MACsec Privacy
May 18, 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.

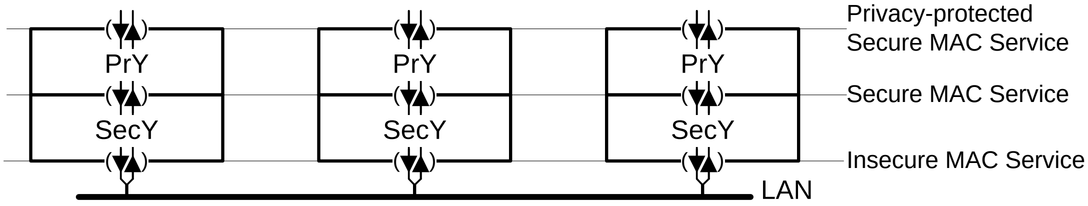


Figure 17-1—Privacy-protected communication between three stations

[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.>>

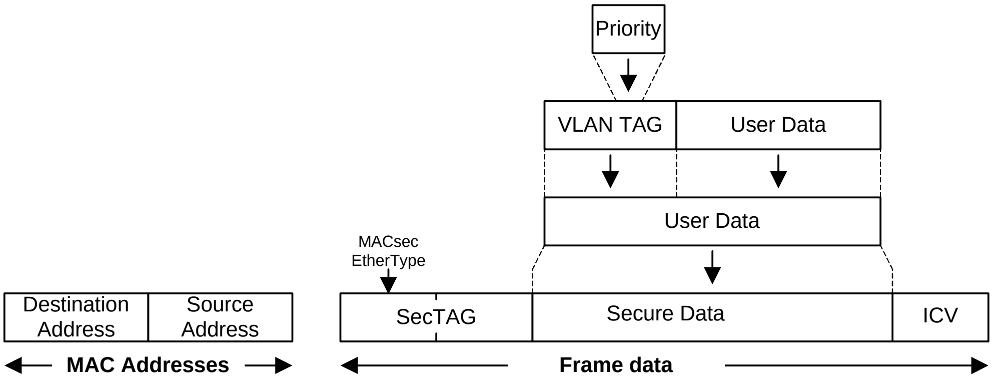


Figure 17-2—A privacy-protected frame

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 mentionned, 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-protecting 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 representation and encoding of protocol fields
b) Specifies the major components of each MPPDU, and the fields they comprise
c) Reviews the purpose of each field, and the functionality provided
d) Specifies validation of the MPPDU on reception
e) Documents the allocation of the MAC Privacy-protection EtherType that identifies MPPDUs

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.

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

19.2 Major components

Each MPDU comprises

a) A Security TAG (SecTAG) (19.3)
b) Secure Data (19.10)
c) An Integrity Check Value (ICV) (19.11)

Each of these components comprises an integral number of octets and is encoded in successive octets of the MPDU as illustrated in Figure 19-1.

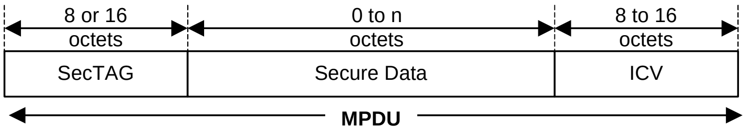


Figure 19-1—MPDU components

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

19.3 Security TAG

The Security TAG (SecTAG) is identified by the MACsec EtherType (19.4), and conveys the

a) TAG Control Information (TCI, 19.5)
b) Association Number (AN, 19.6)
c) Short Length (SL, 19.7)
d) Packet Number (PN, 19.8)

e) Optionally encoded Secure Channel Identifier (SCI, 19.9).

The format of the SecTAG is illustrated in Figure 19-2.

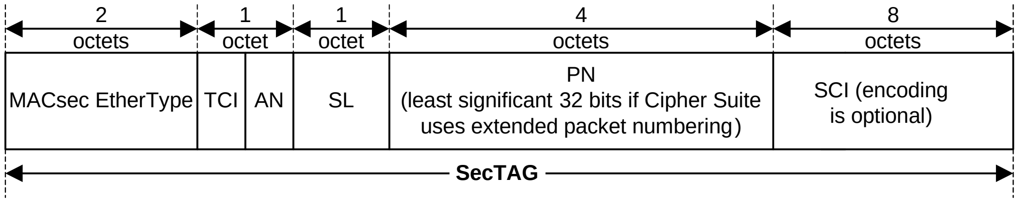


Figure 19-2—SecTAG format

19.4 MACsec EtherType

The MACsec EtherType (Table 19-1) comprises octet 1 and octet 2 of the SecTAG. It is included to allow

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

Table 19-1—MACsec EtherType allocation

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

The encoding of the MACsec EtherType in the MPDU is illustrated in Figure 19-3.

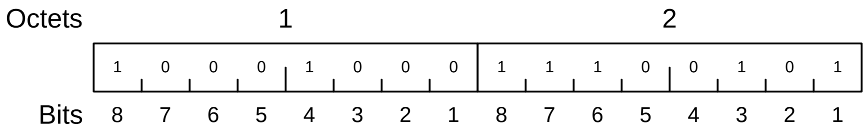


Figure 19-3—MACsec EtherType encoding

19.5 TAG Control Information (TCI)

The TCI field comprises bits 8 through 3 of octet 3 (Figure 19-4) of the SecTAG. These bits facilitate

a) Version numbering of the MACsec protocol without changing the MACsec EtherType
b) Optional use of the MAC Source Address parameter to convey the SCI
c) Optional inclusion of an explicitly encoded SCI (7.1.2, Figure 7-7)

d) Use of the EPON (Clause 12) Single Copy Broadcast capability, without requiring an explicit SCI to distinguish the SCB Secure Channel

e) Extraction of the User Data from MPDUs by systems that do not possess the SAK (8.1.2, 8.1.4) when confidentiality is not being provided

f) Determination of whether confidentiality or integrity alone are in use

The encoding of the MACsec TCI in the MPDU is illustrated in Figure 19-4.

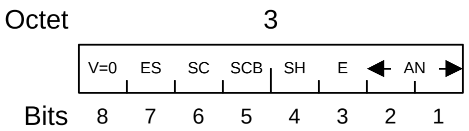


Figure 19-4—MACsec TCI and AN Encoding

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

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

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

If an SCI (19.9, 7.1.2) is explicitly encoded in the SecTAG, bit 6 (the SC bit) of the TCI shall be set. The SC bit shall be clear if an SCI is not present in the SecTAG.

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

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

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

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

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

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

19.6 Association Number (AN)

The AN is encoded as an integer in bits 1 and 2 of octet 3 of the SecTAG (Figure 19-4) and identifies up to four different SAs within the context of an SC.

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

19.7 Short Length (SL)

SL is an integer encoded in bits 1 through 6 of octet 4 of the SecTAG and is set to the number of octets in the Secure Data (19.10) field, i.e., the number of octets between the last octet of the SecTAG and the first octet of the ICV, if that number is less than 48. Otherwise, SL is set to zero. If the number is zero then the frame is deemed not to have been short. The Secure Data field always comprises at least one octet.

Bits 7 and 8 of octet 4 shall be zero.

19.8 Packet Number (PN)

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

a) Provide a unique IV PDU for all MPDUs transmitted using the same SA

b) Support replay protection

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

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

19.9 Secure Channel Identifier (SCI)

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

a) Identification of the SA where the CA comprises three or more SCs

b) Network management identification of the SecY that has transmitted the frame

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

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

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

An explicitly encoded SCI field in the SecTAG is not required on point-to-point links, which are identified by the operPointToPointMAC status parameter of the service provider, if the transmitting SecY uses only one transmit SC. In that case, the secure association created by the SecY for the peer SecYs, together with the direction of transmission of the secured MPDU, can be used to identify the transmitting SecY. Therefore an explicitly encoded SCI is unnecessary. Although the SCI does not have to be repeated in each frame when only two SecYs participate in a CA (see Clause 8, Clause 19, and Clause 10), the SCI (for Cipher Suites using a 32-bit PN) or the SSCI (for Cipher Suites using a 64-bit PN) still forms part of the cryptographic computation.

