User Manual

BCM84756

MACsec User Manual

84756-UM101-R

5300 California Avenue • Irvine, CA 92617 • Phone: 949-926-5000 • Fax: 949-926-5203

July 7, 2014

Revision History

Revision	*Date*	*Change Description*
84756-UM101-R ***Note:*** The reference numbers are valid for this revision of document only.	07/07/14	**Updated:** • "Egress Packet Flow" on page 47.
84756-UM100-R	05/04/12	Initial release

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Table of Contents

List of Figures

List of Tables

About This Document

Purpose and Audience

This document describes how to program MACsec modules in the BCM84756.

In this document, the terms set, set this bit, and enable this bit refer to setting this bit to 1, unless stated explicitly otherwise. Ingress means from line-side to switch-side direction and egress means from switch-side to line-side. Occasionally receive path is used to mean ingress and transmit path is used for egress.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

Acronyms and abbreviations in this document are also defined in Appendix E: “Acronyms and Abbreviations,” on page 82.

For a comprehensive list of acronyms and other terms used in Broadcom documents, go to: http://www.broadcom.com/press/glossary.php.

Document Conventions

The following conventions may be used in this document:

Convention	***Description***
Bold	User input and actions: for example, type **exit**, click **OK,** press **Alt+C**
`Monospace`	Code: `#include <iostream>` HTML: `<td rowspan = 3>` Command line commands and parameters: `wl [-l] <command>`
< >	Placeholders for *required* elements: enter your <username> or `wl <command>`
[ ]	Indicates *optional* command-line parameters: `wl [-l]` Indicates bit and byte ranges (inclusive): [0:3] or [7:0]

References

The references in this section may be used in conjunction with this document.

Note: Broadcom provides customer access to technical documentation and software through its Customer Support Portal (CSP) and Downloads and Support site (see Technical Support).

Document (or Item) Name	*Number*	*Source*
[1] MACsec and Fiber Channel over Ethernet (FCoE) Features	84756_84757_84758-PR1xx-R	CSP

Technical Support

Broadcom provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates through its customer support portal (https://support.broadcom.com). For a CSP account, contact your Sales or Engineering support representative.

In addition, Broadcom provides other product support through its Downloads and Support site (http://www.broadcom.com/support/).

Introduction

This document describes how to program MACsec modules in the BCM84756. The main purposes of this document are to:

- Show how to enable MACsec and bring up MACsec for normal operations.
- Give the programming sequence of initializing various MACsec registers.
- Give the sequence to program MACsec for ingress flow and egress flow.

The MACsec implementation is a single secure port (SP) design, which supports 10 Gbps, 1 Gbps, 100 Mbps, and 10 Mbps. This implementation is called XSP core, which contains the entire MACsec implementation for a single port. Note that XSP core has XGMII, GMII, and MII interfaces, which are connected to PHY and SerDes blocks on the line-side and switch-side, respectively.

In this document, the terms set, set this bit, and enable this bit refer to setting this bit to 1, unless stated explicitly otherwise. Ingress means from line-side to switch-side direction and egress means from switch-side to line-side. Occasionally receive path is used to mean ingress and transmit path is used for egress.

Architecture Overview

This section gives a high-level overview of the architecture.

The BCM84756 has four ports, each of which supports MACsec. Each port is integrated with an identical XSP core. There is no difference for MACsec blocks among the four ports.

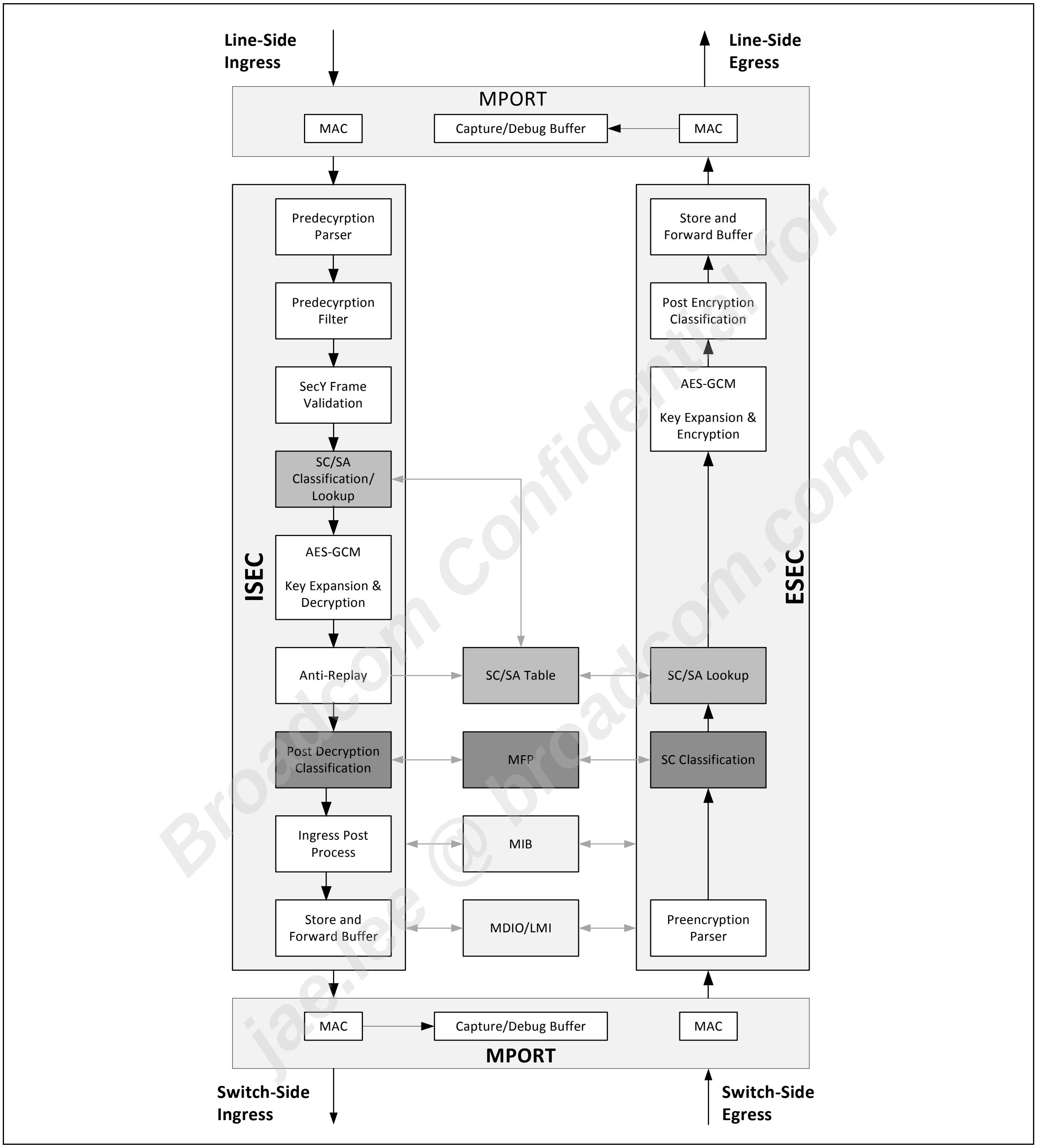
Block Diagram

Each XSP core consists of various subblocks, as described in Table 1.

Table 1: XSP Subblock Description

Subblock	*Subblock Function*
Mport	MACsec port contains both the XMAC (for 10 Gbps speed) and UNIMAC (for 1 Gbps, 100 Mbps, and 10 Mbps speed) along with logic to support pause frame generation, Priority-based Flow Control (PFC) forward, fixed latency mode, ingress and egress flow control, as well as provide a unified interface to the SECY agents (namely Ingress Security and Egress Security). Mport also provides the stats vector for Remote Monitoring (RMON) updates to MIB.
ISEC	The Ingress Security (ISEC) module functions as the MACsec link layer security pipeline for ingress (receive) traffic going from line side to switch (system) side.
ESEC	The Egress Security (ESEC) module functions as the MACsec link layer security pipeline for egress (transmit) traffic going from switch (system) side to line side.
MFP	The MFP consists of Ingress Flow Processor (IFP) and Egress Flow Processor (EFP). The IFP parses the post SECY received packet in terms of the MACsec processing status and the L2 information, and generates the action for the packet. Similar to the receive side, the EFP parses the packet in terms of the L2 information and generates the action for the packet.
SC/SA Table	This module contains the Secured Channel (SC) and Secured Association (SA) look-up tables for the ISEC and ESEC pipelines. The host has the ability to access any table during MACsec ingress and egress packet processing. This is also referred to as the XTABLE.
MDIO and LMI	MACSec core management is achieved through the MDIO interface. The MDIO controller receives the command from host to write/read the registers in Layer Management Interface (LMI).
SECY Config	SECY configures the ISEC, ESEC, MPORT, XTABLE, and MIB.
MIB	MIB block implements RMON counters and security counters in the IEEE 802.1AE and RFC 2863 block. It also implements additional statistics counters.

Figure 1: XSP Block Diagram



Since XSP is a single-port design, each port can be independently reset or powered up/down. When the port is configured to run at a speed other than 10 Gbps, UNIMAC is used instead of XMAC.

The MACsec is running at core clock at 156.25 MHz. It can be gated off by hardware power-down and software power-down.

ISEC

The Ingress Security (ISEC) module performs MACsec SecY receiving functions that are fully compliant to the IEEE 802.1AE standard. The major features are:

- Supports 802.1AE controlled and uncontrolled port at up to 10 Gbps throughput.
- Supports hardware-based L2 packet classification.
- SC/SA lookup based on classification results or explicitly tagged SCI from the system. SC lookup supports bit/byte masking.
- Supports up to 16 SCs with 32 SAs (one, two, or four SAs per SC).
- Frame decryption with flexible ConfidentialityOffset and ICV verification.
- Receive SA anti-replay check and update. Out-of-order can be accepted with a programmable window per SA.
- Receive MIB counter update.
- Non-interrupting receiving SA management.
- Optional modes to modify incoming packet SecTAG or leave unchanged. Modes to change SecTAG include:
  - Remove SecTag.
  - SecTAG to ST-VLAN mapping.
  - Overwriting PN with a programmable MACSEC ID.
  - Overwriting EtherType with programmable value.
  - Overwriting SecTAG EtherType with programmable value and PN with a programmable MACSEC ID.
- Optionally, forward packets that failed a SecY check with a ST-VLAN Tag that indicates an error condition.
- Optionally, redirect packets that failed a SecY check. Redirected packets are L2 encapsulated and the PN in the SecTag may optionally be replaced with the error status.
- Optionally, bypass the security processing and leave packets unchanged.
- Optionally, redirect packets to debug FIFO.
- Optionally, drop packets.
- Operate in cut-through or store-and-forward mode. Cut-through mode supports artificial CRC corruption to signal bad packet to switch.
- 32 KB Store and Forward packet buffer.
- Per-user priority MTU check.
- Minimum packet length is as small as 17 bytes (excluding CRC, which is a design limitation). 60 bytes can be received from the line as a minimum and loopback packets minimum are 21 bytes.
- 21 classification rules with 40 bit counters for each rule hit. Each rule can be enabled or disabled individually.

ESEC

The Egress Security (ESEC) module functions as the MACsec link layer security pipeline for egress (transmit) traffic going from switch (system) side to line side. The ESEC module resides in the MAC Security Core and is fully compliant to the IEEE802.1AE standard. It supports the following features:

- 802.1AE controlled and uncontrolled port.
- Four different data rates: 10 Mbps, 100 Mbps, 1 Gbps, and 10 Gbps.
- Up to 16 Secure Channels (SCs) with 32 SAs (one, two, or four SAs per SC).
- Hardware-based L2 packet classification.
- SC/SA lookup based on the SCI_Index provided by the EFP action table.
- SecTAG insertion with optional SCI and EPON-SCB support for packets protected by controlled port.
- Non-destructive loopback and does not affect fixed latency. Post encrypted packets are looped back to the ingress.
- Frame encryption with flexible ConfidentialityOffset, ICV generation and ICV insertion.
- Short Length (SL) update for short packets.
- Transmit MIB counter update.
- Transmit SA lifetime and packet sequence number (PN) update.
- Non-interrupting transmit SA management.
- Full-duplex flow control on system side and line side MAC port.
- 32 KB Store and Forward packet buffer.
- Automatic AN Switching.
- Packet output of Egress pipeline can be redirected by software to a debug capture FIFO (512 bytes).
- Uncontrolled SecTAG packets EtherType can be optionally replaced with standard EtherType of 0x88e5.
- Optionally drop packets.
- MTU check per user priority on all traffic.
- Optionally bypass the egress security processing and send out the packet without modification.
- Special VLAN Tag processing.

MFP

The Broadcom MACsec implementation of the XSP core supports sophisticated features to process traffic flows. These features are not required by IEEE 802.1AE standards. These extra features are achieved through the MACsec Flow Processor (MFP) block, which consists of the ingress flow processor (IFP) and the egress flow processor (EFP).

