20.7) that supports the detailed functionality including management controls.

d) Details the addressing requirements and specifies the addressing of PrYs (20.8).

<<Transmission of multicast frames. Use same address as the supporting SecY.>>

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.

20.1 SecY overview

Each SecY uses the MAC Service provided by a Common Port (20.4) 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 20-1).

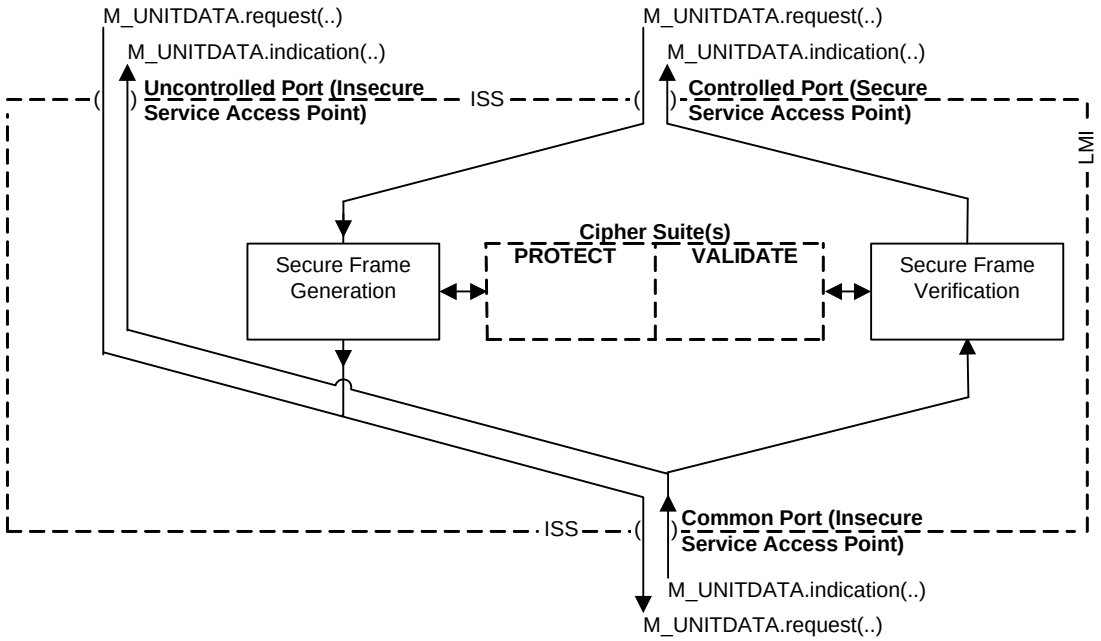


Figure 20-1—SecY

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 20-3.

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 (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 (20.7.4, 10.7.5) consistent with the provisions of this standard (6.4, 6.5, 6.7, 7.1.3, 7.2, 20.5.1, 10.5.2, 20.7.14, 10.7.2, 10.7.25).

The KaY communicates transmit and receive keys and other information (20.2) 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 (20.1, 20.3, 20.4, Figure 20-1), 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.

20.2 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.

20.3 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.

20.4 SecY architecture

A SecY uses an instance of the MAC Internal Sublayer Service (ISS) (see 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 20-2, and comprises

a) The Controlled, Uncontrolled, and Common Ports together with their MAC Status parameters.

b) The Secure Frame Generation process (20.5).

c) The Secure Frame Verification process (20.6).

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 FCS Regenerators.

g) A SecY Management process (20.7).

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. .

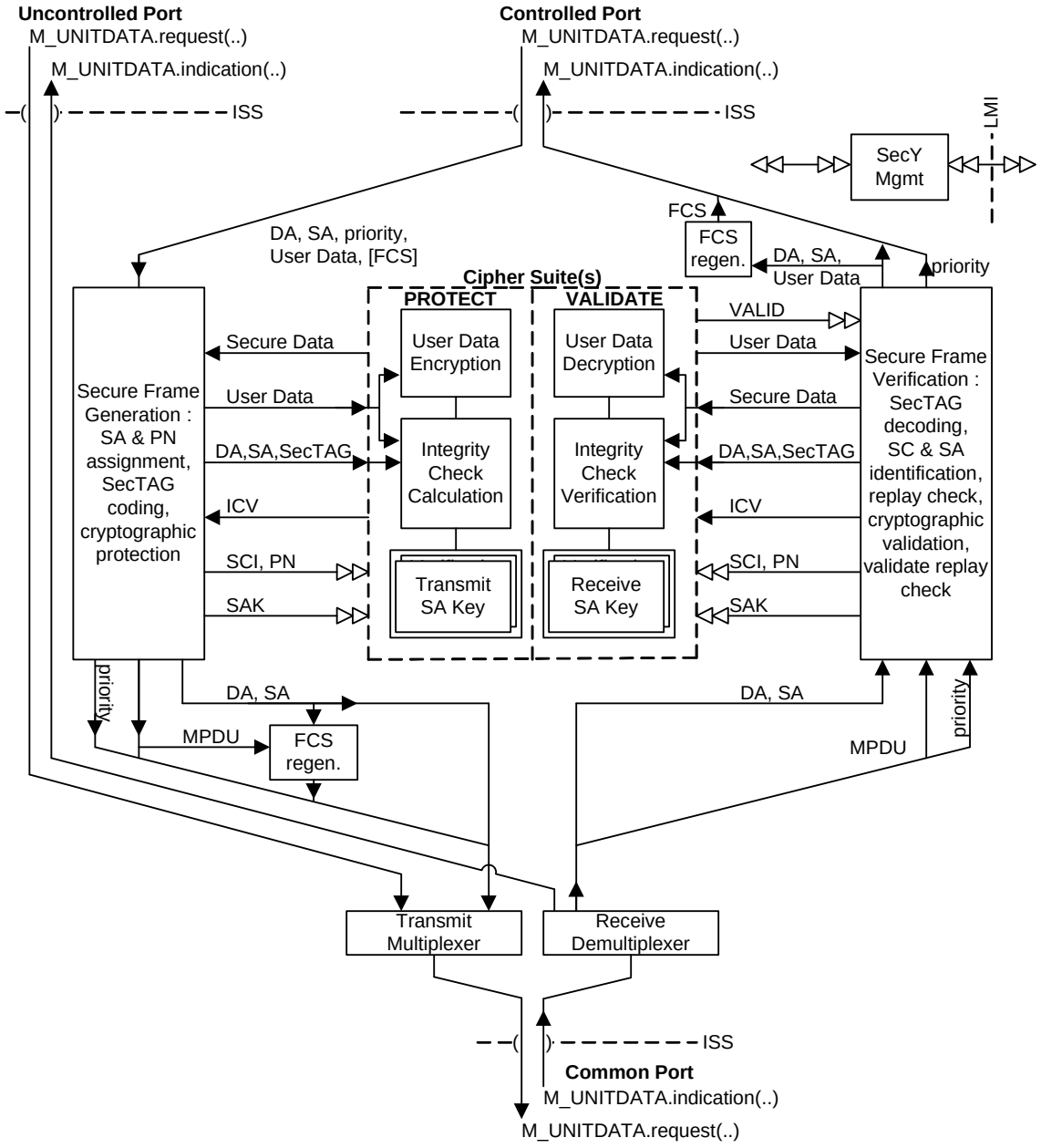


Figure 20-2—SecY architecture and operation

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 20-3 and Figure 20-4.

A frame check sequence (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 integrity check value (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.

20.5 Secure frame generation

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

a) Assigns the frame to an SA (20.5.1)

b) Assigns the nextPN variable for that SA to be used as the value of the PN for that protected frame (20.5.2)

c) Encodes the octets of the SecTAG including the least significant 32 bits of the PN in the PN field (20.5.3)

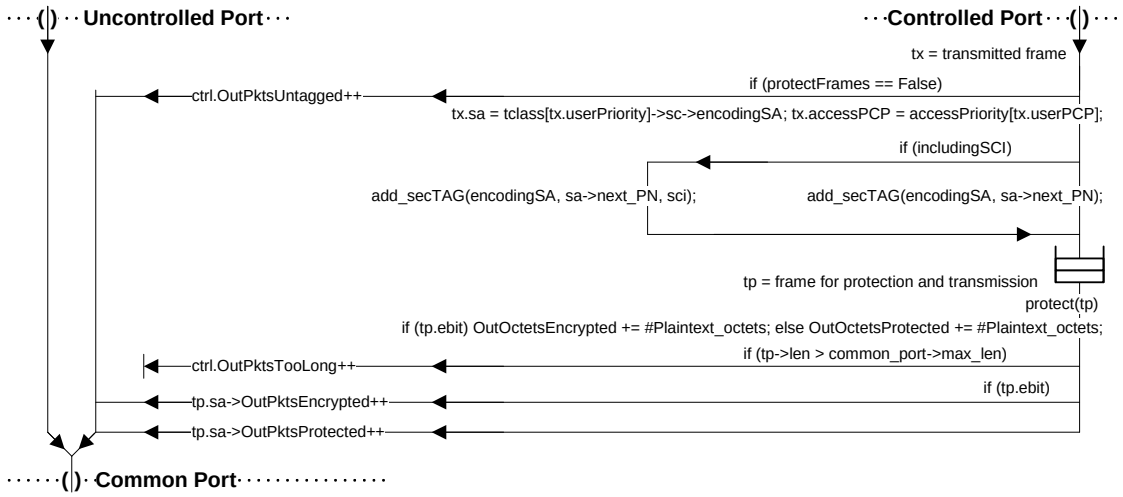
d) Provides the protection function (14.1, 20.5.4) of the Current Cipher Suite with
   1) The SA Key (SAK)
   2) The SCI for the SC used by the SecY to transmit
   3) The PN
   4) The SecTAG
   5) The sequence of octets that compose the User Data

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 (20.5.5). If the SecY does not implement an Access Priority Table (20.7.17) 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.