19.10 Secure Data

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

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

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

19.11 Integrity Check Value (ICV)

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

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

19.12 PDU validation

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

a) It comprises at least 17 octets

b) Octets 1 and 2 compose the MACsec EtherType

c) The V bit in the TCI is clear

d) If the ES or the SCB bit in the TCI is set, then the SC bit is clear

e) Bits 7 and 8 of octet 4 of the SecTAG are clear

f) If the C and SC bits in the TCI are clear, the MPDU comprises 24 octets plus the number of octets indicated by the SL field if that is non-zero and at least 72 octets otherwise

g) If the C bit is clear and the SC bit set, then the MPDU comprises 32 octets plus the number of octets indicated by the SL field if that is non-zero and at least 80 octets otherwise

h) If the C bit is set and the SC bit clear, then the MPDU comprises 8 octets plus the minimum length of the ICV as determined by the Cipher Suite in use at the receiving SecY, plus the number of octets indicated by the SL field if that is non-zero and at least 48 additional octets otherwise

i) If the C and SC bits are both set, the frame comprises at least 16 octets plus the minimum length of the ICV as determined by the Cipher Suite in use at the receiving SecY, plus the number of octets indicated by the SL field if that is non-zero and at least 48 additional octets otherwise

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 SecYs (20.8).

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

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).

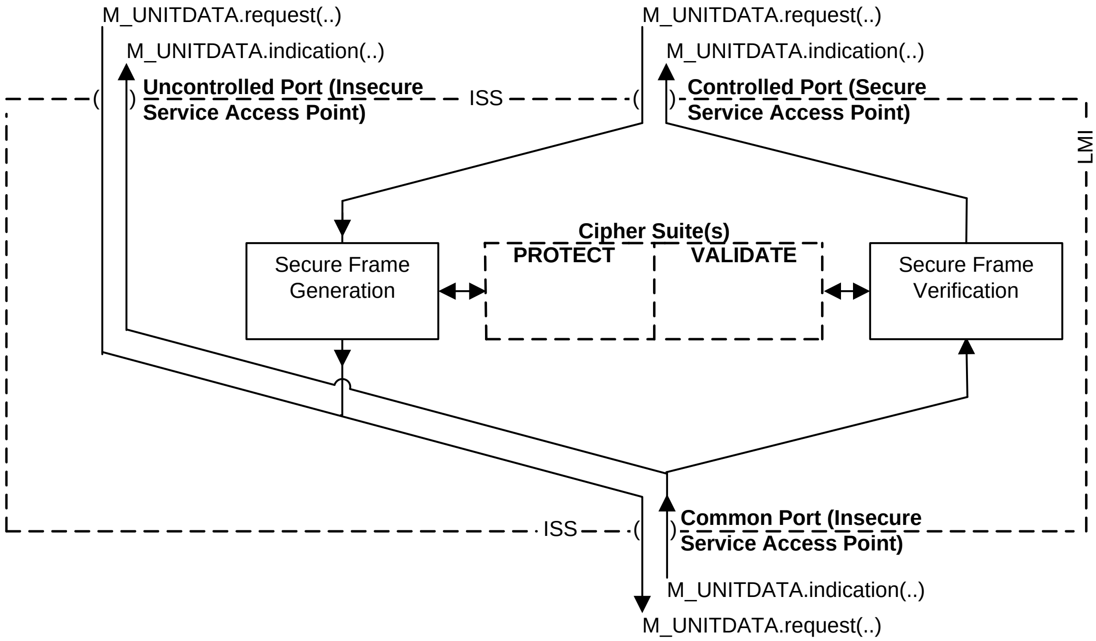

M_UNITDATA.request(..)
M_UNITDATA.indication(..)
Uncontrolled Port (Insecure Service Access Point)
ISS
M_UNITDATA.request(..)
M_UNITDATA.indication(..)
Controlled Port (Secure Service Access Point)
LMI
Cipher Suite(s)
PROTECT
VALIDATE
Secure Frame Generation
Secure Frame Verification
ISS
Common Port (Insecure Service Access Point)
M_UNITDATA.indication(..)
M_UNITDATA.request(..)

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.   .

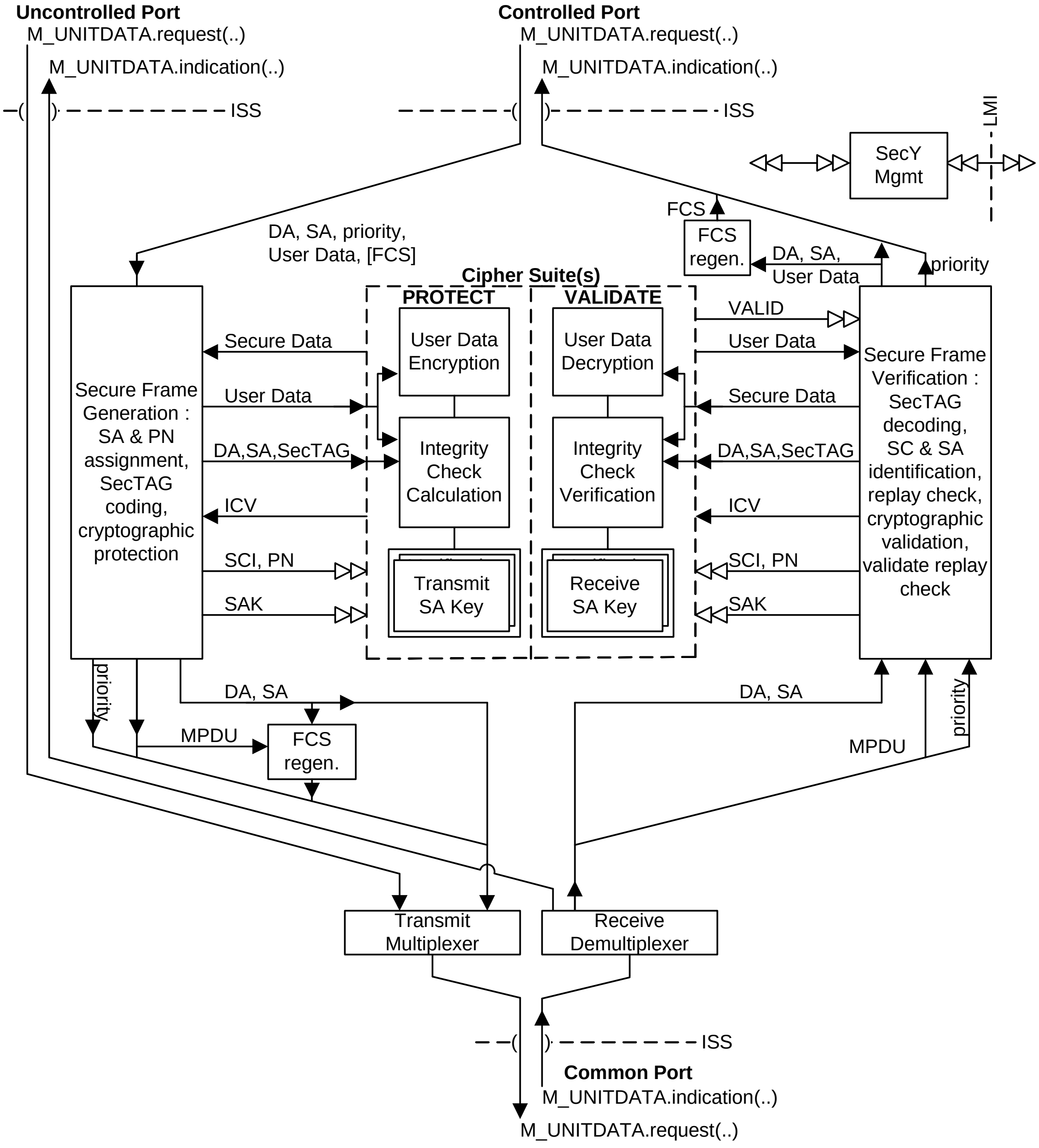


Figure 20-2—SecY architecture and operation

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)

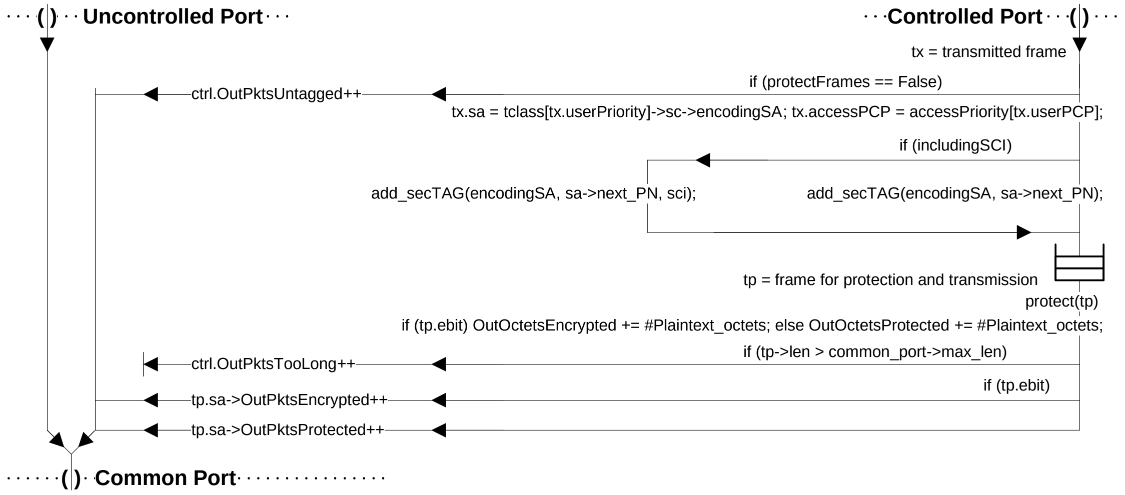
d) Provides the protection function (14.1, 20.5.4) of the Current Cipher Suite with
   1) The SA Key (SAK)
   2) The SCI for the SC used by the SecY to transmit
   3) The PN
   4) The SecTAG
   5) The sequence of octets that compose the User Data

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.