In the MACsec ingress direction, there are requirements beyond the IEEE 802.1AE standard that need the special processing of the receiving packets, switch capabilities, and various applications. For example, some switch ASICs can support SecTAG packets, some only support VLANs. Some switch ASICs may need to insert a special Layer 2 (L2) header to redirect the packet. An ingress post-MACsec packet classification [also called Ingress Filter Processor (IFP)] is needed to parse the packet in terms of the MACsec processing status and the L2 information, and generate the right action for the packet. The major features include:

- Parse the post-MACsec packets with SecTAG and ICV removed, if available.
- Support Ethernet II/LLC/SNAP packet formats.
- Support single VLAN Tag, double VLAN Tags, or no VLAN Tag. Four outer VLAN TAG TPIDs are configurable. One inner VLAN Tag TPID is configurable.
- Parse QTAG and STAG, and extract the user priority (UP) in terms of the configuration.
- Generate the lookup key for MFP_IFP_KEY_TABLE in terms of the L2 header and the ingress MACsec packet processing status.
- Look up MFP_IFP_KEY_TABLE to find the index of the matched entry. Only the lowest entry is selected if multiple entries match.
- Look up MFP_IFP_ACTION_Table to retrieve the action specified for the matched flow.
- Send the action data to ISEC for post processing.
- Increment the 48-bit counter MFP_MIB_TABLE per flow, if MFP_IFP_KEY_TABLE gets a hit. The counter is clear-on-read and saturated based on the global configuration.
- Provide TCAM ECC function.
- 128 IFP flows shared with EFP.

In the MACsec transmit path, any packets from the switch-side need to be classified to the Controlled Port or the Uncontrolled Port traffic. If the packet belongs to the Controlled Port traffic, a policy engine decides how SecY processes this packet. For example, which secure channel this packet belongs to, and which security policy is going to be applied to this packet (i.e., integrity or confidentiality plus integrity). Similar to the receive side, a pre-MACsec packet classification [also called Egress Filter Processor (EFP)] is needed to parse the packet in terms of the L2 information and generate the right action for the packet. The major features include:

- Parse the pre-MACsec packets with SecTAG and ICV, if available.
- Support Ethernet II/LLC/SNAP packets with or without SecTAG.
- Support Special VLAN Tagged packets without SecTAG.
- Support single VLAN Tag, double VLAN Tags, or no VLAN Tag. Four outer VLAN TAG TPIDs are configurable. One inner VLAN Tag TPID is also configurable.
- Parse QTAG and STAG, and extract the user priority (UP) in terms of the configuration.
- Classify the control packets in terms of the configuration. Provide per-rule action and 48-bit counter.
- Derive the packet_type in terms of the packet header information.
- Generate the lookup key for MFP_EFP_KEY_TABLE in terms of L2 information and the configurable key selection register.

- Look up MFP_EFP_KEY_TABLE to find the index of the matched entry. Only the lowest entry is selected if multiple entries match.
- Look up MFP_EFP_ACTION_Table to retrieve the action specified for the matched flow.
- Perform egress MacSecID processing and provide the action to ESEC.
- Derive the C and E bits from the packet or MFP_EFP_ACTION_Table based on the configuration.
- Derive the VLAN ID, which is used to replace the lower 12-bit of SCI, if enabled.
- Send the action data to ESEC for the further processing.
- Increment the 48-bit counter per flow if MFP_EFP_KEY_TABLE gets a hit. The counter is clear-on-read and saturated based on the global configuration.
- 128 EFP flows shared with IFP.
- 21 classification rules with 40 bit counters for each rule hit. Each rule can be enabled or disabled individually.

Fixed Latency Mode and Variable Latency Mode

By default, the SecY module operates in variable latency mode but it can be configured to operate in fixed latency mode in both ingress and egress directions. In the fixed latency mode, packets going through the SecY module have a fixed delay and it is defined via a programmable register. See “Fixed-Latency Configuration” on page 76 on how to program SecY to operate in fixed latency mode.

Registers Overview

MACsec has a large set of registers and memories. This section describes the main and commonly used registers. They can be grouped into three categories: Layer Management Interface (LMI), MACsec Flow Processor (MFP), and SECure PORT (SECPORT).

- The LMI registers consist of address registers, data registers, command registers. and status registers.
- The MACsec registers and memories can be indirectly accessed through the 16-bit LMI registers.
- The SECPORT registers are defined to implement IEEE MACsec standard features and MFP for Broadcom-proprietary features. MFP is for post-processing of ingress traffic and for pre-processing of egress traffic.

MACsec core management is achieved through the MDIO interface. MDIO controller receives the command to write/read the registers in LMI.

Each category has subblocks, as given below:

- LMI
- MFP
  - MFP
  - MFP_KEY
  - MFP_ACTION
  - MFP_MIB
- SECPORT
  - Buffers
    - Ingress
    - INGRESS_CAP_BUF
    - INGRESS_SAF_BUF
    - Egress
    - EGRESS_CAP_BUF
    - EGRESS_SAF_BUF
    - LOOPBACK_BUF
    - MPORT_PACKET_BUFFER
  - SC and SA Tables
    - Ingress
    - INGRESS_SC
    - INGRESS_SC_ACTION_TABLE
    - INGRESS_SA
    - Egress
    - EGRESS_SC
    - EGRESS_SA
  - ESEC
  - ISEC

- MIB
- MPORT
- UNIMAC
- XMAC

Most of the registers in the MFP group are table registers that define the keys to match incoming packets, the action to be taken for matched packets, and the counters for the number of packets matched in ingress or egress direction. Most of the registers are 32-bit except XMAC, and some counter registers that are 40- or 64-bit wide and table registers which have various sizes of width. Some MIB registers related to per SC counters and byte counters are 64 bits.

When the ports are configured to run at non-10 Gbps speed, they use the UNIMAC module, not XMAC.

There is a separate sector for RMON counters in PRG. In the HTML register document, RMON counters are in UNIMAC.

LMI

The main LMI registers are:

- LMI_Interrupt: Indicates if LMI and/or MACsec has generated an interrupt for the port.
- Port_LMI_Status: Indicates the port LMI status such as the done bit for the command executed or register/ memory access time out.
- LMI_Interrupt_Enable: Enables LMI and/or MACsec interrupt.
- Port_LMI_Timer: Set this register to enable time out and the timer.
- Port_LMI_Interrupt_EN: Enables various types of LMI interrupt events.

MFP

MFP has two categories of registers: control registers and table registers. Control registers are either for ingress or egress. Table registers are for MFP action, keys, and MIBs.

MFP Control Registers

The main control registers are:

- Egress_UDFn (n=0, 1): Specifies the offset of two user-defined egress fields starting from the beginning of the packets.
- Egress_KEY_Sel: EFP key selection slice bits. These bits determine how to interrupt EFP lookup keys for slice 1 and slice 2.
- Egress_Special_TPID: The TPID value of the special VLAN Tag used to indicate SCI.
- Egress_Outer_TPIDn (n=1, 2, 3): One of the outer TPID to identify a VLAN Tag.
- Egress_Inner_TPID: The inner TPID value to identify a VLAN Tag.
- Egress_CP_ETYPE_MAX_LEN: The programmable length for EII packet at egress. For packets whose length is equal or more than this length, it is considered as EII packet.
- Egress_VLAN_TAG_Control: Egress VLAN Tag parsing control register.

- MFP_INT_CSR: Enables and reports MACsec MFP interrupt control and status.
- Ingress_UDF: Specifies the offset of ingress user-defined field from the beginning of the packet.
- Ingress_Special_TPID: The TPID value of the special VLAN Tag at ingress.
- Ingress_Outer_TPIDn (n=1, 2, 3): One of the outer TPID value to identify a VLAN tag at ingress.
- Ingress_Inner_TPID: The inner TPID value to identify a VLAN tag at ingress.
- Ingress_CP_ETYPE_MAX_LEN: Programmable length for EII packet at ingress. For packets whose length is equal or more than this length, it is considered an EII packet.
- Ingress_VLAN_TAG_Control: Ingress VLAN Tag parsing control register.

MFP Table Registers

There are various tables in MACsec such as the secure channel (SC) table and its security association (SA) table, and MFP action tables, key tables, etc. These tables can be accessed through registers with different depth. The register with base address is used to access the first entry in the table; the register with base address plus offset 1 is used to access the second entry in the table; and the register with base address plus offset i is used to access the i+1 entry in the table.

MFP table registers are shared between ingress and egress per port. In other words, there are dedicated tables for field processing for each port. But these tables for each port are shared between ingress and egress.

Each port has the following MFP tables:

- Key table–MFP_EFP_KEY_TABLE (for egress); MFP_IFP_KEY_TABLE (for ingress): 128 entries.
- Action table–MFP_EFP_ACTION_TABLE (for egress); MFP_IFP_ACTION_TABLE (for ingress): 128 entries.
- MIB table–MFP_MIB_TABLE: 128 entries.

Note that the egress table, MFP_EFP_KEY_TABLE, is the same as the ingress table, MFP_IFP_KEY_TABLE, and the egress MFP_EFP_ACTION_TABLE is the same as the ingress MFP_IFP_ACTION_TABLE. Each entry in the table has a different meaning (format) for ingress vs. egress. There are 128 entries in each of the above tables. The i-th entry in one table corresponds to the i-th entry in another table. The direction bit (bit 233) of an entry at key table determines whether this entry is for ingress or egress. If the i-th entry at key table is for ingress, then the i-th entry at action table is for ingress, too. If the j-th entry at key table is for egress, then the j-th entry at action table is for egress too. For packet matches, the i-th entry at the key table, the action defined by the i-th entry at action table is applied. At the same time, the counter in the i-th entry at the MIB table is incremented by one. The i-th entry of the table can be accessed by the base address with offset i-1. When one direction uses fewer entries, the other direction can use more entries. For example, if the MFP_IFP_KEY Table uses 28 entries, the MFP_EFP_KEY Table can use a maximum of 100 entries.

There are 128 entries in MFP_MIB_TABLE. Each entry is a mapping to one entry in the Key table and one in the Action table. Whenever a packet matches the i-th entry of the Key Table, the i-th entry of MFP_MIB_TABLE is incremented by 1.

The host can access the table registers in MFP. The software must initialize MFP_IFP_KEY_Table and MFP_EFP_KEY_Table to 0. All counters in MFP_MIB_Table also need to be cleared to 0 by setting SECY_CFG_GLB_MIB_Ctrl.CLR_MIB_CNT.

The MFP module has ECC error detections for all tables. All action tables with double-bit ECC error cause the packet to be treated as error packets like CRC error packets, even if it is an IFP or EFP action table. The MFP MIB table ECC error does not cause the packet to be treated as an error packet. Any ECC errors generated by a CPU memory read may cause an interrupt to be issued. No other actions taken. All single-bit ECC errors are logged to a 16-bit counter and the double-bit ECC errors are logged to another 16-bit counter, namely MFP_ECC_COUNT register, Bits[31:16] (MFP_2bit_ECC_COUNT) and Bits[15:0] (MFP_1bit_ECC_COUNT). Meanwhile, an interrupt per memory is issued to inform the host of the errored address.

The MFP_IFP_KEY_TABLE and MFP_EFP_KEY_TABLE are implemented with TCAM. The TCAM has a Data-scrubber facility to assist in self correction from bit errors. When the host writes the TCAM, the TCAM data ECC fields are stored outside the TCAM. For each TCAM, there are four single-bit error correction and double-bit detection ECC fields (first, second, third, and fourth bit interleaved) protection. Once the background TCAM ECC checking is enabled by setting the bit MFP_SCRUB_Control.TCAM_SCRUB_EN, the TCAM scrub request is sent out, if the timer hits the MFP_SCRUB_Control.SCRUB_Interval. Note that the register MFP_SCRUB_Scale provides more granularities for the timer that increment the timer by 1 whenever the free-running counter is equal to MFP_SCRUB_Scale.SCRUB_SCALE. The MFP search engine starts to read both ECC fields and TCAM data entry-by-entry, and checks the ECC of the TCAM data. If a mismatch or an ECC error is found, the MFP search engine does the following:

- For a single-bit ECC error, if tcam_ecc_1b_en is not set, nothing is done. If both tcam_ecc_1b_en and tcam_ecc_1b_invalidate are set, the TCAM entry is invalidated. If tcam_ecc_1b_en is set and tcam_ecc_1b_invalidate is not set, the corrected data is written back to TCAM.
- For the double-bit ECC error, the TCAM entry is invalidated, if tcam_ecc_2b_invalidate is set. Otherwise, nothing is done.
- For both single-bit and double-bit TCAM ECC errors, an interrupt and the errored address are generated to inform the host.

This ECC error checking is providing a non-interrupt service (i.e., it’s running in the background). Once there are any lookup key requests coming from IFP or EFP, it finishes the current address check and gives the priority to the regular data flow. After the lookup key request is served and there is no next request, the ECC error checking starts again from the next memory address. Once the whole memory is checked, a status is updated to tell the software the ECC check result. MFP_INT_CSR.MFP_KEY_TABLE_2b_ECC_INT or MFP_INT_CSR.MFP_KEY_TABLE_1b_ECC_INT is set and the address is logged in MFP_2b_ECC_ERROR_STATUS.MFP_KEYECC_ERROR_ADDR or MFP_1b_ECC_ERROR_STATUS.MFP_KEYECC_ERROR_ADDR if there is a TCAM ECC error.

SECPORT

Buffers

There are capture buffers and store and forward buffers at both ingress and egress, namely:

- INGRESS_CAP_BUF: Ingress capture buffer. There are 32 entries.
- INGRESS_SAF_BUF: Store and forward packet buffer for ingress. There are 2048 entries.
- EGRESS_CAP_BUF: Egress capture buffer. There are 32 entries.
- EGRESS_SAF_BUF: Store and forward packet buffer for egress. There are 2048 entries.
- LOOPBACK_BUF: Loopback buffer for packet loopbacked from egress to ingress. There are 32 entries.
- MPORT_PACKET_BUFFER: Provide a way to access ingress and egress packet buffer in fibre mode.