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 20-3—Management controls and counters for secure frame generation

20.5.1 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 20.7.17).

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.

20.5.2 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.

20.5.3 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 20-1 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 20-1—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.

20.5.4 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).

20.5.5 Transmit request

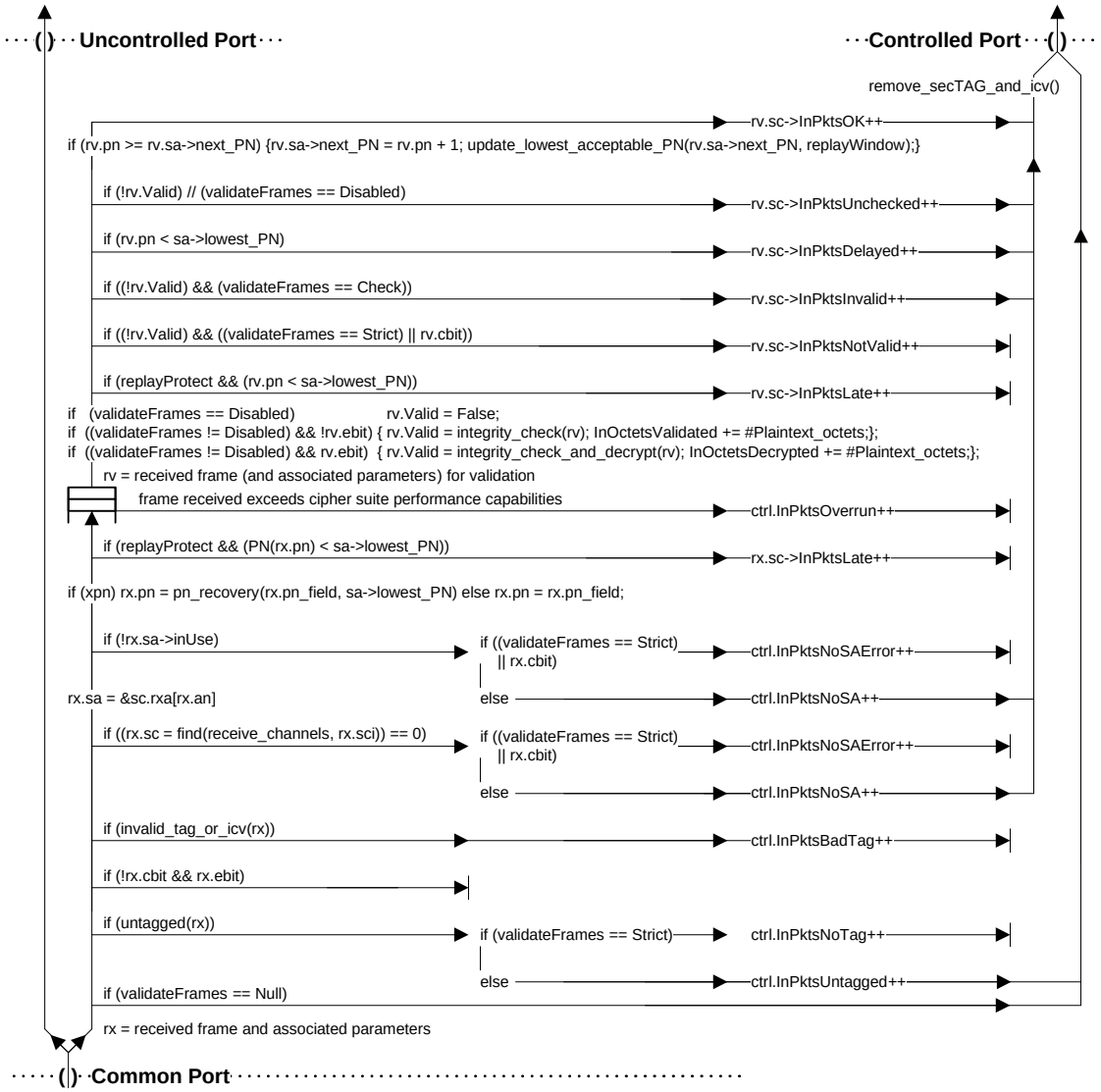
If the MPDU composed of the concatenated octets of the SecTAG, Secure Data, and ICV exceeds the size of the MSDU supported by the Common Port, the frame is discarded and a counter incremented. Details of the discarded frame may be recorded to assist network management resolution of the problem. Otherwise,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

20.6 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 (20.6.1)
f) Recovers the PN and performs a preliminary replay check against the last validated PN for the SA (20.6.2)
g) Provides the validation function (14.1, 20.6.3) of the Current Cipher Suite with
    1) The SA Key (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 (20.6.4)
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 (20.6.5).

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



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 20-4—Management controls and counters for secure frame verification

modification, irrespective of the absence, presence, or validity of a SecTAG, and the processing described in a) through j) above and in 20.6.1 through 20.6.5 is not performed. Figure 20-4 summarizes the operation of secure frame verification management controls and counters.

Setting validateFrames to Null shall also cause the secure frame generation control protectFrames (20.5) 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).

20.6.1 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.

20.6.2 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 20-2 provides examples.

Table 20-2—Extended packet number recovery (examples)

SecTAG PN field value	0x 2A2B 5051
Lowest acceptable PN	0x 0000 0007 1234 DEF0

Table 20-2—Extended packet number recovery (examples)

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 (20.6.5, 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 The XPN recovery algorithm [B11].

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. MKA, the MACsec Key Agreement protocol 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 20.6.5) 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.

20.6.3 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.

20.6.4 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 below (20.6.5). Implementations can process frames as convenient, provided the externally observable result is the same.

20.6.5 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.

20.7 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 (20.7.1)
b) Maintains the MAC Status (6.4) parameters and point-to-point MAC parameters (6.5) for the Uncontrolled (20.7.2) and Controlled (20.7.4) Ports
c) Provides interface statistics for the Uncontrolled (20.7.3) and Controlled Ports (20.7.6), deriving the latter from the detailed statistics maintained by the SecY
d) Provides information on the frame verification (20.7.7) and generation (20.7.16) capabilities
e) Supports control of frame verification (20.7.8) and generation (20.7.17), 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 (20.7.20), each corresponding to one of the values appearing in Traffic Class Table entries
g) Supports creation of transmit SAs (20.7.22), each associated with an SAK, for the transmit SC
h) Supports creation of receive SCs (20.7.11), each corresponding to potential member of the CA
i) Supports creation of receive SAs (20.7.13) for each receive SC, each associated with an SAK
j) Supports control over reception (20.7.15) and transmission (20.7.24) 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 (20.7.25)

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 20-5 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 *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition* [B6]. The containment relationships in Figure 20-5 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 20-5 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 20.7.13, 20.7.22, and 20.7.28.

In Figure 20-5 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 (RFC 2863) value used to identify a SecY within that module (13.3.2, 13.5).

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 (), 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—The IEEE Std 802.1AEcg-2017 amendment to this standard added the SecY System to Figure 20-5 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