Uncontrolled Port
Controlled Port
tx = transmitted frame
if (protectFrames == False)
ctrl.OutPktsUntagged++
tx.sa = tclass[tx.userPriority]->sc->encodingSA; tx.accessPCP = accessPriority[tx.userPCP];
if (includingSCI)
add_secTAG(encodingSA, sa->next_PN, sci);
add_secTAG(encodingSA, sa->next_PN);
tp = frame for protection and transmission
protect(tp)
if (tp.ebit) OutOctetsEncrypted += #Plaintext_octets; else OutOctetsProtected += #Plaintext_octets;
if (tp->len > common_port->max_len)
ctrl.OutPktsTooLong++
if (tp.ebit)
tp.sa->OutPktsEncrypted++
tp.sa->OutPktsProtected++
Common Port

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

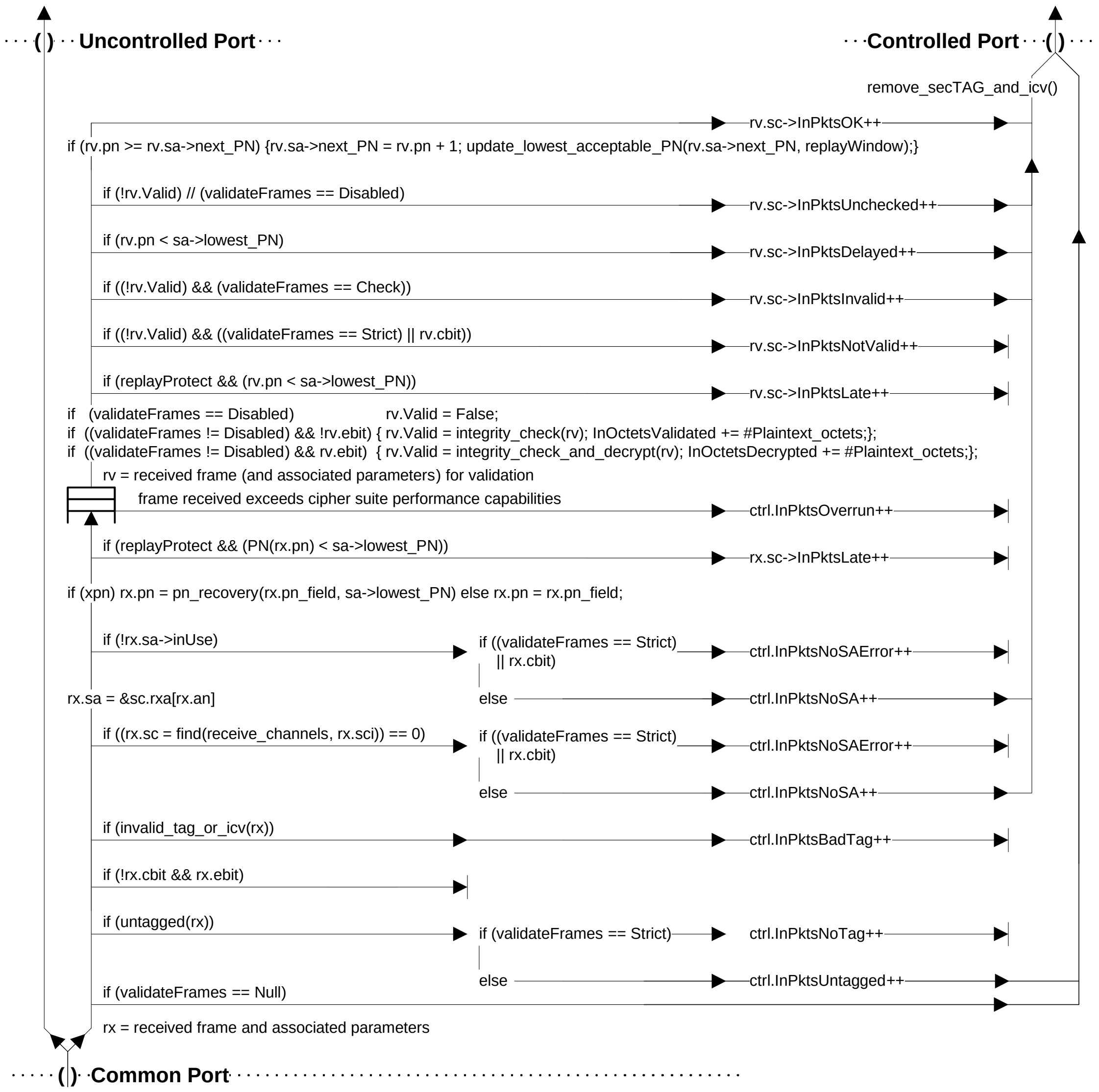
If the MPDU composed of the concatenated octets of the SecTAG, Secure Data, and ICV exceeds the size of the MSDU supported by the Common Port, the frame is discarded and a counter incremented. Details of the discarded frame may be recorded to assist network management resolution of the problem. Otherwise,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


Uncontrolled Port
Controlled Port
remove_secTAG_and_icv()
rv.sc->InPktsOK++
if (rv.pn >= rv.sa->next_PN) {rv.sa->next_PN = rv.pn + 1; update_lowest_acceptable_PN(rv.sa->next_PN, replayWindow);}
if (!rv.Valid) // (validateFrames == Disabled)
rv.sc->InPktsUnchecked++
if (rv.pn < sa->lowest_PN)
rv.sc->InPktsDelayed++
if ((!rv.Valid) && (validateFrames == Check))
rv.sc->InPktsInvalid++
if ((!rv.Valid) && ((validateFrames == Strict) || rv.cbit))
rv.sc->InPktsNotValid++
if (replayProtect && (rv.pn < sa->lowest_PN))
rv.sc->InPktsLate++
if (validateFrames == Disabled) rv.Valid = False;
if ((validateFrames != Disabled) && !rv.ebit) { rv.Valid = integrity_check(rv); InOctetsValidated += #Plaintext_octets;};
if ((validateFrames != Disabled) && rv.ebit) { rv.Valid = integrity_check_and_decrypt(rv); InOctetsDecrypted += #Plaintext_octets;};
rv = received frame (and associated parameters) for validation
frame received exceeds cipher suite performance capabilities
ctrl.InPktsOverrun++
if (replayProtect && (PN(rx.pn) < sa->lowest_PN))
rx.sc->InPktsLate++
if (xpn) rx.pn = pn_recovery(rx.pn_field, sa->lowest_PN) else rx.pn = rx.pn_field;
if (!rx.sa->inUse)
if ((validateFrames == Strict) || rx.cbit)
ctrl.InPktsNoSAError++
rx.sa = &sc.rxa[rx.an]
else
ctrl.InPktsNoSA++
if ((rx.sc = find(receive_channels, rx.sci)) == 0)
if ((validateFrames == Strict) || rx.cbit)
ctrl.InPktsNoSAError++
else
ctrl.InPktsNoSA++
if (invalid_tag_or_icv(rx))
ctrl.InPktsBadTag++
if (!rx.cbit && rx.ebit)
if (untagged(rx))
if (validateFrames == Strict)
ctrl.InPktsNoTag++
else
ctrl.InPktsUntagged++
if (validateFrames == Null)
rx = received frame and associated parameters
Common Port