Note that each entry in the above tables contains 16-byte packet data plus extra bits for sideband information.

SC and SA Table Registers

Each port has the following SC and SA table registers:

- 0x0B10_0000 INGRESS_SC_INDEX_TABLE: 16 entries
- 0x0810_0000 INGRESS_SA_ATTRIBUTE_TABLE: 32 entries
- 0x0810_8000 INGRESS_SA_KEY_TABLE: 32 entries
- 0x0A10_0000 INGRESS_SC_ACTION_TABLE: 16 entries
- 0x0C10-0000 EGRESS_SC_INDEX_TABLE: 16 entries
- 0x0910_0000 EGRESS_SA_ATTRIBUTE_TABLE: 32 entries
- 0x0910_8000 EGRESS_SA_KEY_TABLE: 32 entries

There are separate SC tables, namely INGRESS_SC_INDEX_TABLE and EGRESS_SC_INDEX_TABLE, for ingress and egress, respectively. One entry at the SC table is mapped to two entries at the SA tables (SA_ATTRIBUTE_TABLE and SA_KEY_TABLE) for two SAs per SC configuration; it is mapped to four entries at the SA tables for four SAs per SC. With four SAs per SC, we can only program eight entries at SC tables. With two SAs per SC, all 16 entries of SC table can be programmed.

ESEC Registers

The main control registers at ESEC are:

- SP_Egress_Ctrl: Enables various kinds of interrupts at egress.
- SP_Egress_PN_THD: The threshold to trigger egress soft PN expiration event.
- SP_Egress_SA_status0: A status vector indicates which egress SA has expired (one bit for each SA).
- SP_Egress_SA_status1: A status vector indicates which egress SA has soft-expired (one bit for each SA).
- SP_MTUn (n=0, 1, …, 7): The maximum transmission unit size for the n-th user priority at transmit.
- SP_Egress_Packet_Drop_Count: The total number of packets dropped at egress.
- SP_Egress_Packet_ECC_Drop_Count: The number of packets dropped due to double-bit error at egress.
- SP_Egress_Packet_Drop_Status: The various egress packet drop statuses due to different error conditions. It also has bits to trigger interrupts for these different error conditions.

- SP_Egress_AUTO_AN_SWITCH_CTRL: Enables auto AN switch on per SC basis, 1-bit for each SC.
- SP_EGRESS_CAP_CTR0: Various bits to control capture buffer.
- SP_EGRESS_CAP_CTR1: Controls the size of the egress capture buffer and also the maximum packet length.
- SP_EGRESS_CAP_STAT: Indicates the current usage of the egress capture buffer.
- SP_EGRESS_CAP_DROP_CNT: The number of packets dropped at egress capture buffer.
- SP_Egress_STAT: Egress interrupt status bits. It is cleared after read.
- SP_Egress_ECC_Count: Egress 1-bit and 2-bit error counts. It is cleared after read.
- SP_Egress_Table_ECC_Count: Egress SA and SC table 1-bit and 2-bit error counts. It is cleared after read.

ISEC Registers

These are the main SecY port control registers at ISEC:

- SECY_CFG_GLB_INT_CSR: MACsec interrupt status register.
- SECY_CFG_GLB_MIB_Ctrl: Global MIB control register to clear or disable counter saturation.
- SECY_CFG_CNTMAXSIZE: The maximum packet size used in RMON statistic counter updates.
- SP_MASTER_Ctrl: Secure port master control register.
- SP_Ingress_Ctrl: Ingress interrupt control register. This register has bits each which can enable each interrupt for one kind of error event at ingress.
- SP_Ingress_PN_THD: Threshold to trigger ingress soft PN expiration event.
- SP_Ingress_SA_status0: A status vector indicates which ingress SA has hard-expired.
- SP_Ingress_SA_status1: A status vector indicates which ingress SA has soft-expired.
- SP_Default_VLAN_TAG: The VLAN ID used in the special VLAN TAG when the packet has a SecTAG but no valid SC can be found. This only applies when SP_MASTER_Ctrl.Ingress_SecTAG is set to remove a SecTAG and replace it with a VLAN Tag.
- INGRESS_SPI_CONFIG: Ingress SPI or MACSECID. This register needs to be programmed when SP_MASTER_Ctrl.Ingress_SecTAG or SA.SECTAG_MODE is 1 or 2.
- SP_INGRESS_L2_REDIRECT_HEADER_0: In case of a redirect in L2_ENCAP mode (triggered by INGRESS_SA_ATTRIBUTE_TABLE action or MFP_IFP_ACTION_TABLE), this register defines the 16-bit MSB of DA field of the L2 header.
- SP_INGRESS_L2_REDIRECT_HEADER_1: In case of a redirect in L2_ENCAP mode (triggered by INGRESS_SA_ATTRIBUTE_TABLE action or MFP_IFP_ACTION_TABLE), this register defines the 32-bit LSB of DA field of the L2 header.
- SP_INGRESS_L2_REDIRECT_HEADER_2: In case of a redirect in L2_ENCAP mode (triggered by INGRESS_SA_ATTRIBUTE_TABLE action or MFP_IFP_ACTION_TABLE), this register defines the 16-bit MSB of SA field of the L2 header.
- SP_INGRESS_L2_REDIRECT_HEADER_3: In case of a redirect in L2_ENCAP mode (triggered by INGRESS_SA_ATTRIBUTE_TABLE action or MFP_IFP_ACTION_TABLE), this register defines the 32-bit LSB of SA field of the L2 header.
- SP_INGRESS_L2_REDIRECT_HEADER-4: In case of a redirect in L2_ENCAP mode (triggered by INGRESS_SA_ATTRIBUTE_TABLE action or MFP_IFP_ACTION_TABLE), this register defines the Type field of the L2 header.

- SP_Ingress_LPBK_WAIT_THR: Threshold of waiting time for ingress to take the loopback packet
- SP_Ingress_Pre_Psr_ctrl: Ingress pre-decryption parsing control register. For the Enable control registers, each rule has two bits to control with the following encoding: 0=disable, 1=Pass management packet to further processing, 2=Bypass management packet from ISEC processing, 3=Drop management packet. If the packet hits multiple rules, the packet uses the rule with the highest priority.
- SP_Ingress_L2_Sectag_Override_Cntrl: This register controls whether to replace SecTag.PN and SecTag.EtherType with special values when INGRESS_SA_ATTRIBUTE_TABLE or MFP_IFP_ACTION_TABLE redirects the packet.
- SP_Ingress_Pre_TAG_TPID: The QTAG_TPID and STAG_TPID for ingress pre-decryption parsing.
- SP_Ingress_MTU0_1: Maximum transmission unit for user priority 0 and 1.
- SP_Ingress_MTU2_3: Maximum transmission unit for user priority 2 and 3.
- SP_Ingress_MTU4_5: Maximum transmission unit for user priority 4 and 5.
- SP_Ingress_MTU6_7: Maximum transmission unit for user priority 6 and 7.
- SP_Ingress_MTU_CTRL: When set, it enables packets to be sent in the store and forward mode when it is truncated due to exceeding the MTU defined above.
- SP_Ingress_Buf_Int_EN: Ingress Buffer interrupt enable (overflow or underflow).
- SP_Ingress_STAT: Ingress interrupt status.
- SP_Ingress_Buf_Int_STAT: Ingress buffer interrupt status.
- SP_INGRESS_LPBK_CTR0: Loopback buffer control 0.
- SP_INGRESS_LPBK_CTR1: Loopback buffer control 1: Size setting.
- SP_INGRESS_LPBK_STAT: Loopback buffer status.
- SP_INGRESS_CAP_CTR0: Ingress capture buffer control 0.
- SP_INGRESS_CAP_CTR1: Ingress capture buffer control 1: size setting.
- SP_INGRESS_CAP_STAT: Ingress capture buffer status.
- SP_INGRESS_CAP_DROP_CNT: The number of packets dropped at ingress capture buffer.

SEC supports 21 rules to match incoming packets before decryption. The first two rules match the MAC DA with two hardcoded MAC addresses (0x01_80_c2_00_00_0 and 0x01_00_0c_cc_cc_cc). The remaining 19 rules are specified by the 19 registers below to match MAC DA and/or EtherType:

- Eight individually enabled MAC_DA matches. These MAC DA addresses are specified by registers SP_Ingress_Pre_MAC_DAn_LSB and SP_Ingress_Pre_MAC_DAn_MSB (n=0, 1, …, 7).
- Eight individually enabled Ethertype matches. These are specified by registers SP_Ingress_Pre_ETHERTYPE0_1, SP_Ingress_Pre_ETHERTYPE2_3, SP_Ingress_Pre_ETHERTYPE4_5, SP_Ingress_Pre_ETHERTYPE6_7.
- One programmable MAC Destination address range match. This can be specified by registers SP_Ingress_Pre_MAC_DAn_LSB and SP_Ingress_Pre_MAC_DAn_MSB (n=8 for lower bound; n=9 for upper bound).
- Two sets of programmable MAC_DA and EtherType match. This is specified by registers SP_Ingress_Pre_MACDA_ETHERTYPEn_LSB and SP_Ingress_Pre_MACDA_ETHERTYPEn_MSB (n=0, 1).

All counters can be cleared by setting global register SECY_CFG_GLB_MIB_Ctrl.CLR_OTHER_CNT to 1. If the configurable CNT_SATURATE_DIS is set, the counter rolls over to 0 when it overflows, otherwise, it remains at all 1s. If CNT_RD_CLR_DIS is not set, the counter is cleared on read, otherwise, it is not cleared on read.

Below are the main counters:

- 21 counters for rule matched management packets, 40 bits each, namely:
  - SP_INGRESS_PRE_ETYPE_MATCH_COUNTn (n=0, 1, …, 7),
  - SP_INGRESS_PRE_MACDA_MATCH_COUNTn (n=0, 1, …, 7),
  - SP_INGRESS_PRE_MAC_RANGE_MATCH_COUNT,
  - SP_INGRESS_PRE_MAC_DA_ETHERTYPE_MATCH_COUNTm (m=0, 1),
  - SP_INGRESS_PRE_MAC_DA_HARDCODED_MATCH_COUNTm (m=0, 1),
- One counter for ISEC packet drop, 32 bits, namely, SP_Ingress_Packet_Drop_Count.
- One counter for ISEC packet drop due to 2-bit ECC error, 32 bits, SP_Ingress_Packet_ECC_Drop_Count.
- 32 counters for loopback FIFO packet drop, 8 bits each, for 32 SAs, respectively, namely:
  - SP_INGRESS_LPBK_DROP_CNT_SA0_3,
  - SP_INGRESS_LPBK_DROP_CNT_SA4_7,
  - SP_INGRESS_LPBK_DROP_CNT_SA8_11,
  - SP_INGRESS_LPBK_DROP_CNT_SA12_15,
  - SP_INGRESS_LPBK_DROP_CNT_SA16_19,
  - SP_INGRESS_LPBK_DROP_CNT_SA20_23,
  - SP_INGRESS_LPBK_DROP_CNT_SA24_27,
  - SP_INGRESS_LPBK_DROP_CNT_SA28_31.
- One counter for capture FIFO packet drop, 32 bits, namely SP_INGRESS_CAP_DROP_CNT.
- Two ECC counters for ISEC, 16 bits each for 1-bit and 2-bit ECC error events, namely, SP_Ingress_ECC_Count.

- Two ECC counters for ingress SC/SA tables (INGRESS_SC_INDEX_TABLE, INGRESS_SC_ACTION_TABLE, INGRESS_SA_ATTRIBUTE_TABLE, INGRES_SA_KEY_TABLE). These two counters are contained in one register, namely, SP_Ingress_Table_ECC_Count, with 16 bits each for 1-bit and 2-bit ECC error events.
- One counter for ingress packets that do not match any rules, 40 bits, namely, SP_INGRESS_PRE_RULE_MISS_COUNT.

The special control register SP_Ingress_Pre_Psr_ctrl at ingress has two bits to control each rule. Since there are 21 rules, 42 bits in total are used. The two bits controlling each rule have the following encoding:

0 = This rule is disabled.

1 = Pass packets matching this rule for further processing.

2 = Bypass the packets matching this rule from ISEC processing.

3 = Drop the packets matching this rule.

MIB Registers

MIB registers are the standard registers.

MPORT Registers

MPORT has various registers to control thresholds for flow control.

Fixed Latency Enforcement with Loopback Packets

This block always picks up a packet from line side if the packet buffer has any packet inside. Otherwise, it is input a packet from the loopback FIFO if certain condition met. It won't switch the selection in the middle of a packet.

To meet fixed latency requirement on the line traffic:

- The packet from the line has to be delayed somewhere to compensate any jitter in the pipeline.
- The loopback packet from egress does not interfere with the packet from line. The loopback packet can only be accepted when the gap between the two line packets is big enough for it.

The existing time-stamp based approach, called timing sync or fixed latency, uses the last stage store-and-forward buffer for the delay purpose. Below is the detail.

There are two conditions need to meet for loopback packet entering the ISEC:

1. There is no packet in the MPORT's packet buffer.
2. The local_lb_en has to be 1. The basic idea is, after EOP of any line packet or after SOP of any loopback packet, the arbiter has to wait for a certain amount of time corresponding to the maximum loopback packet size before it can take next loopback packet. The EOP of a line packet or the SOP of the loopback packet clears local_lb_en to 0 and sets the timer to the threshold ISEC_LPBK_WAIT_THR given by register SP_Ingress_LPBK_WAIT_THR. The timer is decremented every cycle. When the timer reaches 0, local_lb_en is set to 1. The threshold is programmed by software by taking into account both loopback FIFO depth and the link speed. Set ISEC_LPBK_WAIT_THR = 0 to eliminate the waiting time.