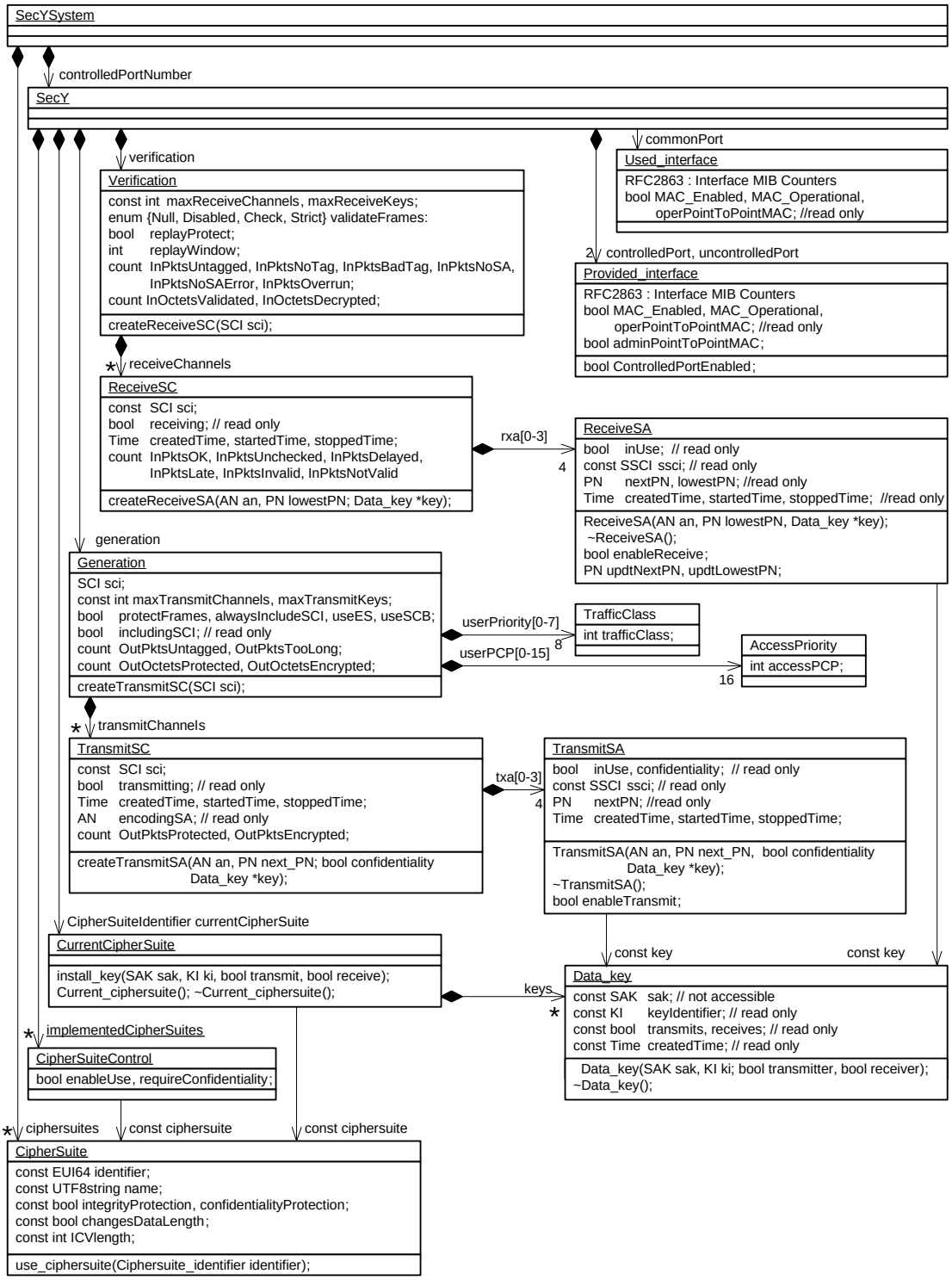


Figure 20-5—SecY managed objects

respect to syntactic elements that are used to describe that interface in this clause. Table 20-3 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.

20.7.1 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 [(e), 20.7.1, 20.7.17], the four most significant bits of the Port Identifier component of this SCI are zero.

20.7.2 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.

20.7.3 Uncontrolled Port statistics

The following statistics are provided to support 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.

20.7.4 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 (20.7.5) is True, and
    2) MAC_Enabled is True for the Common Port, and
    3) transmitting (20.7.21) is True for the transmit SC, and
    4) receiving (20.7.12) 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 (20.7.8) 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 the IEEE Std 802.1AEcg amendment to this standard, 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.

20.7.5 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.

20.7.6 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 (20.6), 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 (20.6, Figure 20-4).

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 (20.5), plus the octets of the destination and source MAC addresses.

The ifOutErrors count is equal to the OutPktsTooLong count (Figure 20-3). If ifOutDiscards is reported as part of RFC 2863 counts, it is zero.

20.7.7 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.

20.7.8 Frame verification controls

Frame verification is subject to the following controls, as specified in 20.6:

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 (20.6) 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 (20.6.2). 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.

20.7.9 Frame verification statistics

Any given received frame increments (20.6) 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 (20.7.12, 20.7.14) and assist correlation of the statistics collected with network events.

20.7.10 Frame validation statistics

Investigation or validation of the performance of the cryptographic functions is supported by maintaining counts of packets (InPktsOverrun, 20.6.3, 20.7.9) 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.

20.7.11 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 (20.7.13) 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.

20.7.12 Receive SC status

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

a) receiving, True if inUse (20.7.14) 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.

20.7.13 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 (20.6, 20.6.5)
c) lowestPN, the lowest acceptable PN value for a received frame (20.6, 20.6.2, 20.6.4, 20.6.5)
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 (20.7.9) 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.

The MACsec Key Agreement (MKA) protocol specified in IEEE Std 802.1X-2010 does not distribute SSCIs explicitly. A KaY that uses MKA as specified in IEEE Std 802.1X-2010 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 the MACsec Key Agreement protocol (MKA) 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).

20.7.14 Receive SA status

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

a) inUse
b) nextPN (20.6, 20.6.5)
c) lowestPN, the lowest acceptable PN value for a received frame (20.6, 20.6.2, 20.6.4, 20.6.5)
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 (20.7.28), 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 802.1X encoded in an octet string as specified by that standard.

20.7.15 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.

20.7.16 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 (20.7.17). 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.

20.7.17 Frame generation controls

Frame generation is subject to the following controls:

a) protectFrames (20.5), True or False, with a default of True

b) alwaysIncludeSCI (20.5.3), True or False, with a default of False

c) useES (20.5.3), True or False, with a default of False

d) useSCB (20.5.3), 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 (20.5.3), 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—The IEEE Std 802.1AEcg-2017 amendment to this standard, introducing the use of multiple transmit SCs, was developed contemporaneously with the IEEE 802.3br-2016 amendment, 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.

20.7.18 Frame generation statistics

Any given transmitted frame (20.5) 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 (20.7.21, 20.7.23) 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.

20.7.19 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.

20.7.20 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 (20.7.22) 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.

20.7.21 Transmit SC status

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

a) transmitting, True if inUse (20.7.23) is True for any of the SAs for the SC, and False otherwise
b) encodingSA (20.5.1)
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.

20.7.22 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 (20.5.2) 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 (20.7.18) 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.

The MACsec Key Agreement (MKA) protocol specified in IEEE Std 802.1X does not distribute SSCIs explicitly. A KaY that uses MKA as specified in IEEE Std 802.1X assigns SSCI values as specified in 20.7.13.

20.7.23 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 (20.5, 20.5.2)
f) confidentiality, True if the SA is providing confidentiality as well as integrity for transmitted frames
g) keyIdentifier (20.7.28), 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 802.1X encoded in an octet string as specified by that standard.

20.7.24 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.

20.7.25 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 (RFC 2279 [B2]) 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 MACsec PDU 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.

20.7.26 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 (20.7.25), with each SecY's capabilities represented by a list of references (each with separate use controls) to some of those sets.

20.7.27 Cipher Suite selection

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

a) currentCipherSuite, the Cipher Suite Identifier (20.7.25) for the cipher suite

If offsetConfidentiality (20.7.25) 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.

20.7.28 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 [B7], a 96-bit parameter provided to the Current Cipher Suite for subsequent protection and validation operations

The MACsec Key Agreement (MKA) protocol specified in IEEE Std 802.1X-2010 does not include explicit parameters for distributing a Salt. Each KaY that uses MKA as specified in IEEE Std 802.1X-2010

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.

20.7.29 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

20.8 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.

20.9 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.

20.10 SecY performance requirements

Table 20-3 places requirements on SecY performance to ensure that MACsec operates correctly.

Table 20-3—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, IEEE Std 802.1Q).

21. MAC Privacy protection in Systems

This clause specifies how MAC Privacy protection is supported within the architecture of

a) End stations (21.2)
b) MAC Bridges (21.3)
c) VLAN-aware Bridges (21.4)
d) Systems that incorporate Link Aggregation (21.5)
e) Systems that incorporate Link Layer Discovery Protocol (21.6)
f) Provider Bridges and VLAN-aware Bridges attached to Provider Bridged Networks (21.7)
g) LANs that provide independently secured access for multiple end stations (21.8)

The figures in this clause illustrate the relative position of components within the interface stacks (21.1) of each of these systems. Both the privacy-protected MAC Service provided by one or more Private Ports and the service provided by the Public Port are shown.

NOTE—An understanding of architectural concepts common to this and other IEEE 802.1 standards is essential to understanding this standard's specification of MAC Privacy protections. The reader is encouraged to review Clause 7 of IEEE Std 802.1AC-2016 and Clause 11 of this standard. For more information on the Public and Private Ports and the operation of the PrY, see Clause 20.

21.1 Privacy-protecting interface stacks

Each PrY uses a MAC Internal Sublayer Service (ISS, IEEE Std 802.1AC) access point and provides the ISS at its Public Port and Private Ports. This allows use of MAC Security with other media-independent functions. However, interoperability between systems using MAC Security requires not only interoperability between Privacy-protecting Entity (PrY) implementations and use of the same LAN MAC technology, but also that the same, or compatible, media interface functions are used in the same relative position within the interface stack, as specified in this clause.

NOTE—An understanding of architectural concepts common to this and other IEEE 802.1 standards is essential to understanding this standard's specification of MAC Privacy protections. The reader is encouraged to review Clause 7 of IEEE Std 802.1AC-2016 and Clause 11 of this standard. For more information on the Public and Private Ports and the operation of the PrY, see Clause 20.

Privacy protection depends not only on the operation of the PrYs that generate and validate the MPPDUs that convey user data frames, but also on MACsec confidentiality-protection of those MPPDUs. A MAC Security Entity (SecY) can directly support a PrY, as in shown in Figure 21-1. Exceptions, where a PrY and its supporting SecY are located in different systems are discussed in <later>.

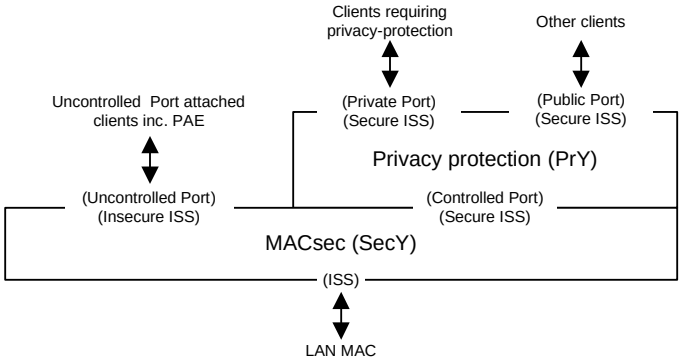


Figure 21-1—A Privacy-protecting interface stack

MAC Service indications received by the PrY with a destination address that matches either the individual address or the group address configured for the PrY (<ref Clause 20>) and that convey the from the SecY's Controlled Port are validated as specified in MPPDUs received by the PrY from the SecY are

21.2 Privacy protection for end station interfaces

<<Limited applicability. Lack of benefit from address hiding, as a temporary address could be chosen without encapsulation. Need for bridges etc. to see frames for resource allocation purposes etc. These thoughts don't really belong here, but in Clause 17. As does the discussion of what clients are attached where. Addressing considerations are probably here though, specifying the use of Nearest Bridge etc. Possibly no use of the Public Port.>>

<<End station can use the interface stack shown in Figure 21-1. See limitations discussed in Clause 17. Addressing considerations.>>

21.3 Privacy protection for bridge interfaces

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 21-2 shows a bridge with and without MACsec.