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

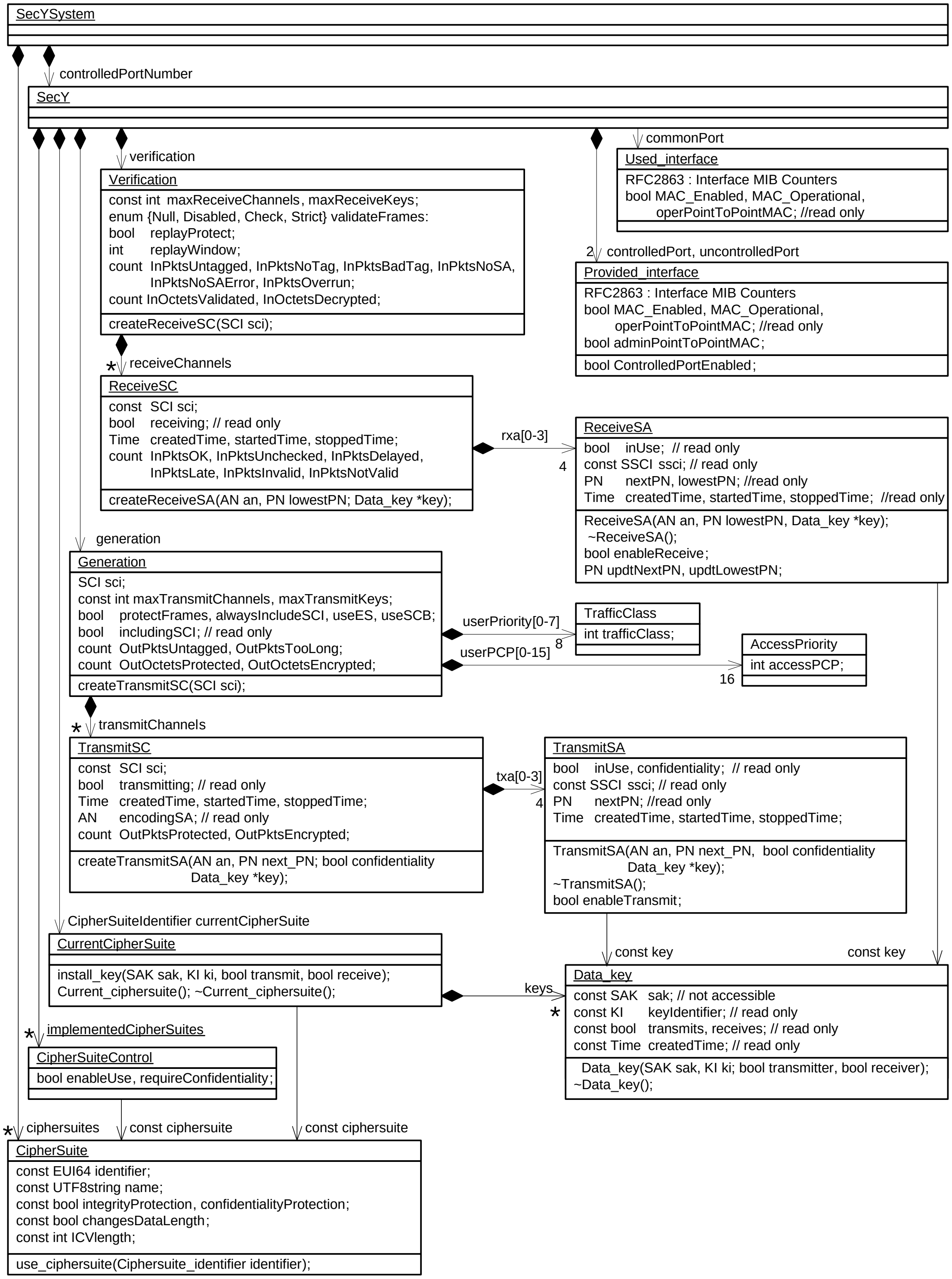

SecYSystem
controlledPortNumber
SecY
verification
commonPort
Used_interface
RFC2863 : Interface MIB Counters
bool MAC_Enabled, MAC_Operational,
operPointToPointMAC; //read only
Verification
const int maxReceiveChannels, maxReceiveKeys;
enum {Null, Disabled, Check, Strict} validateFrames:
bool replayProtect;
int replayWindow;
count InPktsUntagged, InPktsNoTag, InPktsBadTag, InPktsNoSA,
InPktsNoSAError, InPktsOverrun;
count InOctetsValidated, InOctetsDecrypted;
createReceiveSC(SCI sci);
2
controlledPort, uncontrolledPort
Provided_interface
RFC2863 : Interface MIB Counters
bool MAC_Enabled, MAC_Operational,
operPointToPointMAC; //read only
bool adminPointToPointMAC;
bool ControlledPortEnabled;
*
receiveChannels
ReceiveSC
const SCI sci;
bool receiving; // read only
Time createdTime, startedTime, stoppedTime;
count InPktsOK, InPktsUnchecked, InPktsDelayed,
InPktsLate, InPktsInvalid, InPktsNotValid
createReceiveSA(AN an, PN lowestPN; Data_key *key);
rxa[0-3]
4
ReceiveSA
bool inUse; // read only
const SSCI ssci; // read only
PN nextPN, lowestPN; //read only
Time createdTime, startedTime, stoppedTime; //read only
ReceiveSA(AN an, PN lowestPN, Data_key *key);
~ReceiveSA();
bool enableReceive;
PN updtNextPN, updtLowestPN;
generation
Generation
SCI sci;
const int maxTransmitChannels, maxTransmitKeys;
bool protectFrames, alwaysIncludeSCI, useES, useSCB;
bool includingSCI; // read only
count OutPktsUntagged, OutPktsTooLong;
count OutOctetsProtected, OutOctetsEncrypted;
createTransmitSC(SCI sci);
userPriority[0-7]
8
TrafficClass
int trafficClass;
userPCP[0-15]
16
AccessPriority
int accessPCP;
*
transmitChannels
TransmitSC
const SCI sci;
bool transmitting; // read only
Time createdTime, startedTime, stoppedTime;
AN encodingSA; // read only
count OutPktsProtected, OutPktsEncrypted;
createTransmitSA(AN an, PN next_PN; bool confidentiality
Data_key *key);
txa[0-3]
4
TransmitSA
bool inUse, confidentiality; // read only
const SSCI ssci; // read only
PN nextPN; //read only
Time createdTime, startedTime, stoppedTime;
TransmitSA(AN an, PN next_PN, bool confidentiality
Data_key *key);
~TransmitSA();
bool enableTransmit;
CipherSuiteIdentifier currentCipherSuite
CurrentCipherSuite
install_key(SAK sak, KI ki, bool transmit, bool receive);
Current_ciphersuite(); ~Current_ciphersuite();
const key
const key
keys
*
Data_key
const SAK sak; // not accessible
const KI keyIdentifier; // read only
const bool transmits, receives; // read only
const Time createdTime; // read only
Data_key(SAK sak, KI ki; bool transmitter, bool receiver);
~Data_key();
*
implementedCipherSuites
CipherSuiteControl
bool enableUse, requireConfidentiality;
*
ciphersuites
const ciphersuite
const ciphersuite
CipherSuite
const EUI64 identifier;
const UTF8string name;
const bool integrityProtection, confidentialityProtection;
const bool changesDataLength;
const int ICVlength;
use_ciphersuite(Ciphersuite_identifier identifier);

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).

Annex B

(informative)

Bibliography

[B1] IEEE Std 802.11™, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[B2] IETF RFC 2279, UTF-8, a Transformation format of ISO 10646, Yergeau, F., January 1998.

[B3] IETF RFC 3410, Introduction and Applicability Statements for Internet-Standard Management Framework, Case, J., Mundy, R., Partain, D., and Stewart, B., December 2002.

[B4] IETF RFC 4303, IP Encapsulating Security Payload (ESP), Kent, S., December 2005.

[B5] IETF RFC 5116, An Interface and Algorithms for Authenticated Encryption, McGrew, D., January 2008.

[B6] Fowler, M., "UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition," Pearson Education Inc., Boston, 2004, ISBN 0-321-19368-7.

[B7] Generation of Deterministic Initialization Vectors (IVs) and Nonces, McGrew, D., October 2013.[9]

[B8] The Galois/Counter Mode of Operation (GCM), David A. McGrew and J. Viega. May 31, 2005.[10]

[B9] MEF Technical Specification 16 (MEF 16), Ethernet Local Management Interface (E-LMI)[11].

[B10] The Security and Performance of the Galois/Counter Mode (GCM) of Operation. D. McGrew and J. Viega. Proceedings of INDOCRYPT '04, Springer-Verlag, 2004.[12]

[B11] The XPN recovery algorithm, Mick Seaman. June 2012.[13]

[9] Available at https://tools.ietf.org/html/draft-mcgrew-iv-gen-03

[10] A prior revision of this document was the normative reference for GCM in IEEE Std 802.1AE-2006, but has been superseded by NIST SP 800-38D for that purpose. It does contain additional background information, and can be downloaded from https://pdfs.semanticscholar.org/114a/4222c53f1a6879f1a77f1bae2fc0f8f55348.pdf

[11] MEF technical specifications are available from the Metro Ethernet Forum (https://www.mef.net).