UNIMAC Registers

UNIMAC registers consist of MIB registers and others when the port is configured to run at non-10 Gbps speed.

XMAC Registers

XMAC consists of various control registers when the port is configured to run at 10 Gbps speed.

Bringup Procedure

MACsec can be brought up in two test cases:

- MACsec enabled but control port disabled.
- MACsec enabled and control port also enabled.

These are high-level descriptions. For the details and pseudo-code, see Appendix A: “Pseudo Code of MACsec Registers Programming,” on page 68.

Control Port Disabled

Below are the steps to test the case when the control port is disabled but MACsec block is enabled:

1. Program SFI/Copper/SGMII and XFI/SGMII/QSGMII registers to establish the 10G/1G link. Enable Port 0 MACSEC.
2. Program line-side XMAC registers for 10 Gbps.
3. Program line-side UNIMAC registers for non-10 Gbps
4. Program switch-side/system-side XMAC registers for 10 Gbps
5. Program switch-side/system-side UNIMAC registers for non-10 Gbps.
6. Initialize MACsec functions.
7. Start sending the traffic in both directions.

Control Port Enabled

Below are the steps to test the case when the control port is enabled:

1. Program SFI/Copper/SGMII and XFI/SGMII/QSGMII registers to establish the 10G/1G link. Enable Port 0 MACSEC.
2. Program line-side XMAC at 10 Gbps.
3. Program line-side UNIMAC at non-10 Gbps.
4. Program switch-side/system-side XMAC at 10 Gbps.
5. Program switch-side/system-side UNIMAC at non-10 Gbps.
6. Initialize MACsec functions.
7. Ingress flow setup.
8. Egress flow setup.
9. Start sending the traffic in both directions.

MACsec Buffers

Both the ingress pipeline and the egress pipeline contain the store_and_forward packet buffer and the packet capture FIFO. In addition, the loopback FIFO is used for storing packets loopback from egress to ingress.

Store and Forward Packet Buffer (INGRESS_SAF_BUF/EGRESS_SAF_BUF)

The buffer is 135 bits x 2048, storing up to 32K byte packet data. Host can access it by entry based address. Software can issue command to flush out packets stored in it.

Each entry is 135 bits wide:

- bit[134] = ERR
- bit[133] = EOP
- bit[132] = SOP
- bit[131:128] = DE (data byte enable)
- bit[127:0] = packet data

The store_and_forward packet buffer can be accessed only when the pipleline is in idle state. The MORT packet buffer register (MPORT_PACKET_BUFFER) provides access to the buffers when a port is in the FC mode. In FCoE mode, the buffers can be accessed directly.

Capture FIFO (INGRESS_CAP_BUF/EGRESS_CAP_BUF)

Capture FIFO is 147 bits x 32, storing up to 512 bytes of packet data. The FIFO base unit is entry (16B).

The FIFO can be reset by software.

Maskable interrupt can be generated when at least one whole packet is in the FIFO.

There is an enable control for the FIFO to enable packets flowing into the FIFO. The control bits can be changed at anytime, with the hardware response on packet boundary.

When the enable control bits are set to 0, the host can access the FIFO in the same way as Loopback FIFO.

To support redirecting a tail failure packet to Capture FIFO in cut-through mode, all packets have to be copied here first and flushed out if no tail failure.

The size of a packet is programmable with a configurable option to drop or truncate any packet exceeding the programmed value.

The FIFO supports a configurable option to perform head drop or tail drop under overflow conditions.

One 32-bit counter counts the number of packets that are dropped due to overflow.

Each entry of the FIFO is 147 bits wide:

- bit[146] = Double-bit ECC error. This bit is read only.
- bit[145] = Indicates the packet is truncated.
- bit[144] = Indicates where the packet is being redirected from. 0 = FP, 1 = SA table.
- bit[143:142] = 2 bits of packet type
- bit[141:135] = 7 bits of match_index
  - If the packet is from ingress sa_table, this is the sa_table index.
  - If the packet is from ingress FP, this is the match_index field.
  - If the packet is from egress FP, this is the match_index field.
- bit[134:0] = The same as the store and forward buffer.
- Bits [145:135] are only valid when SOP is high.

Loopback FIFO (LOOPBACK_BUF)

Loopback FIFO is 147 bits x 32, storing up to 512 byte packet data. The FIFO depth is configurable between 4B and 512B, in 4B increments. However, entry (16B) is the base unit for the packet data, which means the entire entry is occupied even if only one byte of the entry being used.

When the FIFO is busy on internal arbitration, ISEC2esec_lb_busy is asserted and ESEC stops transmitting data at the same cycle. The FIFO full has no impact on ISEC2esec_lb_busy, instead it causes the packet to be dropped.

Each entry of the FIFO is 147 bits wide:

- bit[146] = Double-bit ECC error, this bit is read only
- bit[145] = Indicates the packet is truncated. valid when SOP valid.
- bit[144:140] = Reserved.
- bit[139:135] = Five bits of SA, valid when SOP valid.
- bit[134:0] = Same as store and forward buffer.

There is a write_enable and read_enable control for the FIFO to enable writing to the FIFO from the egress path and reading from the FIFO to the ingress path. The control bits can be changed at anytime, with hardware response on packet boundary.

When both write_enable and read_enable control bits are set to 0, software can perform read and write directly to the FIFO. There are two ways to access the FIFO.

1. Through the MDIO, each entry of the FIFO is addressable through MDIO. The software can write/read entries, but the operation does not advance the FIFO pointers.
2. Packet based read and write. The packet read is always the first packet in the FIFO and is flushed automatically upon read (that is, the FIFO read pointers are advanced). Writing a packet in this mode writes a packet after the last packet in the FIFO and advances the FIFO write pointer.

The size of a loopbacked packet is programmable with a configurable option to drop or truncate any packet exceeding the programmed value.

The FIFO supports a configurable option to perform head drop or tail drop under overflow conditions.

When a packet is dropped due to FIFO overflow, an 8-bit counter per SA is incremented.

The FIFO can be reset by software, with hardware response on packet boundary.

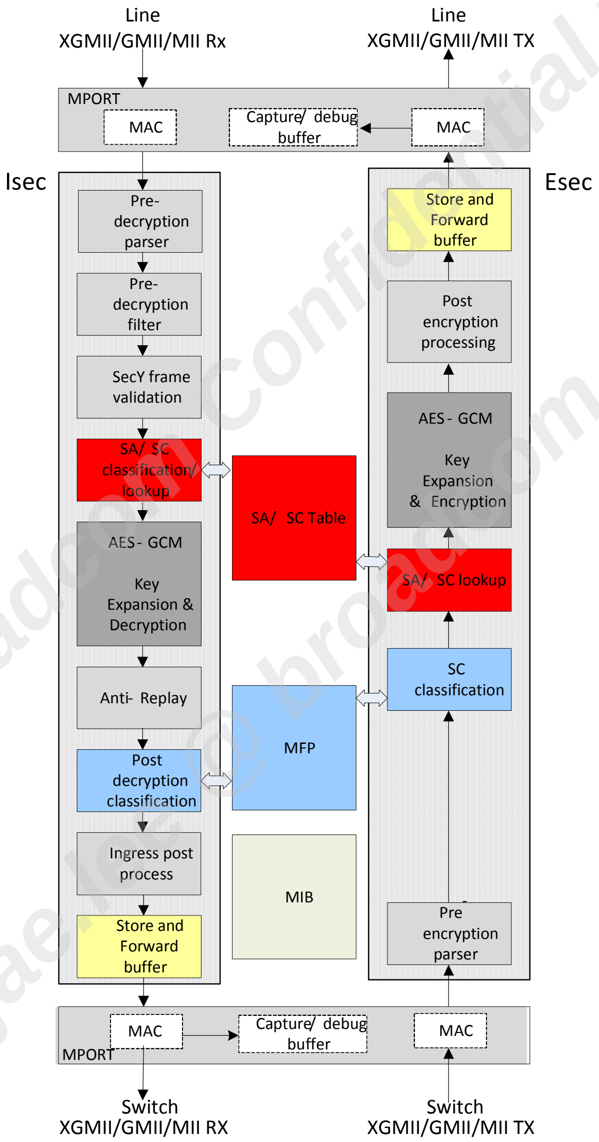
Maskable interrupt can be generated when at least one whole packet is in the FIFO.

The packet can enter the ISEC pipeline only when it is fully stored in the FIFO.

Ingress Packet Flow

incoming traffic from the line side is processed by a pre-filter that separates uncontrolled and controlled packets. MACsec packets go through SC/SA lookup which contains SAK to decrypt the packet at AES. Packets are further processed by IFP for field matching and action and then post processed before sending to switch side. This is illustrated on the left side of Figure 2.

Figure 2: Ingress and Egress Packet Flow



ISEC

The ISEC module resides in the MACsec Core and is responsible for SecY receiving functions as the packets flow from the line side to system side. The main functions of the ISEC include decryption and/or authentication of a MACSec packet according to the security settings, as well as post-decryption process such as optional redirection, drop, VLAN insertion or replacement, SecTag removal, overwriting PN with programmable ID's or status codes and MIB updating.

Figure on page 30 shows the ingress packet processing flow. Normally, a packet goes through the whole flow, but there are two exceptions:

- When ControlledPortEnabled = 0, all SecY functions are bypassed. Only bad packets such as CRC error or ECC error are dropped and all other packets pass through to switch side MAC.
- If any action from SA table is on or the packet is a management packet with drop or bypass action, the packet bypasses AES and IFP key matching logic and only goes through IFP parser to get user priority.

The ISEC consists of six major stages:

- SC/SA lookup and pre-decryption packet parsing
- AES engine
- Anti-replay check
- Post-decryption packet classification
- Store and Forward buffering
- Switch-side MAC Controller

The following sections describes detailed functions of the subblocks within the ISEC module.

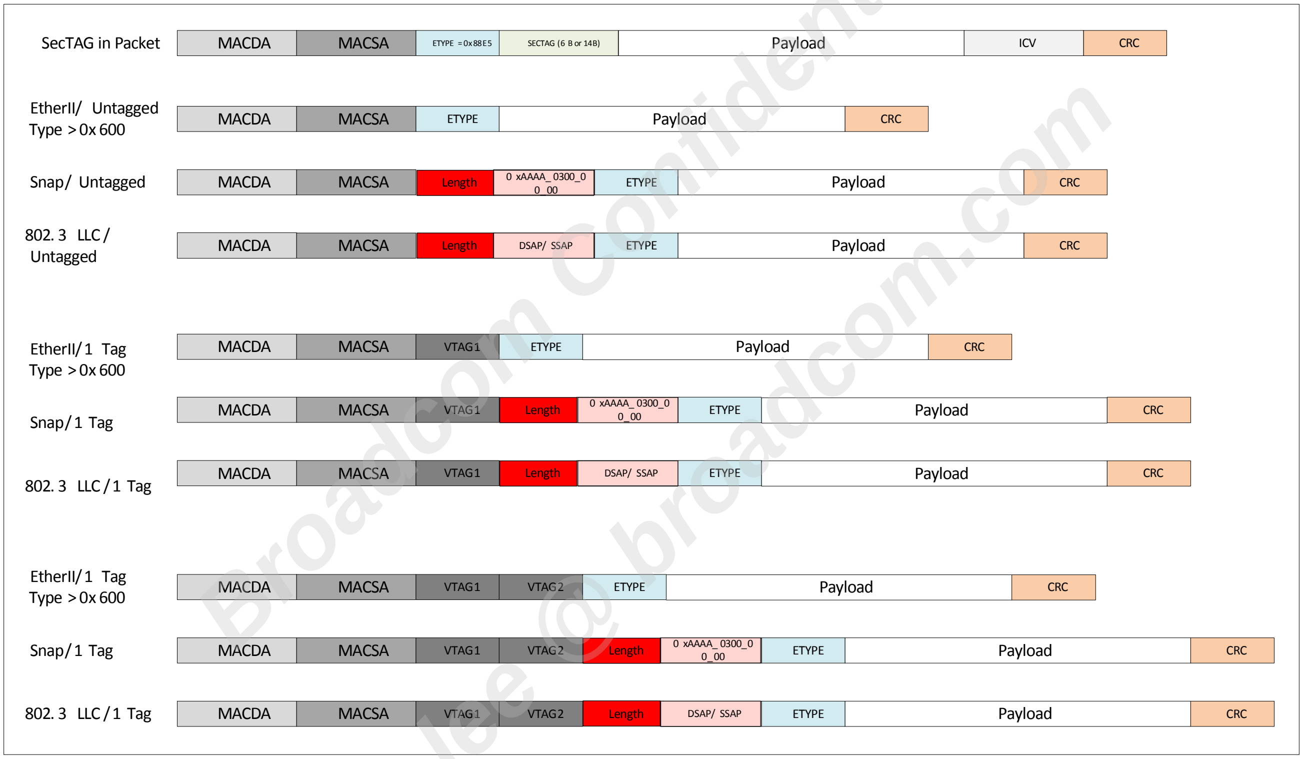
Predecryption Packet Parsing

This block is responsible for:

1. Classifying packets into three main types.
   - MACSec packets–Defined as packets with a SecTAG after the MACSA.
   - Non-MACSec packets–Defined as packets without a SecTAG after the MACSA.
   - Management packets–Packets that match the management packet types, which are defined by the 21 rules in the pre-decryption filter.
2. Parsing fields from the packet to form the SC/SA Lookup KEY.