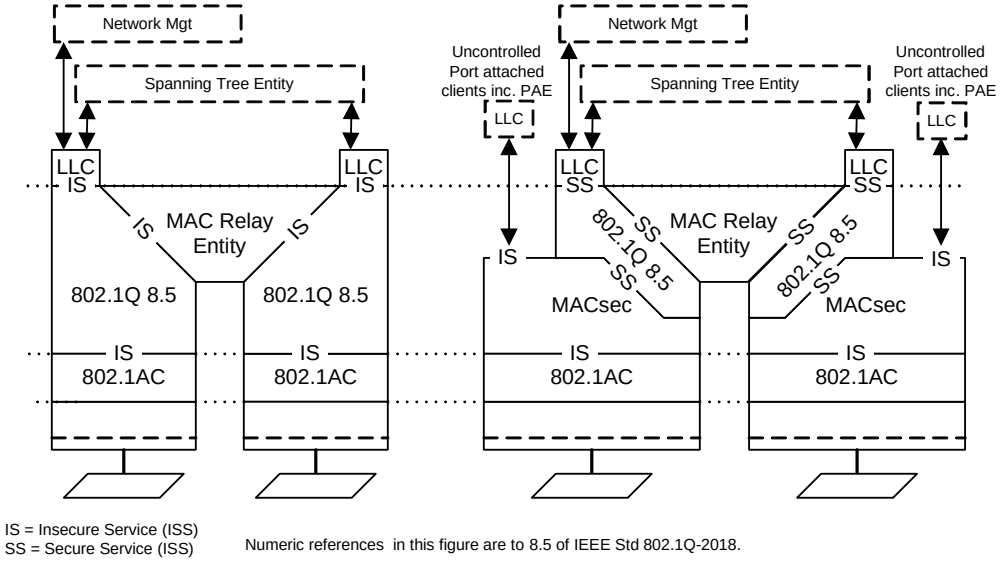


Figure 21-2—MACsec in a VLAN-unaware MAC Bridge

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 21.5. Each aggregated port then provides secure service to the Bridge Port transmit and receive functions.

Figure 21-3 shows the interface stack for each of the Bridge Ports.

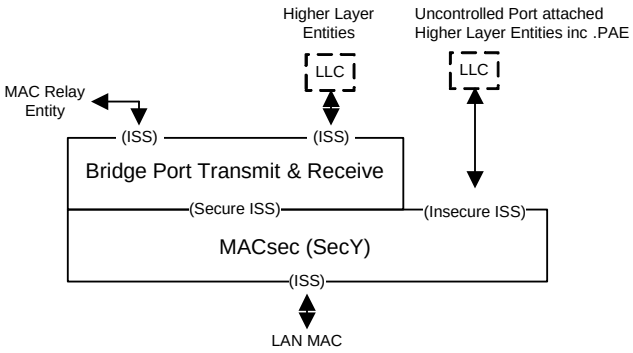
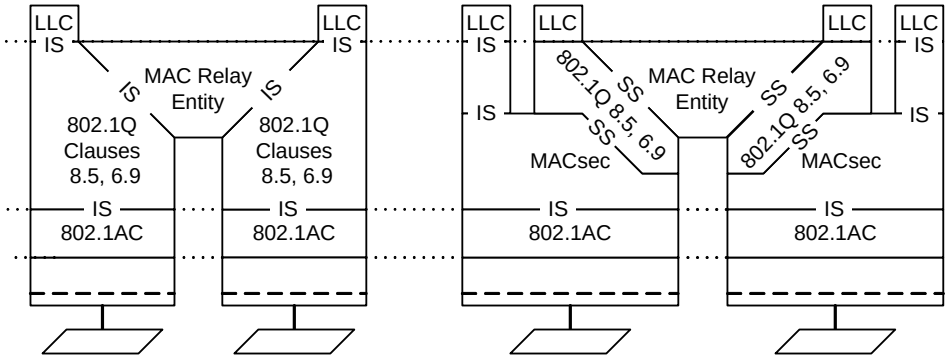


Figure 21-3— VLAN-unaware MAC Bridge Port with MACsec

21.4 MACsec in VLAN-aware Bridges

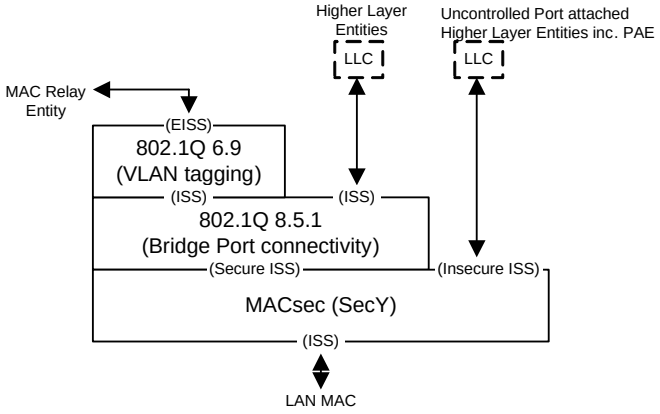
VLAN-aware Bridges are specified in IEEE Std 802.1Q. Figure 21-4 illustrates the addition of MAC Security.



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

Figure 21-4—Addition of MAC Security to a VLAN-aware MAC Bridge

Figure 21-5 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 21-5— 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.

21.5 MACsec and Link Aggregation

Link Aggregation is specified in Clause 43 of IEEE Std 802.3. 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 for details.

Figure 21-6 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.

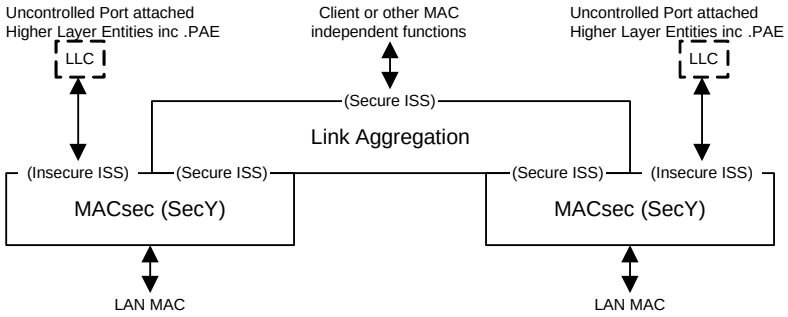
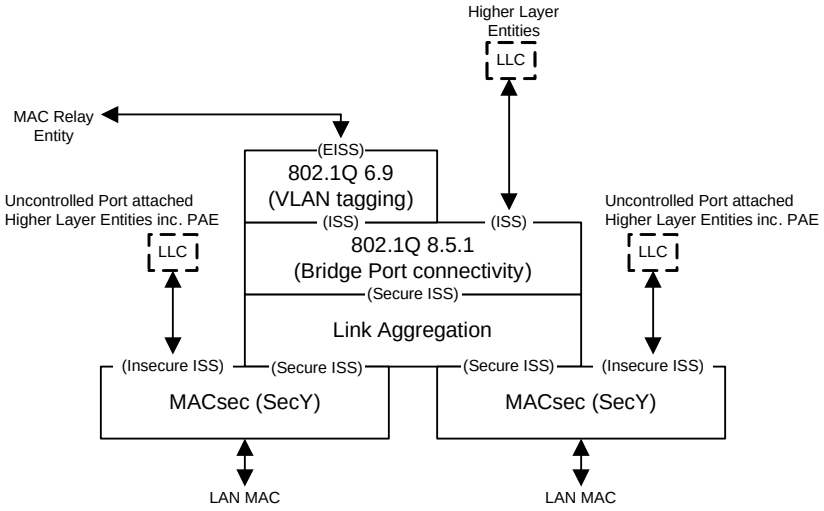


Figure 21-6—MACsec and Link Aggregation in an interface stack

Figure 21-7 shows the addition of link aggregation to the interface stack for a VLAN-aware Bridge Port that also uses MACsec.



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

Figure 21-7—IEEE 802.1Q VLAN-aware Bridge Port with MACsec and Link Aggregation

21.6 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 21-8.