[12] Available from the IACR Cryptology ePrint Archive: Report 2004/193, https://eprint.iacr.org/2004/193

[13] Available at https://www.ieee802.org/1/files/public/docs2012/aebw-seaman-xpn-recovery-0612-v02.pdf

Annex H

(normative)

PICS Proforma for a Privacy-protecting Entity (PrY)[14]

<<At present this is just a copy of the basic MAC Security (SecY) PICS as a basis for editing.>

H.1 Introduction

The supplier of a protocol implementation which is claimed to conform to this standard shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

a) By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight;

b) By the supplier and acquirer—or potential acquirer—of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

c) By the user—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSs);

d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

H.2 Abbreviations and special symbols

H.2.1 Status symbols

M	mandatory
O	optional
O.n	optional, but support of at least one of the group of options labelled by the same numeral n is required
X	prohibited
pred:	conditional-item symbol, including predicate identification: see H.3.4
¬	logical negation, applied to a conditional item's predicate

H.2.2 General abbreviations

N/A	not applicable
PICS	Protocol Implementation Conformance Statement

[14] *Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

H.3 Instructions for completing the PICS proforma

H.3.1 General structure of the PICS proforma

The first part of the PICS proforma, implementation identification and protocol summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire, divided into several subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No), or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked.)

Each item is identified by an item reference in the first column. The second column contains the question to be answered; the third column records the status of the item—whether support is mandatory, optional, or conditional; see also H.3.4 below. The fourth column contains the reference or references to the material that specifies the item in the main body of this standard, and the fifth column provides the space for the answers.

A supplier may also provide (or be required to provide) further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labelled *Ai* or *Xi*, respectively, for cross-referencing purposes, where *i* is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format and presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformation Statement for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation’s configuration capabilities, in case that makes for easier and clearer presentation of the information.

H.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but that have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

H.3.3 Exception information

It may occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer will be found in the Support column for this: instead, the supplier shall write the missing answer into the Support column, together with an *Xi* reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this standard.

NOTE—A possible reason for the situation described above is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

H.3.4 Conditional status

H.3.4.1 Conditional items

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply—mandatory or optional—are dependent upon whether or not certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the “Not Applicable” answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form “**pred:** S” where **pred** is a predicate as described in H.3.4.2 below, and S is a status symbol, M or 0.

If the value of the predicate is True (see H.3.4.2), the conditional item is applicable, and its status is indicated by the status symbol following the predicate: the answer column is to be marked in the usual way. If the value of the predicate is False, the “Not Applicable” (N/A) answer is to be marked.

H.3.4.2 Predicates

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: the value of the predicate is True if the item is marked as supported, and is False otherwise;

b) A predicate-name, for a predicate defined as a Boolean expression constructed by combining item-references using the Boolean operator OR: the value of the predicate is True if one or more of the items is marked as supported;

c) A predicate-name, for a predicate defined as a Boolean expression constructed by combining item-references using the boolean operator AND: the value of the predicate is True if all of the items are marked as supported;

d) The logical negation symbol “¬” prefixed to an item-reference or predicate-name: the value of the predicate is True if the value of the predicate formed by omitting the “¬” symbol is False, and vice versa.

Each item whose reference is used in a predicate or predicate definition, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

H.4 PICS proforma for IEEE Std 802.1AE

H.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) of machines and/or operating system names	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirement for full identification. NOTE 2—The terms *Name* and *Version* should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

H.4.2 Protocol summary, IEEE Std 802.1AE

Identification of protocol specification	IEEE Std 802.1AE, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security
Identification of amendments and corrigenda to the PICS proforma which have been completed as part of the PICS	Amd. : Corr. : Amd. : Corr. :
Have any Exception items been required? (See H.3.3: the answer Yes means that the implementation does not conform to IEEE Std 802.1AE.)	No [ ] Yes [ ]

Date of Statement	

H.5 Major capabilities

Item	Feature	Status	References	Support
SAP	Does the implementation of each MAC Security Entity support the Controlled and Uncontrolled Ports, and use a Common Port as specified in Clause 10?	M	5.3(a), Clause 10, H.6	Yes [ ]
STAT	Does the implementation support the MAC status and point-to-point parameters for the Controlled and Uncontrolled Ports as specified in 6.4, 6.5, and 10.7?	M	5.3(b), 6.4, 6.5, 10.7, H.7	Yes [ ]
GEN	Does the implementation process transmit requests from the Controlled Port as required by the specification of Secure Frame Generation (10.5)?	M	5.3(c), 10.5, H.8	Yes [ ]
VER	Does the implementation process receive indications from the Common Port as required by the specification of Secure Frame Verification (10.6), prior to causing receive indications at the Controlled Port?	M	5.3(d), 10.6, H.9	Yes[ ]
FMT	Does the implementation encode and decode MACsec PDUs as specified in Clause 9?	M	5.3(e), Clause 9, H.10	Yes [ ]
SCI	Does the implementation use a 48-bit MAC Address and a 16-bit Port Identifier unique within the scope of that address assignment to identify each transmit SCI, as specified in 8.2.1?	M	5.3(f), 8.2.1	Yes [ ]
PERF	Does the implementation satisfy the performance requirements specified in Table 10-3 and 8.2.2?	M	5.3(g), 10.1, Table 10-3, 8.2.2	Yes [ ]
FCS	Does the implementation introduce an undetected frame error rate greater than that achievable by preserving the original FCS, as required by 10.4?	X	5.3(n), 10.4, 6.10	No [ ]
KAY	Does the implementation support the LMI operations required by the Key Agreement Entity as specified in Clause 10?	M	5.3(h), Clause 10, H.11	Yes [ ]
MGT	Does the implementation provide the management functionality specified in 10.7?	M	5.3(i), 10.7, H.12.1	Yes [ ]
MIB	Does the implementation support access to MACsec parameters by a network management using SNMP v3 and the MIB module specified in Clause 13?	O	5.3(a), Clause 13	Yes [ ] No[ ]
SNMX	Does the implementation support access to MACsec parameters using any version of SNMP prior to v3?	X	5.3(p)	No[ ]
MSC	Does the implementation support more than one receive SC?	O	5.4(b)	Yes [ ] No[ ]
MSAK	Does the implementation support more than two receive SAKs?	O	5.4(c)	Yes [ ] No[ ]
CS	Does the implementation protect and validate MACsec PDUs by using implemented Cipher Suites as specified in 14.1?	M	5.3(j), 14.1	Yes [ ]
CSI	Does the implementation support Integrity Protection using the Default Cipher Suite specified in Clause 14?	M	5.3(k), Clause 14, 14.5	Yes [ ]
CSC	Does the implementation support Confidentiality Protection using the Default Cipher Suite without a Confidentiality Offset as specified in Clause 14?	¬CSO:O CSO:M	5.4(e), Clause 14, 14.5	Yes [ ]
CSO	Does the implementation support Confidentiality Protection using the Default Cipher Suite with a Confidentiality Offset as specified in Clause 14?	O	5.4(f), Clause 14, 14.5,	Yes [ ]

H.5 Major capabilities *(continued)*

Item	Feature	Status	References	Support
CSA	Does the implementation include Cipher Suites that are specified in Clause 14 in addition to the Default Cipher Suite? (This PICS requires the completion of a copy of Table H.13 for each such Cipher Suite implemented.)	O	5.4(g), H.13	Yes [ ] No[ ]
CSX	Does the implementation include any Cipher Suite that is additional to those specified in Clause 14 and does not meet all the criteria specified in 14.2, 14.3, 14.4.1?	X	5.3(o), 14.2, 14.3, 14.4.1	No[ ]
CSV	Does the implementation include Cipher Suites other than those specified in Clause 14, but meeting the criteria specified in 14.2, 14.3, 14.4.1? (This PICS requires the completion of a copy of Table H.14 for each such Cipher Suite implemented.)	O	5.4(i), H.14	Yes [ ] No[ ]
CSR	Does the implementation support a minimum of one receive SC, two receive SAKs, one transmit SC, and one of the two receive SAKs at a time for transmission as specified in 5.3(l), for each Cipher Suite implemented?	M	5.3(l), Clause 14	Yes [ ]
CSS	Does this completed PICS specify the maximum number of receive SCs, receive SAKS, and transmit SCs for each Cipher Suite implemented?	M	5.3(m), H.13, H.14	Yes [ ]
CSRC	What is the maximum number of receive SCs supported by the Default Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSRK	What is the maximum number of receive SAKs supported by the Default Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSTC	What is the maximum number of transmit SCs supported by the Default Cipher Suite implementation? _ _ _ _ _		5.3(m)	
TC	Does the implementation support more than one transmit SC for any Cipher Suite?	O	5.4(d)	Yes [ ] No [ ]
TCT	Is a Traffic Class Table implemented?	**TC**:M	5.4(h), 10.7.17	Yes [ ] N/A[ ]
TCAPT	Is an Access Priority Table implemented?	**TC**:M	5.4(h), 10.7.17,	Yes [ ] N/A[ ]
FULL	Is a claim for full conformance being made for the implementation?	**CSV**:X ¬**CSV**:O	5.3	Yes [ ] No[ ]
VAR	Is a claim for conformance with cipher suite variance being made for the implementation?		5.3	Yes [ ] No[ ]