The common packet formats that can arrive on a port are shown in Figure 3.

Figure 3: Packet Formats on Ingress Ports


SecTAG in Packet | MACDA | MACSA | ETYPE =0x88E5 | SECTAG (6 B or 14B) | Payload | ICV | CRC
EtherII/ Untagged Type > 0x 600 | MACDA | MACSA | ETYPE | Payload | CRC
Snap/ Untagged | MACDA | MACSA | Length | 0 xAAAA_0300_0 0_00 | ETYPE | Payload | CRC
802. 3 LLC / Untagged | MACDA | MACSA | Length | DSAP/ SSAP | ETYPE | Payload | CRC
EtherII/1 Tag Type > 0x 600 | MACDA | MACSA | VTAG1 | ETYPE | Payload | CRC
Snap/1 Tag | MACDA | MACSA | VTAG1 | Length | 0 xAAAA_0300_0 0_00 | ETYPE | Payload | CRC
802. 3 LLC /1 Tag | MACDA | MACSA | VTAG1 | Length | DSAP/ SSAP | ETYPE | Payload | CRC
EtherII/1 Tag Type > 0x 600 | MACDA | MACSA | VTAG1 | VTAG2 | ETYPE | Payload | CRC
Snap/1 Tag | MACDA | MACSA | VTAG1 | VTAG2 | Length | 0 xAAAA_0300_0 0_00 | ETYPE | Payload | CRC
802. 3 LLC /1 Tag | MACDA | MACSA | VTAG1 | VTAG2 | Length | DSAP/ SSAP | ETYPE | Payload | CRC

The parser recognizes two programmable TPID values, STAG_TPID and QTAG_TPID, which are specified by register SP_Ingress_Pre_TAG_TPID, and can parse up to two stacked VLAN Tags (shown as VTAG1, VTAG2). Each VLAN tag can match either of the two TPID values if enabled by register field SP_Ingress_Pre_Psr_ctrl.PARSE_STAG or .PARSE_QTAG.

In addition, the parser recognizes the special EtherType value of 0x88E5 to identify the presence of a SecTAG. If a SecTAG is present, it is always after the MACSA. The pre-decryption frame parser stops parsing the packet after the SecTAG if present and after the ETYPE otherwise. ETYPE used for LLC packets will be 0. Otherwise, the ETYPE value shown in Figure 3 is used for the other packet formats.

A packet is a management packet if any of the following matches:

- MAC_DA[47:4] == 01_80_c2_00_00_0 (highest priority). Note this address is hardcoded.
- MAC_DA == 01_00_0c_cc_cc_cc. Note this address is hardcoded.
- Eight individually enabled MAC_DA matches (all the same priority). These MAC DA addresses are specified by registers SP_Ingress_Pre_MAC_DAn_LSB and SP_Ingress_Pre_MAC_DAn_MSB (n=0, 1, …, 7).
- Eight individually enabled EtherType matches (all the same priority). These are specified by registers SP_Ingress_Pre_ETHERTYPE0_1, _Ingress_Pre_ETHERTYPE2_3, _Ingress_Pre_ETHERTYPE4_5, _Ingress_Pre_ETHERTYPE6_7.
- One programmable MAC Destination address range match. This can be specified by registers SP_Ingress_Pre_MAC_DAn_LSB and SP_Ingress_Pre_MAC_DAn_MSB (n=8 for lower bound; n=9 for upper bound).
- Two sets of programmable MAC_DA and EtherType match (lowest priority). This is specified by registers SP_Ingress_Pre_MACDA_ETHERTYPEn_LSB and SP_Ingress_Pre_MACDA_ETHERTYPEn_MSB (n=0, 1).

If a packet matches multiple rules, the packet takes the action based on the highest priority action where the priority is listed above (first one listed is the highest priority).

There are a total of 21 40-bit counters corresponding to all rule hits and one 40-bit counter for one missing all the rules. Each counter register is listed below, which matches the rules in the order as listed above, correspondingly:

- SP_INGRESS_PRE_MAC_DA_HARDCODED_MATCH_COUNT0
- SP_INGRESS_PRE_MAC_DA_HARDCODED_MATCH_COUNT1
- SP_INGRESS_PRE_MACDA_MATCH_COUNTn (n=0, 1, …, 7)
- SP_INGRESS_PRE_ETYPE_MATCH_COUNTn (n=0, 1, …, 7)
- SP_INGRESS_PRE_MAC_RANGE_MATCH_COUNT
- SP_INGRESS_PRE_MAC_DA_ETHERTYPE_MATCH_COUNTn (n=0, 1)
- SP_INGRESS_PRE_RULE_MISS_COUNT– to counter the packets that do not match any of the 21 rules.

SC/SA Lookup

SC Lookup and SA Index Generation

This block extracts the first 32-byte packet data and gets the parsing result to form SC key to look up the ingress SC table INGRESS_SC_INDEX_TABLE. If matched, retrieve the corresponding entry from the table INGRESS_SA_ATTRIBUTE_TABLE to get associated actions and SA key from the table INGRESS_SA_KEY_TABLE for AES-GCM decryption and authentication.

For non-MACSec packet, the packet.AN[1:0] is set to 0. Actions in SA table takes precedence to IFP actions.

The SA index is derived as:

```
if(sa_mode==0) // 1 SA per SC
    sa_index = sc_index;
else if (sa_mode == 1) // 2 SA per SC
    sa_index = sc_index <<1 | sectag_AN[0]
else if(sa_mode == 2) // 4 SA per SC
    sa_index = sc_index << 2 | sectag_AN[1:0]
```

In two SA per SC mode, there is also an invalid SA entry check against two AN number (AN0 and AN1) that exist in the INGRESS_SC_INDEX_TABLE. If the incoming sectag_AN is not in the SA Table or not equal to any of the AN0 or AN1 fields in the Ingress SC Table, the AN is an invalid.

The 96-bit Initialization Vector (IV) is generated using a concatenation of the SCI and PN ({SCI,PN}). If SCI does not exist in the packet, it is derived (see "SCI Generation" for more information).

SCI Generation

In the 802.1AE standard based lookup mode, the SCI is derived from the following sources:

- If the SCI is present in the packet.
- If the SCI is not present (SecTAG.SC = 0), an SCI can be derived by checking the flags in the SecTAG.
  - If SecTAG.ES=1 and SecTAG.SCB=0, then SCI = {MAC_SA, 16'h1}.
  - Else if SecTAG.SCB=1, then SCI = {MAC_SA,16'h0}.
- If neither is set and OperPointToPointMAC is set, a default SCI at SC Table Key Entry 0 is used.
- Otherwise, if none of the above is true, SCI is all 1s.

In the L2 Header lookup mode, the SCI is extracted directly from the packet if TCI.SC =1, otherwise it is set to all 0s.

AES Engine

This block implements the AES-GCM core and the interface to it. A MACsec packet is decrypted and authenticated here if needed. ICV is also removed except some error packets when ValidateFrame is in strict mode or SecTAG.TCI.C==1. FIPS testing logic is implemented here and can be accessed through host register access.

The packets coming from the ISEC lookup stage are decrypted if not indicated to be bypassed. The non-bypassed packet's ICV is calculated over MAC addresses, SecTAG (8 or 16 byte) and the packet data.

The AES engine gets BYPASS_AES and AUTHENTICATION_ONLY indications, packet data, and an AES key. It outputs the decrypted packet and updated packet header information including ICV check result to the down-stream block of ISEC_rpy.

AES10G core is implemented for a 10 Gbps port, while AES1G core is implemented for a 1 Gbps port.

Antireplay Check

This block performs anti-replay check and updates the corresponding field of the SA table. It outputs the packet data and updated packet header information, including anti-replay check results to the down-stream block of IFP.

To update the SA table, this block performs a read-modify-write operation. If the software writes the same entry of SA table after the hardware read, but before hardware write, the hardware write is aborted.

The Ingress anti-replay protection is done post-decryption when the ReplayProtect is enabled. A 32-bit programmable replay window size is specified on a per SA basis. It checks if the PN number is less than SA.NextPN – ReplayProtectWindow to determine if the packet is a replayed packet.

Normally, when a valid packet is received with a PN equal to or greater than the NextPN field stored in the SA descriptor, the NextPN field is updated with Packet.PN + 1.

When SA.Next_PN reaches 0xFFFFFFFF, a configuration register field SP_MASTER_CTRL IngrRpyChkInvalidateSaEn is introduced to determine if the SA invalidation is done automatically by the hardware. If IngrRpyChkInvalidateSaEn =1 and SA.Next_PN hits 0xFFFFFFFF, the SA is invalidated by the hardware. If IngrRpyChkInvalidateSaEn =0, the SA does not expire until manually done by software.

The KaY entity can invalidate the SA at anytime by installing a new SA.

The egress loopback packet still goes through anti-replay check and updating SA table procedure.

The NextPN field is updated regardless of packet action as well as start and stop timer except for the following cases:

- SL>=48
- minimum length failed
- SA invalid,
- SC miss
- untagged packet
- V !=0
- SC and SCB set

- PN=0
- SC and ES set
- ICV fail

SecTAG Modification Policy

Since ISEC anti-replay does an SA Table lookup, the SECTAG_MODE field is added to the header information and passed downstream through the pipeline where it is used in this stage.

SA table contains two sets of control bits to allow modification of the SecTAG if one exists in a packet. The SECTAG_MODE controls whether to do SecTAG preservation, removal or VLAN replacement on a per-SA basis, The SecTAG preservation action options include replacing both SecTAG ETYPE and PN or replacing the PN with MACSECID. The default is no SecTAG removal or no change.

Postdecryption Packet Classification

This block provides the interface to the IFP module. It provides the first 64-byte packet data plus packet information based on the packet header from ISEC anti-replay to IFP module, and receives the packet's post actions from IFP. Packet data and updated packet header information are passed to its down-stream block of ISEC SAF.

IFP provides a bunch of actions retrieved from MFP_IFP_ACTION_TABLE if matching IFP key table. If there are multiple actions from the different stages, management packet action takes the highest priority, then followed by SA table action and IFP action.

Store and Forward (ISEC SAF)

This block is responsible for the following:

- Receiving packet data and header information from IFP.
- Outputting packet data with updated header information.
- Based on configuration, storing and forwarding packets before transmit or pass packet data as soon as received (cut-through mode).
- Labeling the packet as drop if drop criteria met.
- Flushing out packets stored in SAF buffer based on software control.

Switch-Side MAC Controller

This block is responsible for:

- Transmitting the packet to switch side MAC or redirecting/copying the packet to capture FIFO.
- Removing SECtag if SECTAG_MODE is to remove SecTAG.
- If SA Attribute Table SECTAG_MODE action is to replace EtherType (and not a redirect packet), replacing SecTAG EtherType with the value programmed in Special EtherType Register and PN with {16'h0,MACSECID[15:0]}.
- If the SA Attribute Table SECTAG_MODE action is to replace PN (and not a redirect packet), PN is replaced with {16’h0, MACSECID[15:0]}. MACSECID comes from Ingress SC Action Table. When SECTAG_MODE in SA Attribute Table is set to VLAN replacement, SecTAG will be replaced with the ST-VLAN tag. The Flow ID field in ST-VLAN Tag will come from Ingress SC Action Table too.
- L2 header insertion for L2 redirection. If the configuration register bit SECTAG_ETYPE_OVRRIDE_EN is set, the MACSec EtherType is replaced with the value programmed in the Special EtherType Register. The PN is replaced with {ERR_STATUS[15:0], 16’h0}. ERR_STATUS definition is defined in Table 4: “ISEC PN Replace Error Status Codes,” on page 39. Otherwise the packet is encapsulated with new L2 header (MAC DA,SA, EtherType) when SECTAG_ETYPE_OVRRIDE_EN=0.
- When SECTAG_MODE is set to keep original SecTAG, the SecTAG original contents are preserved.
- Enforcing the maximum packet size check per user priority (total eight UPs):
  - Configurable MTU per UP.
  - MTU check is based on line-side receive packet length.
  - Store and forward mode drops on MTU failure, except when send_on_saf_trunc_en is set, which allows truncated packets to be sent when this happens. Cut-through mode always truncates packets. The truncated packet length may not reflect the MTU length because ingress packet processing may strip or add bytes. In store and forward mode as well as and cut-through mode, if a truncated packet is being sent, the CRC is always corrupted.
  - Since ingress packet processing may remove or add bytes to the original line-side received packet, the packet sent out the switch-side length may not reflect MTU packet length. However, all received line-side packets that violate MTU are reported as either dropped packets or have its CRC corrupted. Table 2 on page 38 describes the cases to consider when packets are sent out the switch side on MTU violation.

Table 2: Switch Side Behavior on MTU Violation When Packets Sent

Case	Length of Packet	Error Action
MTU len < (SW len = Line len)	MTU len	Truncated, CRC corrupted
SW len < MTU len < Line len	SW len	CRC corrupted
MTU len < SW len < Line len	MTU len	Truncated, CRC corrupted
MTU len < Line Len < SW len	MTU len + SW len – Line Len	Truncated, CRC corrupted
Line Len< MTU len < SW len	SW len	None

- Actual removal of a packet by physically drop it or corrupt CRC.
- Provide SMIB and user-priority MIB vectors.