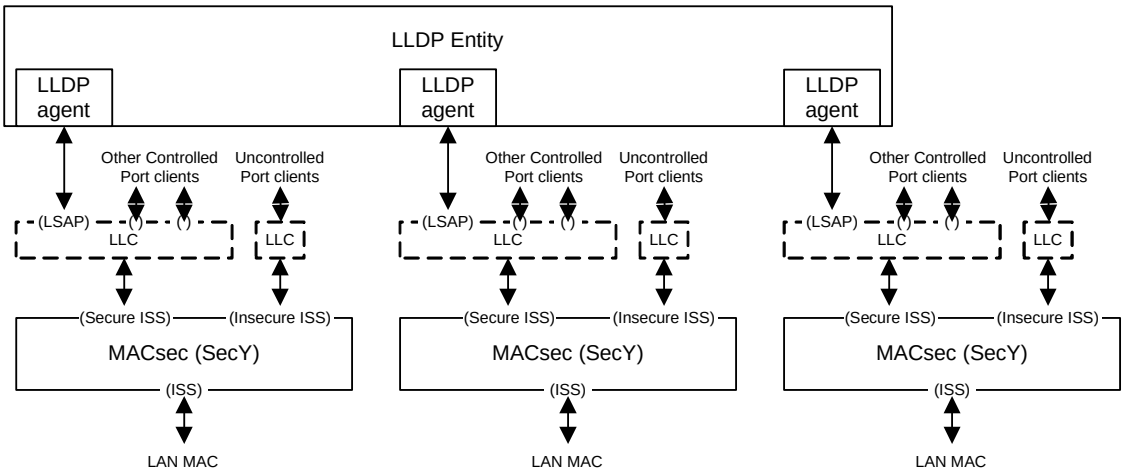


Figure 21-8—MACsec with LLDP

21.7 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 21-9.

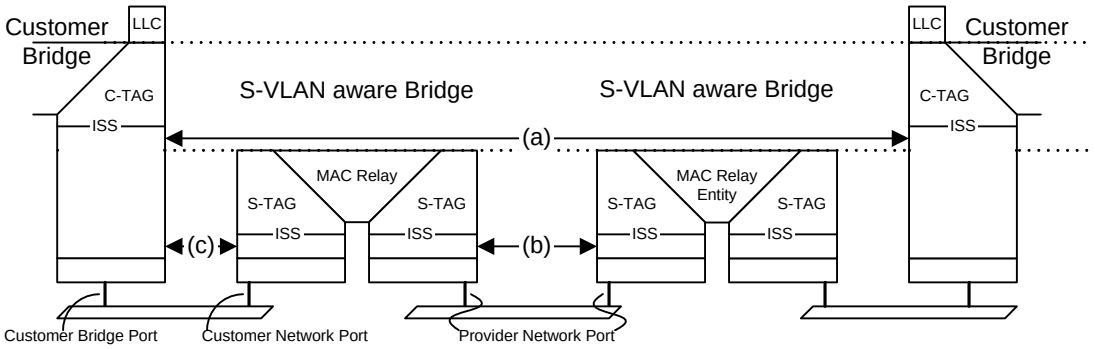


Figure 21-9—Internal organization of the MAC sublayer in a Provider Bridged Network

NOTE—Figure 21-9 is based on Figure 15-1 of IEEE Std 802.1Q.

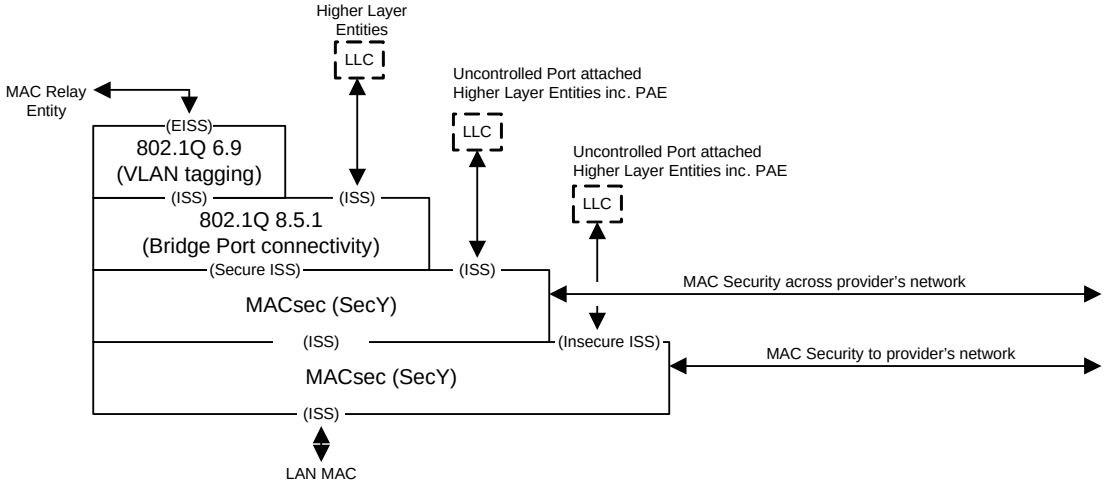
MACsec can be used to secure communication between

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 21-3 (for an end station), Figure 21-3 (for a MAC Bridge), or Figure 21-5 (for a VLAN-aware Bridge) within the customer equipment is sufficient.



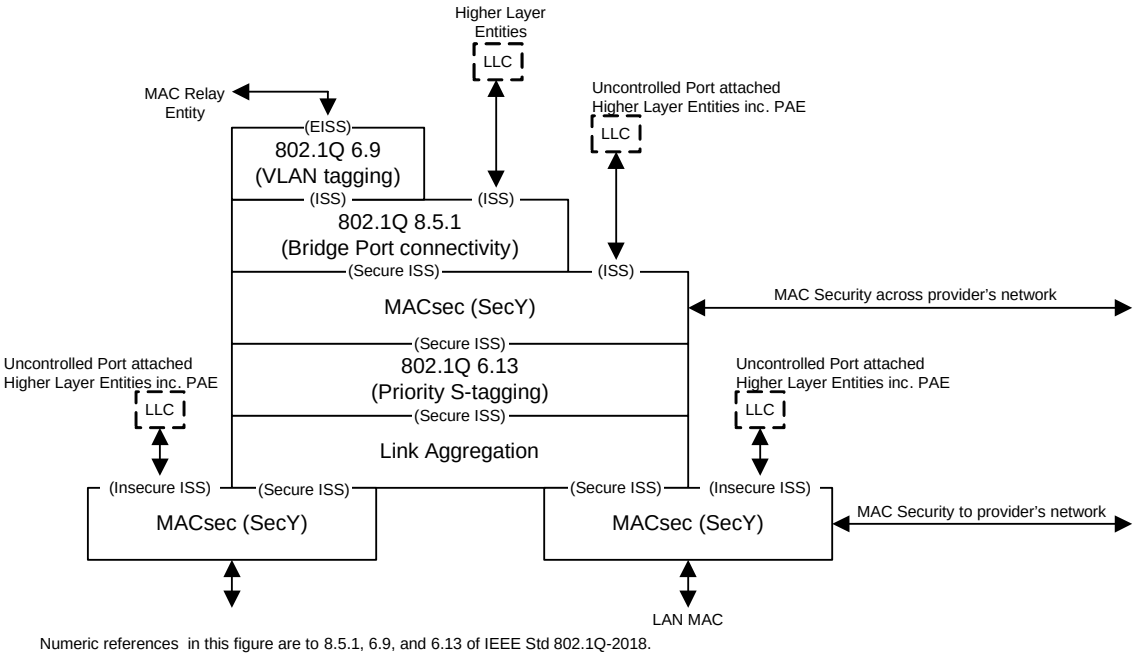
Numeric references in this figure are to 8.5.1 and 6.9 of IEEE Std 802.1Q-2018.

Figure 21-10—Interface stack for MAC Security to and across provider's network

Use of the interface stack illustrated in Figure 21-5 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 21-9), 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 21-10.

Figure 21-11 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 21-10, together with the use of Link Aggregation to support attachment to the provider's network with two LANs.



Numeric references in this figure are to 8.5.1, 6.9, and 6.13 of IEEE Std 802.1Q-2018.

Figure 21-11—Provider network with priority selection and aggregation

21.8 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 21-12 shows one station (A in the figure) with two connections, one to each of two others (B, C).

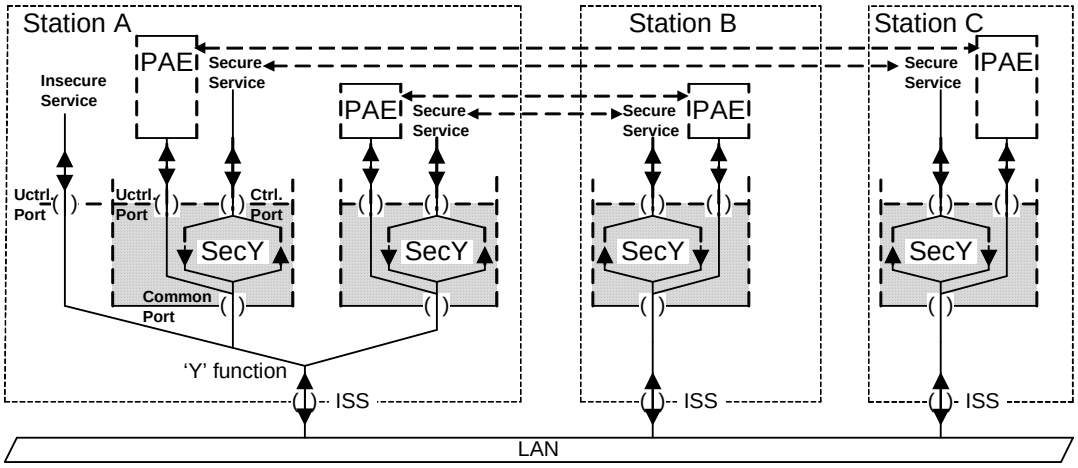


Figure 21-12—An example multi-access LAN

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 the IEEE Std 802.1AEcg amendment to this standard, 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 21-12) 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 21-13 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.