H.6 Support and use of Service Access Points

Item	Feature	Status	References	Support
SAP-1	Does each transmit request from the Uncontrolled Port result in a single request to the Common Port with the same parameters?	M	10.4	Yes [ ]
SAP-2	Does each receive indication from the Common Port result in a single indication to the Uncontrolled Port with the same parameters if any of the users of the Common Port wishes to receive the indication?	M	10.4	Yes [ ]
SAP-3	Does each transmit request from the Controlled Port result in at most one request to the Common Port?	M	10.4	Yes [ ]
SAP-4	Does each receive indication from the Common Port result in at most one indication to the Controlled Port?	M	10.4	Yes [ ]
SAP-5	Are any transmit requests made to the Common Port that do not correspond to requests made at the Uncontrolled or Controlled Port?	X	10.4	No [ ]
SAP-6	Are any receive indications caused at the Uncontrolled or Controlled Port that do not correspond to indications from the Common Port?	X	10.4	No [ ]
SAP-7	Is the order of requests made at the Common Port unchanged from the order of corresponding requests from the Uncontrolled Port?	M	10.4	Yes [ ]
SAP-8	Is the order of requests made at the Common Port unchanged from the order of corresponding requests from the Controlled Port?	M	10.4	Yes [ ]
SAP-9	Is the order of receive indications caused at the Uncontrolled Port the same as the order of reception from the Common Port?	M	10.4	Yes [ ]
SAP-10	Is each transmit request from the Controlled Port processed in accordance with the specification of the Secure Frame Generation process, prior to discarding the request or making a corresponding request to the Common Port?	M	10.4, 10.5	Yes [ ]
SAP-11	Is each receive indication from the Common Port processed in accordance with the specification of the Secure Frame Verification process prior to causing a possible corresponding indication at the Controlled Port?	M	10.4, 10.6	Yes [ ]

H.7 MAC status and point-to-point parameters

Item	Feature	Status	References	Support
STAT-1	Are the values for MAC_Operational and operPointToPointMAC for the Uncontrolled Port identical to those for the Common Port?	M	6.4, 10.7.2	Yes [ ]
STAT-2	Is MAC_Operational False for the Controlled Port, and frames neither accepted or delivered on the port, if the SA identified by the encodingSA is not available for use and protectFrames is set?	M	6.4, 10.5.1, 7.1	Yes [ ]
STAT-3	Is MAC_Operational False for the Controlled Port and frames neither accepted nor delivered, if the nextPN for the encodingSA is zero or 2^{32}?	M	6.4, 10.5.2	Yes [ ]
STAT-4	Is MAC_Operational True only if MAC_Enabled is True and MAC_Operational for the Common Port is True?	M	6.4, 10.7.4	Yes [ ]
STAT-5	Is the value of operPointToPointMAC for the Controlled Port always as specified in 10.7.4.	M	6.5, 10.7.4	Yes [ ]

H.8 Secure Frame Generation

Item	Feature	Status	References	Support
GEN-1	Does each transmit request from the Controlled Port result in an identical transmit request at the Common Port if the management control protectFrames is False?	M	10.5	Yes [ ]
GEN-2	Does each transmit request at the Common Port resulting from a request at the Common Port convey request parameters, i.e., a frame, protected in accordance with Clause 10.5 if the management control protectFrames is True?	M	10.5	Yes[ ]
GEN-3	Is each protected frame assigned to the SA with AN corresponding to the current value of encodingSA as specified by the KaY?	M	10.5.1	Yes[ ]
GEN-4	Are frames to be protected discarded if the assigned SA cannot be used?	M	10.5.1	Yes[ ]
GEN-5	Is the PN value of zero used?	X	10.5.2	No [ ]
GEN-6	Following assignment of a PN to a protected frame, is the next frame to be protected for the same SA assigned the next higher value of PN?	M	10.5.2	Yes[ ]
GEN-7	Is the SecTAG encoded as specified in Clause 9?	M	10.5.3, Clause 9	
GEN-8	Is the ES bit set or clear as required by the management controls useES and alwaysIncludeSCI?	M	10.5.3	
GEN-9	Is the SC bit set or clear and the SCI explicitly encoded or not as required by the management controls useES, use SCB, alwaysIncludeSCI, and by the number of receive SCs?	M	10.5.3	
GEN-10	Is the SCB bit set or clear as required by the management controls useSCB and alwaysIncludeSCI?	M	10.5.3	
GEN-11	Is the E bit set if the frame is confidentiality protected, and clear otherwise?	M	9.5	
GEN-12	Is the C bit set if the octets of the Secure Data differ from those of the User Data or the ICV is not 16 octets, and clear otherwise?	M	9.5	
GEN-13	Is each frame transmitted from the Controlled Port protected using a Cipher Suite as specified in Clause 14 if protectFrames is set?	M	10.5	
GEN-14	Is OutOctetsEncrypted incremented by the number of octets in the User Data if confidentiality protections is provided, and OutOctetsProtected incremented otherwise?	M	10.5.4	
GEN-15	Is the protected frame transmitted if the MACsec PDU (SecTAG, Secure Data, and ICV) does not exceed the maximum data unit size supported by the Common Port and discarded otherwise?	M	10.5.5	

H.9 Secure Frame Verification

Item	Feature	Status	References	Support
VER-1	For each receive indication, does the Secure Frame Verification process examine the user data for a SecTAG and validate frames with a SecTAG as specified in 9.12, extracting and decoding the SecTAG as specified in 9.3 through 9.9, and extracting the User Data and ICV as specified in 9.10 and 9.11?	M	10.6, 9.3 through 9.9, 9.10, 9.11, 9.12	Yes[ ]
VER-2	Is a received frame without a SecTAG delivered to the Controlled Port if validateFrames is not Strict, and discarded otherwise?	M	10.6	Yes[ ]
VER-3	Is a received frame with the SecTAG E bit set and C bit clear discarded and not delivered to the Controlled Port?	M	10.6	Yes[ ]
VER-4	Is the received frame discarded if the SC is unknown and validateFrames is Strict or the C bit is set, and delivered to the Controlled Port otherwise?	M	10.6.1	Yes[ ]
VER-5	Is the received frame discarded if the SA is unused and validateFrames is Strict or the C bit is set, and delivered to the Controlled Port otherwise?	M	10.6.1	Yes[ ]
VER-6	Is the received frame discarded if the PN is less than the lowest acceptable packet number for the SA and replayProtect is enabled?	M	10.6.2, 10.6.4	Yes[ ]
VER-7	Is the InPktsOverrun counter incremented if a received frame is discarded for reasons not attributed to the data conveyed?	M	10.6.3	Yes[ ]
VER-8	If validateFrames is Disabled, is Cipher Suite validation omitted and a received frame delivered to the Controlled Port if the C bit is not set?	M	10.6.3, 10.6.5	Yes[ ]
VER-9	If validateFrames is not Disabled is the Cipher Suite used to validated the received frame?	M	10.6.3	Yes[ ]
VER-10	Are frames that are not successfully validated discarded if validateFrames is Strict or the C bit is set?	M	10.6.5	Yes[ ]
VER-11	Are the values for the next expected and lowest acceptable PN updated as specified in 10.6.5 following receipt of a MACsec PDU successfully validate by the Cipher Suite, and not modified by received frames otherwise?	M	10.6.5	Yes[ ]
VER-12	Are received frames not discarded by Secure Frame Verification delivered to the Controlled Port after removal of a SecTAG and ICV?	M	10.6	Yes[ ]
VER-13	Are all received frames delivered to Controlled Port unmodified if validateFrames is Null?	M	10.6	Yes[ ]
VER-14	Is protectFrames set False if validateFrames is set to Null?	M	10.6	Yes[ ]