The ST-VLAN tag for ingress is capable of indicating packet categories, error status as well as preserving certain information that is carried in the SCI of the original packet. The format of the ingress ST-VLAN tag is defined in Table 3.

Table 3: ST-VLAN Tag Format

Field	Bits	Description
TAG Protocol ID	31:16	TPID to indicate this is a ST-VLAN tag for SecY.
Status	15:12	4’b0000–Good packet. [E,C] = 00. 4’b0001–Good packet [E, C] = 01. 4’b0010–Reserved. 4’b0011–Good packet [E, C] = 11. 4’b0100–Bad packet – invalid SecTAG. 4’b0101–Bad packet- no SCI or SA found. 4’b0110–Bad packet – replay check failed. 4’b0111–Bad packet – ICV check failed. 4’b1xxx–Indicates the packet came in with a VLAN tag that matches the TPID specified for the ST-VLAN tag or the packet is an unprotected packet
Flow ID	11:0	12-bit value from the ingress SC action table or Default VLAN tag register.

Table 4: ISEC PN Replace Error Status Codes

Status	Description	Supported Mode	ErrCode
NoError	Successful Decrypted Packet.	Cut-through, Store-and-Forward	0
VersionChkFail	SecTag version check failure bit.	Cut-through, Store-and-Forward	1
TciChkFail	Illegal TCI combination failure bit.	Cut-through, Store-and-Forward	2
IllegalSL	Illegal Short Length which is By SL byte value in SecTag is equal or larger than 48.	Cut-through, Store-and-Forward	3
bindingFail	Fail to find the SC for the SecTag.	Cut-through, Store-and-Forward	4
ReplayChkFail	Replay check failure bit.	Cut-through, Store-and-Forward ***Note:*** In Cut-through mode, Replay fail (early replace check fail) can be set in PN.status but the nextPN update has to wait after ICV validated. Replay check fail only happens when replay check is performed.	5
AuthChkFail	Authentication Failure.	Store-and-Forward ***Note:*** For Cut-through device, MACsec device corrupts CRC for AuthChkFail by bitwise inversion of the correct CRC.	6

ISEC Operation

Reset and Clock

ISEC runs in system clock domain.

Reset is already synchronized to the clock.

Configuration

To assure ISEC functions correctly, the software must configure the following tables and registers

- Tables
  - Ingress SC table
  - Ingress SA table, including attribute and key
  - IFP key table and action table
  - Ingress SC action table
- Registers
  - MASTER_Ctrl
  - Ingress_Ctrl
  - Ingress_PN_THD
  - Default_VLAN_TAG
  - INGRESS_SPI_CONFIG

- L2 redirect header related registers
- Packet type determination related registers
- Pre-decryption parser related registers
- User-priority MTU related registers

Memory Access

The host can access the following packet buffers in ISEC.

- Store and forward packet buffer: Only supports address-based access (see "ISEC Operation" on page 39).
- Loopback FIFO: Supports both address-based access and packet-based access (see "ISEC Operation" on page 39).
- Capture FIFO: Supports both address-based access and packet-based access (see "Capture FIFO (INGRESS_CAP_BUF/EGRESS_CAP_BUF)" on page 28).

Error Detection and Handling

ISEC has ECC error detections for internal memories. Each memory has two ECC interrupts: one for single bit ECC error and the other for double bits ECC error.

ISEC also has two counters for all memories' ECC errors: one is for all 1-bit error and the other is for all 2-bit errors.

Once a double-bit ECC error occurs, the corresponding packet is dropped, otherwise, the CRC is corrupted. Below lists sources that may cause the 2-bit ECC error for an ingress packet:

- Packet data ECC error caused by MPORT's packet buffer.
- Packet data ECC error caused by FP packet data FIFO.
- Packet data ECC error caused by SAF packet data FIFO.
- Packet data ECC error caused by loopback packet data FIFO.
- Packet header ECC error caused by SAF header FIFO.
- Packet header ECC error caused by SA table.
- Packet header ECC error caused by SC table.

In addition, the capture FIFO may cause an ECC error. However, the packet in capture FIFO is only read by host, the ECC error won't result in a drop of the packet. Instead, the ECC error status goes with the packet data while the host reads the packet.

Ingress MFP (IFP)

All receiving packets after SecY processing can be matched against an Ingress Packet Classification Engine called IFP. This includes both traffic identified as Uncontrolled and Controlled. IFP is organized similarly to EFP. As described earlier, the Key Table CAM and the Action Table RAM are shared between the IFP and EFP although the content of the memories are organized differently.

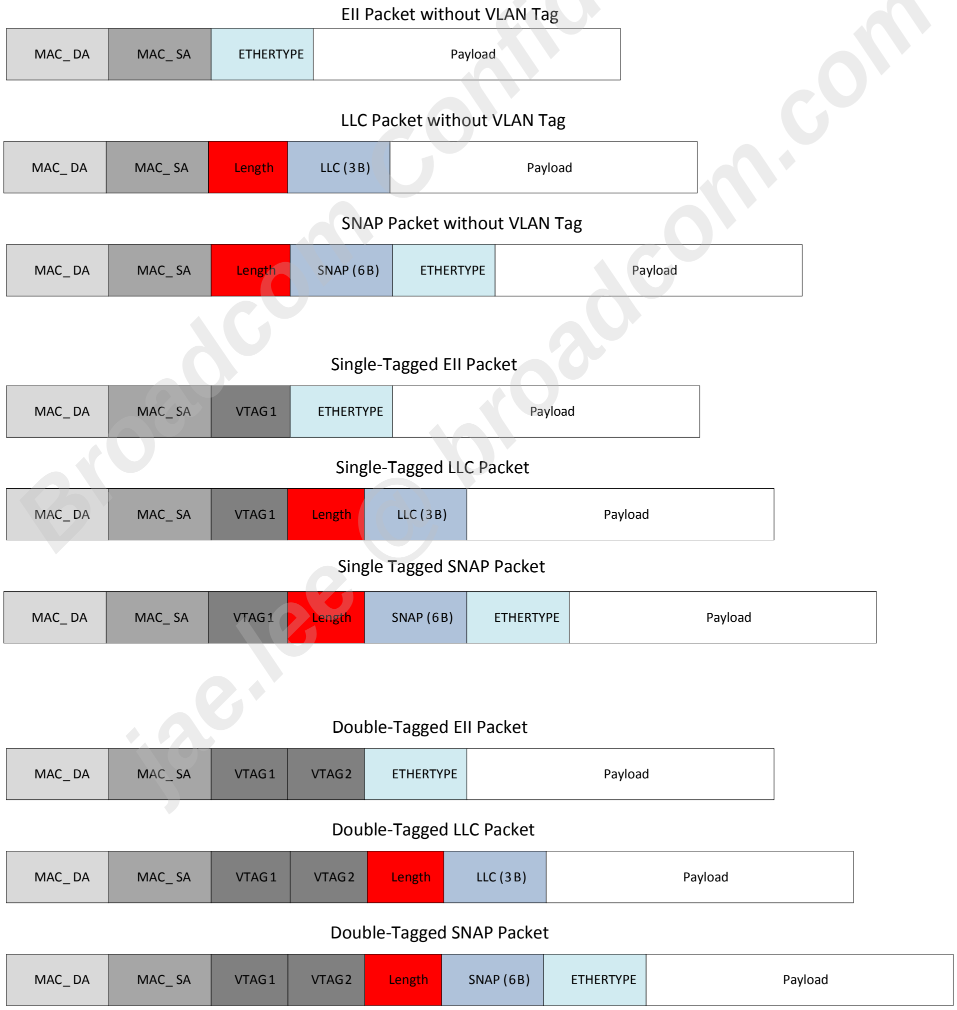
IFP IBUF

IFP IBUF is a pipe to receive the request from the ISEC and store the packet data in a temporary buffer before the EOP data is received. Only 64 bytes of packet data are sent to IFP to save the buffer space, which can cover the longest packet header information required by the IFP parser. A signal called ISEC2fp_up_only is added for IFP to extract the user priority for per-UP MTU check and per-UP RMON MIB purpose. This signal is asserted by ISEC when ControlledPortEnabled=0, or ISEC_parser action is DROP or BYPASS, or SA table actions are DROP/REDIRECT/DO_NOT_MODIFY/DROP_IF_NOT_LB/REDIR_DBUG_FIFO.

IFP Parser

The IFP Parser is a robust L2 header parsing module that supports the packet data from less than 16B to 64B. It starts the packet parsing whenever the packet SOP data is received. The supported packet formats are shown in Figure 4.

Figure 4: IFP Supported Packet Formats


EII Packet without VLAN Tag
MAC_DA | MAC_SA | ETHERTYPE | Payload
LLC Packet without VLAN Tag
MAC_DA | MAC_SA | Length | LLC (3B) | Payload
SNAP Packet without VLAN Tag
MAC_DA | MAC_SA | Length | SNAP (6B) | ETHERTYPE | Payload
Single-Tagged EII Packet
MAC_DA | MAC_SA | VTAG1 | ETHERTYPE | Payload
Single-Tagged LLC Packet
MAC_DA | MAC_SA | VTAG1 | Length | LLC (3B) | Payload
Single Tagged SNAP Packet
MAC_DA | MAC_SA | VTAG1 | Length | SNAP (6B) | ETHERTYPE | Payload
Double-Tagged EII Packet
MAC_DA | MAC_SA | VTAG1 | VTAG2 | ETHERTYPE | Payload
Double-Tagged LLC Packet
MAC_DA | MAC_SA | VTAG1 | VTAG2 | Length | LLC (3B) | Payload
Double-Tagged SNAP Packet
MAC_DA | MAC_SA | VTAG1 | VTAG2 | Length | SNAP (6B) | ETHERTYPE | Payload

ISEC will strip the SecTAG in the packet header before it transfers the packet data to IFP. In case the SecTAG is not stripped, IFP parses the packet header ahead of the SecTAG. Any packet parsing from the SecTAG of the packet becomes unknown.

The IFP Parser supports single VLAN Tag or Double VLAN Tag packets. Three Outer TPID values (TPID1, TPID2 and TPID3) and one Inner TPID value are defined to identify the outer and inner VLAN Tags. TPID1 corresponds to QTAG, and TPID2 corresponds to STAG. Both of them can be used as inner VLAN Tags when Q-in-Q is enabled. One special TPID value is also defined to identify the special VLAN Tag which may carry proprietary information for the special applications. The special VLAN Tag is always treated as one of the Outer VLAN Tags. VTAG1 is specified by one of the three ingress outer TPID registers, or the ingress special TPID register for the special VLAN Tag. With all outer VLAN Tags, the priority is from higher to lower in the order of TPID2, TPID1, TPID3, and Special_TPID. VTAG2 is specified by the ingress inner TPID register or TPID1 and TPID2 registers. QTAG_TPID and STAG_TPID is programmed to TPID1 and TPID2 registers, respectively. If q-in-q is disabled for double tag checking, the EtherType of the packet is the TPID of the VTAG2. The parser encodes the final VLAN Tag parsing result to packet_vlan_tag_format defined in Table 12: "Packet_VLAN_Tag_Format Encoding," on page 56.

Besides the TPID enable bit for each TPID register, there is a user priority (UP) enable bit for each TPID register except the inner TPID register. If VTAG1 matches one of the TPID registers and UP is enabled for the TPID, the PRI bits are extracted from VTAG1 and mapped to the configurable value if this TPID corresponds to a QTAG_TPID (i.e.TPID1). If VTAG1 doesn't match any of the programmed TPIDs or the matched TPID doesn't have UP enabled, a programmed default UP is used as the UP of this packet.

The IFP Parser decodes the two bytes of data followed by the VLAN Tag. If it is larger than or equal to a configurable length value for EII, the packet is classified to be an EII packet. The two-byte data is the EtherType. Otherwise, it is a length field of the LLC/SNAP packet. The LLC header and SNAP header are the same format as defined in Table 13: "LLC Header Format," on page 57 and Table 14: "SNAP Header Format," on page 57.

Any packets with DSAP/SSAP/Control that do not match the specified value in SNAP header are classified to the LLC packets. The EtherType of LLC packets will be {DSAP, SSAP}.

The IFP Parser decodes the MAC Control frames with EtherType 0x8808. The decoding result is passed to ISEC for MIB counter update.

One separate four-byte UDF is supported. The UDF which is defined by INGRESS_UDF as parsed from the first 64 bytes of the packet starting at the beginning of the packet.

IFP Key Selection

The IFP lookup key consists of the 234-bit fixed fields, which consist of the packet validation status, the L2 fields parsed from the packet and the user defined fields (UDF) extracted at the configurable offsets from the first 64 bytes of the packet. The 234-bit MFP_IFP_KEY_TABLE lookup key is defined as follows:

Table 5: Lookup Key of MFP_IFP_KEY_TABLE

Field Name	Bits field	Description
direction	233	0:ingress 1:egress
Port_num	232:229	Reserved.
SecTAG_Status	228:227	SecTAG status. 0b00: frame does not contain a SecTAG 0b01: frame contains a SecTAG Others: Reserved
Frame_format	226:225	Type of Ethernet frame 0:Ethernet II packet (LENTYPE>=CP_ETYPE_MAX_LEN) 1:SNAP packet (aa-aa-03-00-00-00) 2:LLC packet (LENTYPE < CP_ETYPE_MAX_LEN and !SNAP) 3:Reserved
Vlan_tag_status	224:221	Type of VLAN Tags found on frame 0x0:untagged packet 0x1:Single VLAN Tag. It's Inner Tag 0x4:Single VLAN Tag. It's ST-VLAN Tag 0x5:Single VLAN Tag. It's Outer TPID1 (i.e.QTAG) 0x6:Single VLAN Tag. It's Outer TPID2 (i.e.STAG) 0x7:Single VLAN Tag. It's Outer TPID3 0x8:Double VLAN Tag. Outer Tag is ST-VLAN Tag 0x9:Double VLAN Tag. Outer Tag is Outer TPID1 (i.e. QTAG) 0xa:Double VLAN Tag. Outer Tag is Outer TPID2 (i.e.STAG) 0xb:Double VLAN Tag. Outer Tag is Outer TPID3 others: Reserved
Pkt_type	220:219	Type of packet 00–Non-MACsec packet 01–MACsec packet 1x–Management packet (matches in pre-filter table)
Reserved	218:217	–
Egr_ingr_lb_bit	216	Indicates the packet is loopbacked from egress direction

Table 5: Lookup Key of MFP_IFP_KEY_TABLE (Cont.)