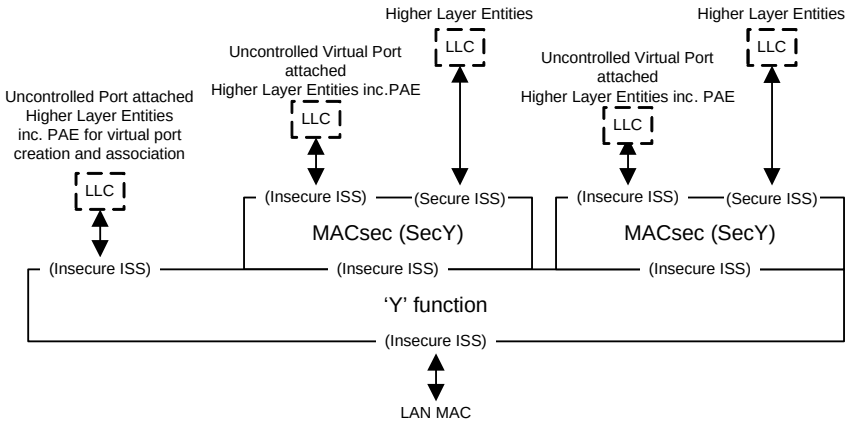


Figure 21-13—Multi-access LAN interface stack

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.

Annex I

(informative)

Privacy considerations in bridged networks

This informative annex describes privacy considerations related to the use, design, and deployment of bridged networks based on IEEE Std 802.1Q and related standards (IEEE Std 802.1X, IEEE Std 802.1AB, IEEE Std 802.1AE, IEEE Std 802.1AR, IEEE Std 802.1AS, IEEE Std 802.1AX, IEEE Std 802.1BA, IEEE Std 802.1BR, IEEE Std 802.1CB, and IEEE Std 802.1CM).

The unintentional or unauthorized disclosure of personal information arises from a combination of the following factors:

a) The use of personal devices that are attached to, or form part of, the network (I.1)
b) The type of information that adversaries might wish to acquire (I.2)
c) The efficient operation and management of the network (I.3)
d) The use of security protocols for authentication, authorization, integrity, and confidentiality (I.5)
e) The frame fields that contain information useful to an adversary, the sophistication of, and the network access afforded to, that adversary (I.5)

Privacy considerations particular to a given referenced standard are discussed in I.6.

This annex is informative. It does not modify the mandatory or optional provisions or the recommendations contained in any referenced standard.

I.1 Personal devices

Privacy, in the context of bridged networks, relates to the use of personal devices i.e. devices used by one person or a small group of people. Information that identifies a personal device or is associated with that device identification can thus yield information about the location and activities of a person.

Shared service devices, in contrast, support applications for a large enough group of people such that correlation between any given person and the observable behavior of the device is weak. Other devices, e.g. sensors in industrial networks, have no direct correlation with a person.

In general IEEE 802.1 standards are applicable to both personal devices and shared service and other devices. However some protocol roles, e.g. Grandmaster in IEEE Std 802.1AS Timing and Synchronization for Time-Sensitive Applications, are unlikely to be associated with personal devices in other than the smallest bridged networks, and are even more unlikely to be associated with mobile personal devices.

I.2 Goals of adversaries

An adversary can be interested in the following personal information:

a) Who is using a personal device (identification)
b) Where are they (location, and location tracking)
c) What are they doing (activity, application use)
d) With whom are they associated (communicating, shared interest).

The information on all, or indeed on any of these, need not be complete to be useful to an adversary. The adversary can, for example, be interested in facts such as:

— an identified person appears to be engaging in the same, unknown, activity as a group of unknown persons at another identified location

— there appears to be no one at an identified location.

The information obtained need not be particularly accurate to be useful to an adversary. It is sufficient that the cost of acquiring the information is less than the benefit expected from its use, allowing for the probability that it is incorrect and any costs associated with the use of incorrect information. Use of incorrect information can negatively affect a targeted person.

I.3 Network operation

Bridged networks support frame based transmission, with variable length frames and without requiring attached stations (except for certain time-sensitive network applications) to adhere rigidly to a clocked transmission schedule. Stations are not obliged to transmit when there is nothing to transmit and frames are not all padded to the same length, so the use of network resources benefits from statistical multiplexing. At the same time some network applications have requirements for timely delivery that cannot be met simply by relying on that multiplexing and increasing transmission speeds but require signaling, to bridges in the network, of the differential service requirements of individual frames. Time-sensitive network applications with more stringent delivery requirements require bandwidth allocation, supported by end station protocols or management configuration, and sufficient information in individual frames for bridges to associate each frame with an allocation (and thus with an individual end station and a particular type of end station application). Bridged networks provide more bandwidth than is available from each of their constituent individual LANs by restricting data frames to paths to their intended destinations. One of a number of alternate paths to a given destination end station or set of end stations can be used to further increase the available bandwidth, but common network application frame ordering requirements constrain the distribution of frames amongst such paths to those that bridges can distinguish as belonging to separate application flows.

Bridges in the network can distinguish between application flows using each frame’s destination MAC address (DA), source MAC address (SA), the VLAN identifier (VID) and priority code point (PCP encoded in the VLAN tag (if present), the EtherType (or LSAP) identifying the higher layer protocol conveyed by the frame, and the initial fields of that protocol. Protocols that operate over the bridged network and are used by personal devices to support network applications and to communicate with application servers and other devices (as opposed to reserving network resources for that communication) typically use the Internet Protocol (IP). It is rare for two personal devices to communicate without transmitting frames via one or more intervening routers. Any given IP subnet is often supported by a single VLAN, so bridges that support parallel paths for routed application flows from individual end stations typically use the source and destination IP addresses, the conveyed protocol type (IP, UDP, or SCTP), and source and destination ports for that protocol (see 9.1.5 of IEEE Std 802.1CB-2017).

Some IEEE Std 802.1 protocols, e.g. IEEE Std 802.1Q Stream Reservation Protocol (SRP), transmit frames with group destination MAC addresses. These addresses identify the type of the intended recipient protocol entity and allow bridges to use address filtering to restrict those frames to an appropriate scope, reaching only the nearest bridge, for example. Some group addresses support a particular type of application, and thus associates the source MAC address (and the station that is using it) with that application.

The deployment and operational costs of bridged networks have been considerably reduced by the use of protocols that volunteer device information (e.g. IEEE Std 802.1AB) even when those protocols are not used to support full ‘plug-and-play’ operation. Management and trouble shooting of faulty devices or apparently incorrect network behavior depends on the recording of device location and gathering statistics on network

use. Stations implementing protocols whose operation depends on the presence of a reachable collaborating peer (e.g. IEEE Std 802.1AS gPTP time synchronization) typically advertise their capabilities, either by using IEEE Std 802.1AB or by sending their own messages.

I.4 Network security and privacy

As described above (I.3) efficient use of network resources, particularly for data frames that require other than best effort delivery, depends on the bridges in the network being able to identify end stations and (for some applications) service characteristics (priority, bandwidth and delay) required by their network applications. Where physical access and attachment to the whole or part of a bridged network is restricted to authorized personnel, confidentiality protection can be limited to that provided by higher layer protocols, notably TLS or IPsec. This leaves all the identifying information specified in IEEE 802.1 standards exposed to an adversary that does gain access to the network media.

The end stations and bridges in bridged networks are typically connected by IEEE Std 802.3 Ethernet links. MACsec (IEEE Std 802.1AE) can be used to provide both confidentiality and integrity protection hop by hop, leaving (in the most common configuration) just the MAC source and destination addresses, frame length, and frame transmission timing visible to an adversary with access to the network media. MACsec adds fields to each frame, but an adversary can recover the original frame length. MACsec operation can affect frame timing, but implementations suitable for use in time-sensitive networks impose a small fixed delay so as not to degrade the operation of IEEE Std 802.1AS time synchronization or IEEE Std 802.1Q timing gates supporting traffic shaping and bandwidth allocation. The frame to frame timing relationships that an adversary might observe remain unaltered. Where MACsec is used with Ethernet frame preemption and in-order delivery of preemptable and (separately) of preempting frames is enforced, an observer can distinguish these two classes of frames. Privacy considerations particular to IEEE Std 802.1X (Port-Based Network Access Control) support of MACsec are described below (I.6.2).

Unlike IEEE Std 802.11 operation in which a mobile end station participates in observable protocol to discover and select a suitable service it is rare for an end station to be connected to an Ethernet link that does not provide the expected service. Authentication, authorization, and confidentiality protection of subsequent data frames, if required, typically occurs before additional end station information is disclosed. For exceptions see I.6.2, I.6.3.

I.5 Privacy exposures

A personal device can be identified explicitly by a single frame field, notably by using a universal MAC address as the source address of transmitted frames.

A station can use a locally assigned MAC address, chosen randomly from the entire local address space or from a subset large enough to yield a sufficiently low probability of address assignment collision, or explicitly assigned by a higher layer protocol. However once a local MAC address has been assigned to a station and is being used to support higher layer protocols (such as IP), to restrict data frames to the path to that station, and to reserve resources in bridges along the path, any further MAC address change can be expected to interrupt or degrade the MAC Service. Moreover the disappearance of one address coupled with rapid appearance of another facilitates correlation of the two addresses and cannot be expected to reduce an adversary’s ability to infer information from the frame fields and other characteristics of persistent flows.