H.10 MACsec PDU encoding and decoding

Item	Feature	Status	References	Support
FMT-1	Does each MACsec PDU transmitted contain an integral number of octets?	M	9.1	Yes[ ]
FMT-2	Does each MACsec PDU transmitted comprise a SecTAG, formatted as specified in Clause 9, one or more octets of Secure Data, and an ICV of the length specified by the Cipher Suite in use?	M	9.1, 9.2, 9.3, Figure 9-1, 10.5.3	Yes[ ]
FMT-3	Is the EtherType encoded in the SecTAG the value specified in Table 9-1?	M	9.3, 9.4	Yes[ ]
FMT-4	Is the version number in the SecTAG encoded as zero?	M	9.5	Yes[ ]
FMT-5	Is the SC bit clear and the SCI not explicitly encoded if the ES bit is set?	M	9.5	Yes[ ]
FMT-6	Is the SC bit set if an SCI is explicitly encoded and clear otherwise?	M	9.5	Yes[ ]
FMT-7	Is the SC bit clear if the SCB bit is set?	M	9.5	Yes[ ]
FMT-8	Are bits 7 and 8 of octet 4 of the SecTAG zero?	M	9.7	Yes[ ]
FMT-9	Is each received MACsec PDU validated as specified in 9.12.	M	9.5	Yes[ ]

H.11 Key Agreement Entity LMI

Item	Feature	Status	References	Support
KAY-1	Does the implementation allow the KaY to read the values of the MAC_Enabled, MAC_Operational, and operPointToPointMAC parameters?	M	10.7.2	Yes[ ]
KAY-2	Does the implementation allow the KaY to set and clear the ControlledPortEnabled parameter, acting on the parameter as specified?	M	10.7.4, 10.7.5	Yes[ ]
KAY-3	Does the implementation allow the KaY to discover which Cipher Suites are implemented and how many receive SCs each can support?	M	10.2, 10.7.7, 10.7.16, 10.7.25	Yes[ ]
KAY-4	Does the implementation allow the KaY to create a receive SC?	M	10.6.1, 10.7.11	Yes[ ]
KAY-5	Does the implementation allow the KaY to create receive SAs as specified in 10.7.13?	M	10.7.13	Yes[ ]
KAY-6	Does the implementation allow the KaY to control the use of each receive SA and to update the values of the next expected PN and lowest acceptable PN as specified in 10.7.15?	M	10.7.15	Yes[ ]
KAY-7	Does the implementation allow the KaY to create transmit SAs as specified in 10.7.22?	M	10.7.22, 10.5.2	Yes[ ]
KAY-8	Does the implementation allow the KaY to control the use of each transmit SA as specified in 10.7.24?	M	10.7.24, 10.5.1, 10.5.2	Yes[ ]
KAY-9	Does the implementation allow the KaY to monitor the nextPN associated with each transmit SA in order to create a new SA with a fresh SAK prior to PN exhaustion?	M	10.7.2	Yes[ ]
KAY-10	Does the implementation allow the KaY to select the Current Cipher Suite as specified in 10.7.27?	M	10.7.27	Yes[ ]
KAY-11	Does the implementation allow the KaY to create and control an SAK as specified in 10.7.26 and 10.7.28?	M	10.7.26, 10.7.28	Yes[ ]

H.12 Management

H.12.1 Management—control and status information

Item	Feature	Status	References	Support
Can each of the following parameter values be read by management?				
MGT1-1	The SCI for the SecY	M	10.7.1	Yes[ ]
MGT1-2	MAC_Enabled, MAC_Operational, and operPointToPointMAC for the Uncontrolled Port	M	10.7.2	Yes[ ]
MGT1-3	MAC_Enabled, MAC_Operational, and operPointToPointMAC for the Controlled Port	M	10.7.4	Yes[ ]
MGT1-4	The maximum number of receive SCs and SAKs that can be in simultaneous use	M	10.7.7	Yes[ ]
MGT1-5	validateFrames, replayProtect, and replayWindow	M	10.7.8	Yes[ ]
MGT1-6	The SCI, receiving, createdTime, startedTime, and stoppedTime for each receive SC	M	10.7.12	Yes[ ]
MGT1-7	inUse, nextPN, lowestPN, createdTime, startedTime, stoppedTime, and Key Identifier for each receive SA	M	10.7.14	Yes[ ]
MGT1-8	The maximum number of SAKs that can be in simultaneous use for transmission	M	10.7.16	Yes[ ]
MGT1-9	protectFrames, useES, useSCB, and alwaysIncludeSCI	M	10.7.17	Yes[ ]
MGT1-10	transmitting, createdTime, startedTime, and stoppedTime for the transmit SC	M	10.7.21	Yes[ ]
MGT1-11	inUse, nextPN, lowestPN, createdTime, startedTime, stoppedTime, and Key Identifier for each transmit SA	M	10.7.23	Yes[ ]
MGT1-12	The currentCipherSuite identifier and the confidentialityOffset for frames with confidentiality protection	M	10.7.27	Yes[ ]
MGT1-13	transmits, receives, and createdTime for each SAK	M	10.7.29	Yes[ ]
MGT1-14	Can the management information for each implemented Cipher Suite be read?	M	10.7.25	Yes[ ]

H.12.2 Management—basic controls

Item	Feature	Status	References	Support
Can the following parameters be written by management, independently for each Controlled Port?				
MGT2-1	validateFrames	O	10.7.8, 10.6	Yes[ ] No [ ]
MGT2-2	replayProtect	O	10.7.8, 10.6.2, 10.6.4	Yes[ ] No [ ]
MGT2-3	replayWindow	O	10.7.8, 10.6.5	Yes[ ] No [ ]
MGT2-4	protectFrames	O	10.7.17, 10.5	Yes[ ] No [ ]
MGT2-5	useES	O	10.7.17, 10.5.3	Yes[ ] No [ ]
MGT2-6	useSCB	O	10.7.17, 10.5.3	Yes[ ] No [ ]
MGT2-7	alwaysIncludeSCI	O	10.7.17, 10.5.3	Yes[ ] No [ ]
Can the following parameters be written by management, independently for each Controlled Port for each CipherSuite implemented ?				
MGT2-15	enableUse	O	10.7.26	Yes[ ] No [ ]
MGT2-16	requireConfidentiality	O	10.7.26	Yes[ ] No [ ]
Can write access by management to each of the following parameters be disabled individually?				
MGT2-8	validateFrames	MGT2-1:M	10.7.8	Yes[ ]
MGT2-9	replayProtect	MGT2-2:M	10.7.8	Yes[ ]
MGT2-10	replayWindow	MGT2-3:M	10.7.8	Yes[ ]
MGT2-11	protectFrames	MGT2-4:M	10.7.17	Yes[ ]
MGT2-12	useES	MGT2-5:M	10.7.17	Yes[ ]
MGT2-13	useSCB	MGT2-6:M	10.7.17	Yes[ ]
MGT2-14	alwaysIncludeSCI	MGT2-7:M	10.7.17	Yes[ ]
Can write access by management to each of the following CipherSuite use parameters be disabled individually for each Controlled Port?				
MGT2-17	enableUse	MGT2-15:M	10.7.26	Yes[ ]
MGT2-18	requireConfidentiality	MGT2-16:M	10.7.26	Yes[ ]

H.12.3 Management—control over secure communication

Item	Feature	Status	References	Support
Can the following be created, controlled, or selected by management?				
MGT3-1	Receive SCs and SAs	O	10.7.11, 10.7.13, 10.7.15	Yes[ ] No [ ]
MGT3-2	Transmit SAs	O	10.7.22, 10.7.24	Yes[ ] No [ ]
MGT3-3	The current CipherSuite	O	10.7.27	Yes[ ] No [ ]
MGT3-4	confidentialityOffset	O	10.7.27	Yes[ ] No [ ]
MGT3-5	SAKs	O	10.7.28, 10.7.29	Yes[ ] No [ ]
Can creation, control, or selection by management of the following be disabled individually?				
MGT3-1	Receive SCs and SAs	MGT3-1:M	10.7.11	Yes[ ]
MGT3-2	Transmit SAs	MGT3-2:M	10.7.22, 10.7.24	Yes[ ]
MGT3-3	The current CipherSuite	MGT3-3:M	10.7.27	Yes[ ]
MGT3-4	confidentialityOffset	MGT3-4:M	10.7.27	Yes[ ]
MGT3-5	SAKs	MGT3-5:M	10.7.27	Yes[ ]