Field Name	Bits field	Description
Security_Status	215:208	0x1: controlled_port_packet. Indicate the controlled port packet 0x2: replay_fail. Replay failed error 0x4: SA miss. SA miss error 0x8: SC miss. SC miss error 0x10: VersionChkFail. SecTAG version check failure 0x20: TciChkFail. Illegal TCI combination failure 0x40: IllegalSL. Illegal Short Length which is by SL byte value in SecTAG is equal to or larger than 48. 0x80: IllegalST. Illegal SECTAG (OR of IllegalSL, TciChkFail, VersionChkFail, PN=0)
Reserved	207:205	–
SA_index[4:0]	204:200	Ingress SA Table index
Reserved	199:184	–
INNER_TAG	183:168	Inner VLAN Tag (i.e.PRI+CFI+VID)
UDF	167:136	The 4-byte user-defined field. It's defined by register Ingress_UDF.
Reserved	135:132	–
SecTAG.C_E	131:130	SecTAG.TCI C and E bit, i.e.{SecTAG.TCI.C, SecTAG.TCI.E}
Reserved	129:128	–
ETHERTYPE	127:112	EtherType for Ethernet II/SNAP packets. It's DSAP+SSAP for LLC packets
OUTER_TAG	111:96	Outer VLAN Tag (i.e.PRI+CFI+VID)
SA	95:48	MAC source address
DA	47:0	MAC destination address

MFP Search Engine Arbiter

Since IFP shares the same search engine as EFP, the MFP Searching Arbiter performs the round-robin algorithm to select IFP lookup key request and EFP lookup key request. The winner is marked with a label in port ID to the MFP Search Engine. The MFP Search Engine takes different actions depending upon the source of the lookup key request. After the MFP Search Engine generates an action for the lookup request, the MFP Searching Arbiter distributes it to IFP_EBUF if IFP is the winner. Otherwise, it is sent to EFP_EBUF.

MFP Search Engine

The MFP Search Engine consists of a TCAM, which implements both MFP_EFP_KEY_TABLE and MFP_IFP_KEY_TABLE. The MFP_IFP_KEY_TABLE is defined as follows:

Table 6: MFP_IFP_KEY_TABLE Definition

Field Name	*Bits Field*	*Description*
KEY_VALID	468	It indicates this entry is valid. 1:valid 0:invalid
KEY_MASK	467:234	Per-bit key comparison enable. 1:compare 0:ignore
KEY	233:0	234-bit lookup key generated by IFP Key Selection

The TCAM size is 128x234, with a 234 bit lookup key, 234 bit key mask and a valid bit to indicate whether this entry is valid or not. Even though MFP_EFP_KEY_TABLE and MFP_IFP_KEY_TABLE have different definitions, they share the same TCAM. The software has the flexibility to allocate the dynamic memory space for the MFP_IFP_KEY_TABLE and MFP_EFP_KEY_TABLE, respectively. If the lookup key matches any of the entries in the TCAM, the packet is classified as belonging to the flow defined by the entry.The action associated with the flow is applied to the packet. If more than one entry matches, the first match is picked up.

The IFP_ACTION_Table is defined as follows:

Table 7: IFP_ACTION_Table Definition

Field Name	Bits Field	Description
POLICY	[31:29]	3’b000–Block/drop packet. 3’b001–Drop if packet ICV failed. 3’b010–Do not drop if packet failed. 3’b011–Add a 4 byte special VLAN Tag with VLAN ID =0, and {CFI, PRI} = {2’b11,E,C} (This action can only apply on Unprotected packets). 3’b100–NO-OP. 3’b101–Redirect. L2_ENCAP mode will be used to encapsulate the packet with a new L2 header (defined in register SP_INGRESS_L2_REDIRECT_HEADER_*). A per-port configuration register, SP_Ingress_L2_Sectag_Override_Cntrl.SECTAG_ PN_OVRRIDE_EN, will determine whether or not to update the SecTAG.PN with Error Status Code. If yes, PN MSB will be replaced by Status Code and the LSB will be replaced by 16-bit zeros. 3’b110 –Redirect if the packet failed ICV checks. L2_ENCAP mode will be used to encapsulate the packet with a new L2 header (defined in register SP_INGRESS_L2_REDIRECT_HEADER_*). A per-port configuration register, SP_Ingress_L2_Sectag_Override_Cntrl.SECTAG_ PN_OVRRIDE_EN, will determine whether or not to update the SecTAG.PN with Error Status Code. If yes, PN MSB will be replaced by Status Code and the LSB will be replaced by 16-bit zeros.
DROP_IF_NOT_LB	[28]	Drop packet if LB bit not set.
REDIR_DBUG_FIFO	[27]	Redirect to Debug capture FIFO.
COPY_DBUG_FIFO	[26]	If set, copy the packet to debug FIFO. Optionally flush packet at EOP depending on packet errors.
MATCH_INDEX	[25:19]	To be carried in the reason code to DEBUG FIFO.
BYPASS	[18]	Do not modify the packet.
Reserved	[17:0]	Reserved bits.

As ISEC needs the user priority (UP) for per-UP MIB counters and per-UP MTU check, the MFP Search Engine puts UP decoded from the packet VLAN Tag in the reserved bits 3:1 of the action data.

If IFP_ACTION_Table has a double-bit ECC error, the MFP Search Engine uses the reserved bit 0 in the action data to notify ISEC to treat the packet as a failed packet. The packet is dropped like CRC error packets.

IFP EBUF

IFP EBUF receives the action data from the MFP Search Engine Arbiter. The action data is sent to ISEC with MFP_IFP_KEY_TABLE match or mismatch information. If it is a match, the action_data is valid. Otherwise, the packet is treated as an IFP mismatch packet, the programmed action in SP_MASTER_Ctrl register provides the default behavior of the packet post-processing.

IFP Register

The IFP register implements all IFP registers. It receives the request from MFP and acknowledges when the read/write operation is completed.

In addition, the IFP register implements ISEC2fp_up_only packet counter and MFP_IFP_KEY_TABLE miss counter. The miss counter does not include the miss caused by ISEC2fp_up_only packets.

Egress Packet Flow

The ESEC module contains the following pipe stages and functions as the packets flow from system side to the line side:

- Classification action of system side packets
- Secured Channel/Security Association Lookup
- AES Engine
- Line Side MAC processing
- Store and Forward Buffer
- Debug Capture Buffer
- Configuration Register
- FIPS testing interface
- MIB interface

Classification Action

This subfunction is responsible for:

- Receiving packets from the system side MPORT block and forward directly to EFP block as requests. Requests are less than the first 64 bytes of the packet.
- Storing the packet header into ESEC FP FIFO until responses are received from the EFP.
- When EFP responses are returned, read out packet header of FP FIFO and send packet header and EFP responses to the SC/SA lookup subfunction unless there is a delete packet condition (FP remove outer VLAN action).
- The global configuration setting in the ISEC register SP_MASTER_Ctrl.ControlledPortEnabled is set to 0 here to direct all traffic as uncontrolled.

Packet Action Priority

The following table describes actions taken for uncontrolled and controlled traffic and how the logic uses the management packet action field in the EFP response.

Table 8: Packet Action Priority

ControlledPort Enabled	*EFP Hit/Miss*	*EFP Management Packet Action*	*ESEC Action Taken*
Yes	Hit	Pass or Miss	Follow EFP response policy and action.
Yes	Miss	Pass or Miss	Packet is considered uncontrolled. Use UnmatchedPacketPolicy register to decide if drop.
Yes	Don’t Care	Drop/Bypass	Packet is considered uncontrolled. Action of drop will drop the packet.
No	Don’t Care	Pass/Bypass or Miss	Packet is uncontrolled, look at user priority and packet_has_sectag fields in EFP response.
No	Don’t Care	Drop	Packet is considered uncontrolled and is dropped.

SC/SA Lookup (esec_lkup)

The esec_lkup subfunction is responsible for:

- Using the EFP response, perform SC table lookup information based on the SCI_INDEX.
- Deleting SecTag bytes if the incoming packet had a sectag in it.
- Using the SC Table AN field, looks up the SA information which contains key information to encrypt egress packets.
- Sets AES bypass signal based on drop or uncontrolled packet classification from the EFP.
- The SA (secured association) index is determined based on the AN field. The SA index is used to lookup from the XTABLE block the SA information that contains the key to use for encryption.
- Generate expire indication when PN reaches max or threshold value programmed in egress PN expire threshold register.
- Add a per-port configuration register UNCONTR_STAG_STD_ETYPE_EN for the uncontrolled_port packets. If this register is set, the EtherType of the SecTAG packet is replaced with a standard value:0x88e5 for the uncontrolled_port packets. FP parses the packet header and indicate in its response that the packet has sectag.

- Determine ES, SCB, SC etc. SecTAG fields in terms of the EFP response.
- The esec_lkup contains a read interface with a read request and acknowledge handshake to the XTABLE block that returns data described in Table 9: "SA Lookup Result .Fields," on page 49.
- The 6-bit address to the table is derived from the SC_INDEX and AN field obtained from the read of the SC TABLE:
  - If SC to SA one-to-one mapping:
    SA TABLE address = SC_INDEX[3:0];
  - If SC to two SAs:
    SA TABLE address = {SC_INDEX[3:0], AN[0]};
  - If SC to four SAs:
    SA TABLE address = {SC_INDEX[2:0], AN[1:0]};

The SA TABLE is 32 entries by 256 bits. Physically, the Egress SA TABLE is 64 x 128 bits. The esec_lkup performs two reads to get one SA TABLE entry. Even addresses get the AES Key information and odd addresses get the SA attribute information. Any double bit ECC errors that occur during this lookup is passed along in the header to the AES stage.

Table 9: SA Lookup Result .Fields

Field	Bit	Definition
Status[1:0]	[127:126]	00- invalid, 01- Reserved, 10- Valid and fresh, 11- Valid and in use
Reserved	[125:123]	
PNControl	[122:121]	0- Normal Mode, Invalidate SA when NextPN = 0xFFFFFFFF. 1- Rollover mode, increment NextPN from 0xFFFFFFFF to 0x1.
LB_TO_INGRESS	[120]	If set to one-loop packet back to ingress
ProtectionOffset[7:0]	[119:112]	The offset to apply confidentiality protection from the start of the MSDU
Reserved	[111:97]	
NextPN[31:0]	[95:64]	The next packet sequence number
SA Started Time[31:0]	[63:32]	The system time when this SA last started transmitting packets
SA Stopped Time[31:0]	[31:0]	The system time when this SA last stopped transmitting packets
SA Key[127:0]	[127:0]	SA Key obtained on odd addresses

AES Engine

The esec_aes subfunction is responsible for:

- Perform AES encryption based on the information provided by the esec_lkup subfunction.
- Provide an interface for AES FIPS testing.
- Provide buffering for headers and allow host access.
- Insert Sectag Bytes.
- Calculate and insert ICV bytes.
- Strips any padding bytes.

AES Engine core supports AES-GCM with 128 bit data path and can support 10 Gbps throughput.

Store and Forward Buffer (esec_saf)

The esec_saf is responsible for the following:

- Receive packet header and data information from esec_lmc subfunction.
- Transmitting packets to line side MAC (MPORT block).
- Based on configuration, store and forward packets before transmit or pass packet data as soon as received (cut-through mode).
- Provide MIB with information vector about egress traffic.
- Redirect packets to esec_dbg_buf block if indicated in the header information.
- Provide access to header and data packet buffers.
- Provide drop packet strobe for drop packet counters due to software pipe flush or TXERR or TX_CRC_ERR.
- Report ECC errors in packet and header buffers.
- Allow software control of flushing out pipe.

MTU Check and Behavior

Each user priority has an MTU check controllable by eight separate 16-bit MTU check registers SP_MTUn (n=0, 1, ..., 7). To disable this check, the software can program this MTU value to its maximum value. The MTU check logic always checks and truncates any packet transmitted that exceeds the allocated value for its user priority. For Cut-through mode, when the packet is truncated, the CRC is always corrupted. For SAF mode, the truncated packet is optionally sent based on a control register bit (SAF_SEND_ON_TRUNC_EN) with CRC corrupted, otherwise, the packet is dropped.

The MTU Check is performed on all traffic, including uncontrolled traffic.

Egress Packet Processing

This includes SecTAG generation and insertion, short packet handling, packet encryption and authentication with ICV insertion.

For packets that do not match in the EFP, it is either sent out unmodified or dropped, depending on a per port configurable option set by SP_MASTER_Ctrl.UnmatchedPacketPolicy. If it matches an EFP entry, but the policy indicates drop or uncontrolled bypass, it is processed accordingly. For all other packets, packet modifications are done as described in the sections below.