Where an individual frame field does not directly identify a personal device, either persistently as in the case of a universal MAC address or temporarily while the device is continuously active, an adversary can correlate those frame fields and other frame characteristics to identify (to an acceptable probability) the frames and frame flows associated with a single device and even to ascribe a permanent identity to that

device or the particular network applications and activities supported by the device. Such a correlation is called a ‘fingerprint’, and the process of obtaining it ‘fingerprinting’. Fingerprinting does not necessarily require a detailed understanding of the protocols used by a device, but can use general correlation and machine learning techniques to find any persistent pattern in the behavior of a device. Indeed a fingerprint can use device characteristics, such as the persistent scheduling of a transmission by one activity immediately after transmission for another activity, that do not appear in protocol specifications. In the absence of information that all the personal devices of a given type in widespread use consistently use the same network applications in the same way (and consequently exhibit indistinguishable network behavior) it has to be assumed that devices and activities can be distinguished by a sufficiently interested adversary.

The pattern of frame sizes transmitted and received by a personal device can fingerprint application activity and reveal details of that activity. The Ethernet MAC does, however, impose a minimum frame size, and higher layer protocols include fields that allow them to determine the applicable data length. To support Ethernet bridging of frames to and from media without the minimum size requirement, MACsec can encode the short length of those frames, but short frames that have been padded prior to being protected with MACsec will appear to be of uniform length, thus depriving an adversary of the opportunity of fingerprinting application types using the small frame sizes that can be used in initial capability advertisement.

NOTE 1—Frame size patterns have been used to identify banking applications for specific financial institutions, approximate account balances, and whether money is being added to or removed from the account.

Static personal devices, e.g. desktop computers and home routers, typically connect to bridged network using an individual wired IEEE Std 802.3 Ethernet connection. An adversary that can gain access to that wired connection has usually already identified (knowledge of home occupancy, etc.) the person or people associated with such a device and there is no question of tracking device movement. However the pattern of device activity (e.g. turning on security cameras when there is nobody at home) can reveal important personal location information.

NOTE 2—This annex does not detail privacy exposures resulting from media access control method operation, but notes that they can exists. For example, PoE (Power over Ethernet) use can reveal the identity and software version of some consumer electronics devices even when the adversary is restricted to observing the neighboring electromagnetic field.

Bridged networks are typically intraconnected with Ethernet links. Where these are wholly on private premises, access by an adversary can be prevented or at least made so difficult and expensive as to limit the targets to previously identified persons. Where personal device traffic to and from those private premises passes through an IP router, the privacy considerations are those applicable to the use of IP.

NOTE 3—At the time of preparation of this annex, discussion of the extension of TSN capabilities beyond the scope of bridged networks to the use of IP under the heading of ‘DetNet’ (deterministic networks) was still at an early stage. The privacy impacts of explicit flow identification and resource allocation described in this annex can be expected to apply.

IEEE Std 802.11, non-standard wireless connectivity, and in-home electrical power wiring can also be used to connect devices to personal bridged networks and to connect bridges within those networks. Where IEEE Std 802.11 is used to connect to an access point (AP) operating as an IP router, the security considerations applicable to 802.11 and IP apply. Where non-standard wireless connectivity and electrical power wiring are used, an adversary located sufficiently close as to be able to intercept the wireless signal or access power wiring outside possibly secured premises can be assumed to have access to MAC address, frame size, and frame timing information at a minimum with the further possibility of access to all the resource allocation and flow identification information conveyed. Frames with specific group and individual MAC addresses can be filtered by bridges in the network and do not necessarily traverse those links.

An adversary with management access to bridges in the network will have access to resource allocation and flow identification information, but not (at least with standardized objects) the sizes of specific frames and their transmission timing. Such adversaries can include organizations that have a business relationship with the targeted person and are considered trustworthy by that person.

I.6 Standard specific considerations

This clause summarizes particular ways in which each of the bridged network related standards can, when supporting personal devices, expose information that can be used to fingerprint the device's identity or use of network applications. Unless otherwise stated the general considerations described above (I.3, I.4, I.5) also apply to the use of each standard. The brief summary of each standard's capabilities is intended to provide the context for privacy considerations, and is not a substitute for the text of each referenced standard.

I.6.1 IEEE Std 802.1Q Bridges and bridged networks

The general considerations described above (I.3, I.4, I.5) all apply to the use of IEEE Std 802.1Q.

I.6.2 IEEE Std 802.1X Port-Based Network Access Control

IEEE Std 802.1X specifies a general method regulating access to a network, both by systems that are the source and destination of frame carried by the network and by relay systems that are to be connected to multiple other systems in the network and that forward frames between those connections. In both cases each of the system's ports either participates in a mutual authentication exchange with the neighboring system or proves the success of past authentication and authorization to access the network. This clause discusses potential privacy exposures arising from the use of the media-independent capabilities of IEEE Std 802.1X with Ethernet, for privacy considerations related to the use of IEEE 802.11 connections to or within bridged networks see IEEE Std 802.11.

Extensible Authentication Protocol (EAP, IETF RFC 3748) messages are encapsulated in EAP over LANs (EAPOL) PDUs so they can be sent between a Supplicant port (also referred to as a Peer in IETF EAP RFCs), that wishes to gain access to the network, and a Authenticator port, on a system that provides network access. EAP is an authentication framework, not a specific authentication mechanism, and more than 40 specific authentication methods have been defined. An Authenticator is typically supported by an Authentication Server (AS) that executes the particular method or sequence of methods selected. The authentication credentials supported by different methods can differ, as can the degree to which they expose the identity claimed by a Supplicant. EAP messages between the Authenticator and the Authentication Server are typically encapsulated in the RADIUS (IETF RFC 3579) or Diameter (IETF RFC 4072) protocols. IEEE Std 802.1X-2010 mandates the use of mutual authentication methods, and requires support for EAP-TLS (IETF RFC 5216) if integration with the use of IEEE Std 802.1AR is claimed.

Following EAP authentication, RADIUS or Diameter server can provide the Authenticator with attributes that include access controls appropriate to the authorization accorded to the Supplicant and information that supports subsequent reattachment of a device to the network without repetition of the full authentication exchange and authorization process. These attributes can include persistent identifiers, e.g. the EAP-Key-Name (the IEEE 802.1X secure Connectivity Association Key Name, CKN) and Network-Id-Name (2.2 and 2.7 of IETF RFC 7268). Privacy considerations relating to communication between the Authenticator, the Authentication Server, a RADIUS or Diameter Server, and any Online Certificate Status Protocol (OCSP) Server are described in the relevant IETF RFCs.

If data transmission, following successful authentication and authorization, between the Supplicant and Authenticator ports is protected by MACsec, the MACsec Key Agreement protocol (MKA) is used to distribute the succession of Secure Association Keys (SAKs) used to provide confidentiality and integrity protection. MKA uses keys derived from a secure Connectivity Association Key (CAK) and the CKN to integrity protect MKPDUs and to confidentiality and integrity protect (using AES Key Wrap) distributed SAKs. The contents of MKPDUs (other than distributed keys) are not confidentiality protected to support network monitoring and debugging without needing to share the CAK or derived keys. The CAK and CKN

can be derived from an EAP authentication or can be pre-shared by other means, including local device management. The initial octets of each MKPDU contain the CKN, so a peer MACsec capable system knows which (if any) of its key to use to verify that the MKPDU has been transmitted by a previously authenticated system. A device can be configured to attempt, or require, EAP authentication each time it is connected to the network, thus obtaining a fresh CAK and CKN. Shared service infrastructure devices typically need to be capable of restoring connectivity to their neighbours without re-authentication, since neither they or their neighbors are guaranteed to have connectivity to an Authentication Server or other supporting services.

EAPOL frames, and integrity protected MKPDUs which are carried in EAPOL frames, can convey network announcements (Clause 10 of IEEE Std 802.1X-2010). These can be used by personal devices, but are expected to be transmitted by shared service devices.

I.6.3 IEEE Std 802.1AB Station and Media Access Control Connectivity Discovery

The Link Layer Discovery Protocol (LLDP) allows a station to advertise, to others attached to the same LAN, the station's management address and major capabilities. The receiving stations allow management access to received LLDP information to support network topology discovery and configuration checking. The point of LLDP would be lost if the advertised attributes were to be temporary or unavailable to intended recipients. Standard attributes include a system name and description. The range of attributes has been extended by other standards and organizations such as equipment suppliers.

LLDP is a one way protocol: it does not contain mechanisms for soliciting or confirming receipt of information. The destination address of each LLDPDU is usually one of the reserved group addresses specified in IEEE Std 802.1Q and filtered by bridges to limit the scope of its propagation through the network. This filtering allows a management application to use the information received by end stations and bridges in the network to build a map of the network topology. The filtering also restricts exposure of any station's advertised attributes to adversaries that have access to the individual LANs traversed by the LLDPDUs that station transmits, or that have management access to their recipients or to the management application. IEEE Std 802.1AB-2016 mandates support for the Nearest Bridge group address (01-80-C2-00-00-0E, also referred to as the Individual LAN Scope group address). This address is filtered by all bridges.

Where port access is controlled by IEEE Std 802.1X, IEEE Std 802.1AB mandates Controlled Port support for LLDP exchanges, thus providing confidentiality (on the LAN) if MACsec is used. Unprotected transmission using the Uncontrolled Port is permitted.

I.6.4 IEEE Std 802.1AE MAC Security

The exposure of personal information, including information that can contribute to fingerprinting a device or activity, conveyed in frames that are confidentiality protected by MAC Security (MACsec) can be reduced as described above (I.4). The potential exposure of personal device information by the supporting IEEE Std 802.1X MACsec Key Agreement protocol (MKA) is discussed in I.6.2.