H.12.4 Management—statistics

Item	Feature	Status	References	Support
Are each of the following interface statistics provided for the Controlled Port as specified in 10.7.6?				
MGT4-1	ifInOctets	M	10.7.6	Yes[ ]
MGT4-2	ifInUcastPkts, ifInMulticastPkts, ifInBroadcastPkts	M	10.7.6	Yes[ ]
MGT4-3	ifInDiscards	M	10.7.6	Yes[ ]
MGT4-4	ifInErrors	M	10.7.6	Yes[ ]
MGT4-5	ifOutOctets	M	10.7.6	Yes[ ]
MGT4-6	ifOutUcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts	M	10.7.6	Yes[ ]
MGT4-7	ifOutErrors	M	10.7.6	Yes[ ]
Are each of the following frame verification statistics recorded as specified in 10.6 and maintained for the frame verification process as a whole?				
MGT4-8	InPktsUntagged	M	10.7.9, 10.6 Figure 10-4	Yes[ ]
MGT4-9	InPktsNoTag	M	10.7.9, 10.6 Figure 10-4	Yes[ ]
MGT4-10	InPktsBadTag	M	10.7.9, 10.6 Figure 10-4	Yes[ ]
MGT4-11	InPktsNoSARcv	M	10.7.9, 10.6.1	Yes[ ]
MGT4-12	InPktsNoSADiscard	M	10.7.9, 10.6.1	Yes[ ]
MGT4-13	InPktsOverrun	M	10.7.9, 10.6.3	Yes[ ]

H.12.4 Management—statistics *(continued)*

Item	Feature	Status	References	Support
Are each of the following frame verification statistics recorded as specified in 10.6 and maintained for each receive SC?				
MGT4-14	InPktsUnchecked	M	10.7.9, 10.6.5	Yes[ ]
MGT4-15	InPktsDelayed	M	10.7.9, 10.6.5	Yes[ ]
MGT4-16	InPktsLate	M	10.7.9, 10.6.2, 10.6.4	Yes[ ]
MGT4-17	InPktsOK	M	10.7.9, 10.6.5	Yes[ ]
MGT4-18	InPktsInvalid	M	10.7.9, 10.6.5	Yes[ ]
MGT4-19	InPktsNotValid	M	10.7.9, 10.6.5	Yes[ ]
Are each of the following frame validation statistics recorded as specified in 10.7?				
MGT4-22	InOctetsValidated	M	10.7.10	Yes[ ]
MGT4-23	InOctetsDecrypted	M	10.7.10	Yes[ ]
Are each of the following frame generation statistics recorded as specified in 10.5 and maintained for the frame verification process as a whole?				
MGT4-24	OutPktsUntagged	M	10.7.18, 10.5	Yes[ ]
MGT4-25	OutPktsTooLong	M	10.7.18, 10.5.5, Figure 10-3	Yes[ ]
Are each of the following frame generation statistics recorded as specified in 10.5 and maintained for each transmit SC?				
MGT4-26	OutPktsProtected	M	10.7.18, 10.5.4	Yes[ ]
MGT4-27	OutPktsEncrypted	M	10.7.18, 10.5.4	Yes[ ]
Are each of the following frame protection statistics recorded as specified in 10.7?				
MGT4-28	OutOctetsProtected	M	10.7.19	Yes[ ]
MGT4-29	OutOctetsEncrypted	M	10.7.19	Yes[ ]

H.13 Additional fully conformant Cipher Suite capabilities

Item	Feature	Status	References	Support
CSA-1	Name of Cipher Suite as specified in Clause 14. _			
CSA-2	Does the Cipher Suite implementation provide integrity without confidentiality?	O	14.2(a)	Yes[ ] No [ ]
CSA-3	Does the Cipher Suite implementation provide confidentiality for all the octets of the User Data?	¬CSV-19: O CSV-19: M	14.2(d), 14.3(c)	Yes[ ] No [ ]
CSA-4	Does the Cipher Suite implementation provide offset confidentiality for the User Data?	O	14.2(e), 14.3(c)	Yes[ ] No [ ]
CSA-5	What is the maximum number of receive SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSA-6	What is the maximum number of receive SAKs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSA-7	What is the maximum number of transmit SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	

H.14 Additional variant Cipher Suite capabilities

Item	Feature	Status	References	Support
CSV-1	Name of Cipher Suite or other commonly used identification (to be supplied) _			
CSV-2	Identify the specification(s) of the Cipher Suite, including any additional information necessary to acquire the specification(s) (supply items of Additional Information if necessary) _	M	14.3	
CSV-3	Does the specification include interoperable protection and verification procedures specified in terms of the parameters of 14.1?	M	14.3, 14.1	Yes [ ]

CSV-4	Does the specification state:
	Whether confidentiality of the User Data is provided?
	The maximum difference in the lengths of the User Data and Secure Data?
	The length of the ICV?
	The length and properties of the keys required, including assumptions of the scope and uniqueness?
		M	
			14.3(a)
			
			14.3(b)
			14.3(c)
			
			14.3(d)
				
				Yes [ ]
				
				Yes [ ]
				Yes [ ]
				
				Yes [ ]

CSV-5	Do the Cipher Suite algorithms have an effective key length of at least 128 bits, and does any block cipher used have a block width of at least 128 bits?	M	14.4.1(a)	Yes[ ]

H.14 Additional variant Cipher Suite capabilities *(continued)*

Item	Feature	Status	References	Support
CSV-6	If serviced by separate algorithms, the properties of the authentication and confidentiality mechanisms are combinable in accordance with well-established security results?	M	14.4.1(b)	Yes[ ]
CSV-7a	Is the underlying cryptographic cipher approved by either a national or international standards body or a government agency?	O.1	14.4.1(c)(1)	Yes[ ] No[ ]
CSV-7b	Does the additional Cipher Suite meet the conditions expressed in 14.4.1(c)(2)?	O.1	14.4.1(c)(2)	Yes[ ] No[ ]
CSV-8	Does the Cipher Suite satisfy the message authentication requirements of 14.4.1? Identify the proof of security, including any additional information necessary to acquire the proof _	CSV-7b:M	14.4.1(c)(2)(i)	Yes [ ]
CSV-9	Does the Cipher Suite satisfy the confidentiality requirements of 14.4.1? Identify the proof of security, including any additional information necessary to acquire the proof _	CSV-7b:M	14.4.1(c)(2)(ii)	Yes [ ]
CSV-10	Does the Cipher Suite use mechanisms for confidentiality and authentication in a way that is consistent with the proofs of security?	CSV-7b:M	14.4.1(c)(2)(iii), 14.4.1(c)(2)(iv),	Yes[ ]
CSV-11	Does the Cipher Suite provide integrity protection for the SCI, PN, Source Address, Destination Address, SecTAG, and User Data?	M	14.2(a)	Yes[ ]
CSV-12	Does the Cipher Suite provide protection for at least 2^{32}-1 invocations without requiring a fresh SAK?	M	14.2(b)	Yes[ ]
CSV-13	Does the Cipher Suite generate a predictable number of octets of Secure Data and ICV given any specific number of octets of User Data?	M	14.2(c)	Yes[ ]
CSV-14	Does the maximum difference in length of the User Data and the Secure Data plus ICV exceed 896 octets?	X	14.2(f)	Yes[ ]
CSV-15	What is the maximum difference in length of the User Data and the Secure Data? _ _ _ _ _ octets		14.3(b)	
CSV-16	What is the length of the ICV _ _ _ _ _ octets		14.3(e)	
CSV-17	Does the specification specify the length and properties of the keys required, including assumptions of the scope of uniqueness?	M	14.3(f)	Yes[ ]
CSV-18	Does the Cipher Suite implementation provide confidentiality for all the octets of the User Data?	¬CSV-19: O CSV-19:M	14.2(d), 14.3(c)	Yes[ ] No [ ]

H.14 Additional variant Cipher Suite capabilities *(continued)*

Item	Feature	Status	References	Support
CSV-19	Does the Cipher Suite implementation provide offset confidentiality for the User Data?	O	14.2(e), 14.3(c)	Yes[ ] No [ ]
CSV-20	Does the Cipher Suite modify or constrain the values of the SCI, PN, Source Address, Destination Address, or SecTAG fields other than as specified in Clause 14?	X	14.2(g)	No [ ]
CSV-21	Does the Cipher Suite require an SAK exceeding 1024 bits long?	X	14.2(h)	No [ ]
CSV-22	Does the Cipher Suite require different keys for the protect and validate operations?	X	14.2(i)	No [ ]
CSV-23	What is the maximum number of receive SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSV-24	What is the maximum number of receive SAKs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	
CSV-25	What is the maximum number of transmit SCs supported by the Cipher Suite implementation? _ _ _ _ _		5.3(m)	