MACsec protects communication between neighboring systems, but the scope of that protection depends on what each system considers to be a potential neighbor. By default frames conveyed by the IEEE Std 802.1X Port Access Control Protocol (PACP) that encapsulates the Extensible Authentication Protocol (EAP, IETF RFC 3748) are transmitted to the Nearest non-TPMR Bridge group address (also referred to as the IEEE Std 802.1X PAE address), so any intervening TPMR cannot access confidentiality protected frame fields (see I.4). However MACsec can also be used to secure a point-to-point connection across a Provider Bridge Network exposing any priority information required by PBN systems to provide the desired class of service, and to secure connectivity where the PBN uses VLAN tag information to select a provider service

instance (15.4 and 15.5 of IEEE Std 802.1AE-2018). Where a Provider Backbone Bridge (PBB) is used, the source MAC address of the originator of the frame is encapsulated and confidentiality protected. A PBB is not, itself, likely to be a personal device.

I.6.5 IEEE Std 802.1AR Secure Device Identity

IEEE Std 802.1AR specifies Secure Device Identifiers (DevIDs) for use with IEEE Std 802.1X and other industry authentication, provisioning, and authorization protocols. Privacy consideration for use of DevIDs are discussed in 6.5 of IEEE Std 802.1AR-2018.

I.6.6 IEEE Std 802.1AS Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks

The generalized precision time protocol (gPTP), state machines, and algorithms specified in IEEE Std 802.1AS support time-sensitive applications such as audio, video, and time-sensitive control, by maintaining synchronized time across packet networks, including bridged networks, comprising interconnected time-aware systems. Each time-aware system exchanges messages with its immediate neighbor to measure the link propagation delay experienced by packets forwarded by that neighbor. Time-aware end stations receive time information, either directly or indirectly via one or more time-aware relay systems, from a grandmaster that is the source of time information in a network domain. Each system adjusts the time information received to account for the link propagation delay, and in the case of time-aware relays for the residence time of the information in the relay prior to forwarding. The current grandmaster, and the port used to receive information from that grandmaster, is selected by a best master clock algorithm (BMCA) that constructs a time-synchronization spanning tree throughout the network domain with a spanning tree priority vector that allows each time-aware system to select its best port for receiving (and in the case of a time-aware relay, as the basis for forwarding) timing information.

Each time-aware system port that supports gPTP is identified by a sourcePortIdentity, comprising a clockIdentity and a portNumber. The clockIdentity identifies the clock being used by a specific time-aware bridge or end station for a particular instance of distributed time and is constructed using an NUI-48 or NUI-64 (see IEEE Std 802c): i.e. while it is an identifier and not a protocol address it is constructed in the same way as MAC address, is intended to be unique within a network, and it is possible to tell by examining one of the bits derived from the NUI in the construction (the bit corresponding to the U/L bit when the an NUI is used as a MAC Address) whether the clockIdentity is intended to be locally or globally unique.

The media-independent specification of gPTP is supported by media-dependent procedures. Neighboring time-aware systems connected by full-duplex point-to-point links, such as those specified by IEEE Std 802.3, use gPTP messages to measure the propagation delay and convey timing information. Each message includes the transmitter’s sourcePortIdentity. If the connection is confidentiality protected by MACsec, this message field will only be visible to the communicating systems.

Neighboring IEEE Std 802.11 stations, whether AP capable or not, do not use gPTP messages to measure propagation delay and convey timing. They use the IEEE 802.11 MAC Layer Management Entity (MLME) which generates, timestamps, and consumes measurement frames to provide timing information.

NOTE 1—For privacy considerations related to the IEEE 802.11 MLME see IEEE Std 802.11.

The BMCA spanning tree conveys a trace of each port’s sourcePortIdentity on the best path (for timing distribution) from each potential grandmaster. A personal device attached to the network is thus aware of, and receives a permanent identifier for each system that is part of, that path. The BMCA protocol does not propagate path information in the reverse direction (i.e. towards the grandmaster root of a timing tree): the sourcePortIdentity of a personal device that has a single port attached to the network is only conveyed to its immediate neighbor.

NOTE 2—While use of the redundant grandmasters and the BMCA allows the precision timing service provided by IEEE Std 802.1AS to be resilient in the face of system and link failures, it is highly desirable that the network remain stable and the standard provide priority values for grandmaster selection and timing path selection that discriminate against devices that are not permanently part of the network and powered on. A personal device, and particularly a mobile personal device, is therefore unlikely to find itself in the position of propagating BMCA path trace information including the sourcePortIdentity of one of its ports, even if it has more than one port.

Time-aware stations connected by media for which gPTP is supported by media-independent procedures send Signaling messages (10.4 of IEEE Std 802.1AS-2011) that signal the stations ability to participate in the protocol together with station dependent parameters that control aspects of protocol operation (e.g. message interval request). For stations connected by IEEE Std 802.11 media this capability is provided by the IEEE 802.11 MLME.

I.6.7 IEEE Std 802.1AX Link Aggregation

Link Aggregation allows parallel point-to-point links to be aggregated to form a Link Aggregation Group (LAG) that is treated as a single link. A bridge or end station port generally distributes frames amongst the links so as to preserve frame ordering within flows (see I.3 above). The distribution algorithm and parameters for its use can be specified by using Conversation-sensitive Collection and Distribution (CSCD). A further capability Distributed Resilient Network Interface (DRNI), that provides system level redundancy by allowing two cooperating systems to mimic the behavior of a single system terminating a LAG, is unlikely to be used by personal devices.

The addition and removal of links to and from a LAG is facilitated by the operation of the Link Aggregation Control Protocol (LACP) in each of systems they connect. To ensure that the candidate links for a given LAG do connect the same pair of systems, LACP exchanges a System Identifier that is a combination of System Priority and System MAC Address. This System MAC Address needs to be unique amongst any set system capable of aggregating links with each other, but does not have to be globally unique and is not necessarily (except for any conditions imposed to avoid the profligate assignment of unique identifiers) the address used as a source MAC address by transmitted frames originating from the system. Other LACP parameters, because of potential system to system differences, can contribute to system fingerprinting though not in such a clear way. LACPDUs are transmitted to a group address selected to limit their propagation within the network, typically the Nearest non-TPMR Bridge group address (01-80-C2-00-00-03).

When MACsec is used to protect communication between neighboring system, the MAC Security Entity is instantiated in the interface stack associated with each of the individual aggregatable links (see 11.5 of IEEE Std 802.1AE-2018) and thus can confidentiality protect both the conversations carried over those links and operation of LACP.

I.6.8 IEEE Std 802.1BA Audio Video Bridging (AVB) Systems

IEEE Std 802.1BA specifies the selection of specific features and options from IEEE Std 802.1Q, IEEE Std 802.1AS, and LAN MAC/PHY standards that facilitate manufacture of AVB-capable components. A person not skilled in networking can use those components to build networks that provide working audio and video services. This standard does not introduce additional privacy considerations beyond those inherent in the referenced standards.

I.6.9 IEEE Std 802.1BR Virtual Bridged Local Area Networks—Bridge Port Extension

IEEE Std 802.1BR specifies a method for increasing the effective geographical extent of the control parameters of a single bridge by supporting multiple instances of the Enhanced Internal Sublayer Service, each associated with a single bridge port, over a single LAN connected to an External Bridge Port Extender that can support one or more ports attached to LANs, each serving a single end station, and zero or more ports connected to further External Bridge Port Extenders. Bridge Port Extenders also support frame replication for multicast. While Bridge Port Extenders extend the effective extent of a single bridge they do require port extender specific configuration to support time-sensitive network flows.

Bridge Port Extenders were standardized to meet data center bridging requirements and are not expected to be personal devices or to provide services directly to personal devices.

I.6.10 IEEE Std 802.1CB Frame Replication and Elimination for Reliability

Frame Replication and Elimination for Reliability (FRER) increases the probability that any given packet will be delivered by replicating each of an identifiable sequence of packets, transmitting the replicates on disjoint network paths, and eliminating duplicates where those paths meet. The sequence of duplication, duplicate transmission, and elimination, can be repeated between transmission by the original source of the packets and reception by the eventual destination(s). Resources can be reserved on each of the paths that support duplicate transmission so that TSN delivery objectives (timeliness and extremely low loss) can be met even if LANs or relay systems fail (on all but one of the potential paths between the original source and a destination).

FRER requires, at a minimum, the addition of a sequence number to each packet. IEEE Std 802.1CB specifies a redundancy tag (R-TAG) that adds just that sequence number, and also allows use of the High-availability Seamless Redundancy (HSR) sequence tag or the Parallel Redundancy Protocol (PRP) sequence trailer both specified by IEC 62439-3:2016 (7.8, 7.9, and 7.10 of IEEE Std 802.1CB-2016). None of these contribute significantly to an adversary’s ability to assign each packet to a stream or flow as is necessary, using the contents of other frame fields, by bridges and end stations supporting resource allocation and FRER. IEEE Std 802.1CB does not specify positioning of its processing relative to that carried out by the IEEE Std 802.1AE MAC Security Entity in interface stacks, but the usual considerations place the latter closer to the PHY. MACsec confidentiality protection, where used, will apply to the FRER tags and trailer just as it would to the stream and flow identifying frame fields.

While IEEE Std 802.1CB addresses the requirements addressing from industrial networks it can be used to support personal devices.

I.6.11 IEEE Std 802.1CM Time-Sensitive Networking for Fronthaul

IEEE Std 802.1CM specifies the selection of specific features and options from IEEE Std 802.1Q, IEEE Std 802.1AC, IEEE Std 802.3, IEEE Std 1588, ITU-T G.8275.1, ITU-T G.8261, ITU-T G.8262, and ITU-T G.8264, to enable the transport of time-sensitive fronthaul streams in Ethernet bridged networks. This standard does not introduce additional privacy considerations beyond those inherent in the referenced standards.