Four types of packet formats are expected to be generated by the system-side and fed into the MACsec PHY device for egress MACsec operation in addition to management traffic.

Native Packet Processing

For a Native Packet, if bit[31] from the corresponding matched entry at the EFP action Table (MFP_EFP_ACTION_TABLE) is set, the packet needs to be encrypted and a SecTAG and ICV inserted into the packet. The EFP action table provides the SCI_Index which selects the outgoing SC. The packet’s C and E bits come from the EFP action table. The protection mode indicated by the C and E bits are checked against the security policy of the SC, stored in EGRESS_SC_INDEX_TABLE.CIPHER_SUITE_PROTECTION, to determine if the packet is encrypted or authenticated only.

The SecTAG is formed according to the ACTION field from the EFP action Table, and the fields in the SC table EGRESS_SC_INDEX_TABLE. The SC table provides the 64-bit SCI and indication whether to include the SCI in the SecTAG. The ACTION field also indicates whether to override the lower 12-bits of the SCI with a 12-bit VID from the Outer or Inner VLAN TAG.

The resulting 8- or 16-byte SecTAG is inserted into the packet. The SL is calculated in terms of the packet payload length.

Special VLAN Tagged Packet Processing

The C and E bits are either taken from the Special VLAN Tag or the EFP action table, depending on bit[30] in the table. The mode indicated by the resulting C and E bits are checked against the Cipher_Suite_Protection capability of the SC, stored in the SC table.

The SecTAG is formed according to the Policy and ACTION fields from the EFP action table, and the fields in the SC table EGRESS_SC_INDEX_TABLE. The SC table provides the 64-bit SCI and indicates whether to include the SCI in the SecTAG. The ACTION field indicates whether to override the lower 12-bits of the SCI with a 12-bit VID from the inner or outer VLAN Tag.

The resulting 8- or 16-byte SecTAG is inserted into the packet after MAC SA. The SL is calculated in terms of the packet payload length.

The Special VLAN Tag can be optionally removed.

SecTAG Tagged Packet Processing

The packet processing is similar to the Special VLAN Tag case, except the incoming packet's SecTAG may contain additional SL information, and some SecTAG fields may come from the incoming packet instead of the SC table and EFP action table. The field SecTag.SL from incoming packet indicates where the 16-byte ICV field is in the outgoing packet (the ICV field is not carried in the packet):

- When the SL=0, the ICV calculated by the egress cipher suite is appended at the end of packet data (excluding original CRC).
- When the SL!=0, the ICV calculated by the egress cipher suite is appended at the offset (indicated by SL) from end of SecTAG before being padded by the MAC if necessary. (The original padding bytes and CRC are removed before appending the ICV).

Maintaining PN Number for Egress Packets

Each egress packet is assigned a 32-bit PN number in the SecTAG. The PN number is maintained in the SA_ATTRIBUTE_TABLE.Next_PN and incremented prior to applying MACsec to the packet. The PN number represents the lifetime of the SA. Only a valid SA is applied to protect the traffic. Typically, when the PN number maintained by the SA reaches 0xFFFFFFFF, the SA is invalidated by the hardware and the packet is dropped. An early warning threshold can be set via the configuration register SP_Egress_PN_THD so that an interrupt can be triggered to allow the host to rekey the SA before it expires. Alternatively, the PN number in the SA is allowed to roll over. In which case, the early warning threshold doesn't take effect. The PN number is incremented to 0x1 after it reaches 0xFFFFFFFF. See "SA Management" on page 66 for more details of SA lifetime handling.

Egress Short Packet Padding

When the MSDU to be protected is smaller than 48 bytes, the ShortLength field of the SecTAG contains the length of the MSDU. Otherwise, the ShortLength field is set to 0. When the packet is transmitted, it is padded by the MAC to satisfy the minimum packet size required. The receiver uses the ShortLength field to correctly identify the location of the ICV.

Egress Packet Encryption and Authentication ICV Generation

After SecTAG Insertion, the packet is forwarded to the AES-GCM engine for security processing. The MACsec logic provides control information to specify the starting location of the security payload to apply authentication and encryption. The AES-GCM engine generates the ICV after the packet is encrypted. The ICV is 16 bytes long. It is appended to the end of the packet.

Egress Post Processing

There is very little post processing to be done for egress traffic. The MIB counters for the controlled port are updated when the payload length information is collected at the end. Afterwards, the egress packet is queued into the egress packet buffer for transmit.

Egress Controlled Port MTU check

If the resulting packet after SecTAG and ICV insertion exceeds the programmed value, the logic truncates the packet and corrupts the CRC of the packet and records the event in the MIB counter.

Egress Store and Forward Mode

When Egress store and forward mode is enabled, the packets are stored in the 32 KB SAF buffer to check against error packets after post processing and MTU check. The error packets that carry along with mac_err bit or egress pipe generated error packets are dropped before sending out to line-side MAC for transmission. In cut through mode, all packets are sent to line-side MAC regardless of error.

Egress Packet Transmission

The uncontrolled port traffic and controlled port traffic are combined by the MACsec transmission control logic into a single stream of packets and forwarded to the line side MAC for transmission. The line side MAC is responsible for regenerating the FCS and inserting it to the end of packet.

Egress Non-Destructive Loopback

The field Loopback_L2Redirect in table EGRESS_SA_ATTRIBUTE_TABLE can be set to loopback packets from egress to ingress. Note in this mode the maximum size of the loopback packets is 512 bytes.

Egress MFP (EFP)

In the MACsec egress direction, any packets from the higher layer device need to be classified to the Controlled Port or the Uncontrolled Port traffic. If the packet belongs to the Controlled Port traffic, a policy engine needs to decide how SecY processes this packet. For example, which secure channel this packet belongs to, and which security policy is going to be applied to this packet (i.e. integrity or confidentiality plus integrity). Similar to the receive side, a pre-MACsec packet classification (also called Egress Flow Processor (EFP)) is needed to parse the packet in terms of the L2 information and generate the right action for the packet. The major features include:

- Parse the pre-MACsec packets with SecTAG and ICV, if available.
- Support Ethernet II/LLC/SNAP packets with or without SecTAG.
- Support Special VLAN Tagged packets without SecTAG.
- Support single VLAN Tag, double VLAN Tags, or no VLAN Tag. Four outer VLAN TAG TPIDs are configurable. One inner VLAN Tag TPID is also configurable.
- Parse QTAG and STAG, and extract the user priority (UP) in terms of the configuration.
- Classify the management packets in terms of the configuration. Provide per-rule action and 40-bit counter.
- Derive the packet_type in terms of the packet header information.
- Generate the lookup key for MFP_EFP_KEY_TABLE in terms of the configurable key selection register.
- Look up MFP_EFP_KEY_TABLE to find the index of the matched entry. Only the lowest entry is selected if there are multiple entries matched.
- Look up MFP_EFP_ACTION_TABLE to retrieve the action specified for the matched flow.
- Perform egress MacSecID processing and provide the action to ESEC.
- Derive C and E bit from the packet or MFP_EFP_ACTION_TABLE based on the configuration.
- Send the action data to ESEC for the further processing.
- Increment the 48-bit counter MFP_MIB_TABLE per flow if MFP_EFP_KEY_TABLE gets a hit. The counter is clear-on-read and saturated based on the global configuration.
- 128 EFP flows shared with IFP.

The following types of packet formats are expected to be generated by the system side and fed into the MACsec PHY device for egress MACsec operation.

- Native Packet: A packet that carries no explicit tag field for MACsec egress operation.
- Special VLAN Tagged Packet: A packet that carries a special 16-bit tag (indicated by a configurable proprietary EtherType) as its outermost tag to indicate explicit information for MACsec egress operation. The Special VLAN Tag also contains C and E bits to control whether the packet is encrypted or authenticated only.
- SecTAG Tagged Packet: A packet that carries an 8- or 16-byte SecTAG (indicated by the MACsec specific EtherType). The SecTAG.TCI field contains C and E bits to control whether the packet is encrypted or authenticated only.

Table 10 shows the format of the Special VLAN Tag:

Table 10: Egress ST-VLAN Tag Format

Field Name	***Bits***	***Description***
TAG Protocol ID	31:16	TPID to indicate this is a ST-VLAN tag packet.
Reserved	15:14	Reserved.

Table 10: Egress ST-VLAN Tag Format (Cont.)

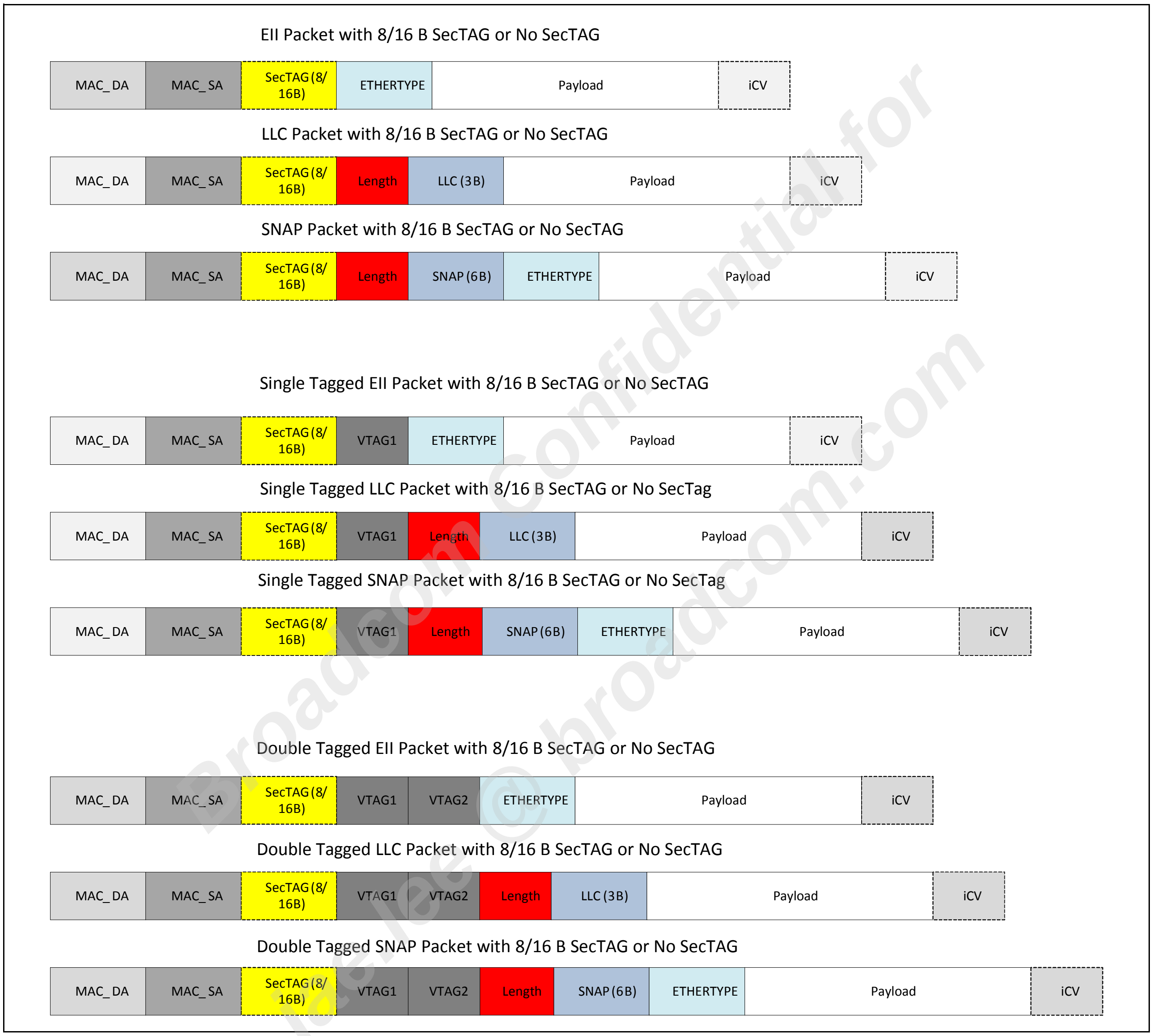
Field Name	Bits	Description
C	13	C bit in SecTAG TCI.
E	12	E bit in SecTAG TCI.
Flow Identifier	11:0	12-bit value to identify an SCI or replace the lower 12-bit of the SCI.

Although all three types of packet formats could be used concurrently in system design, the Special VLAN Tagged Packet format and the SecTAG Tagged Packet have similar and duplicated purpose for conveying MACsec processing information across the interface between the system and the MACsec PHY, so we do not support the presence of both the Special VLAN Tag and SecTAG concurrently in a packet.

EFP Parser

The EFP Parser is a robust L2 header parsing module that supports the packet data up to 64B. It starts the packet parsing whenever the packet SOP data is received. The supported packet formats are shown in Figure 5:

Figure 5: EFP Supported Packet Formats



The SecTAG field can be 8B SecTAG without SCI, 16B SecTAG with SCI, or blank for non-SecTAG packets. The EFP Parser recognizes the special EtherType value of 0x88E5 and a per-port configurable register to identify the presence of a SecTAG. If a SecTAG is present, it is always after the MACSA. SecTAG.TCI.SC tells the parser whether it's a 8B SecTAG or 16B SecTAG. SecTAG.TCI, SecTAG.AN and SecTAG.SCI is extracted for MFP_EFP_KEY_TABLE lookup key generation. MacSecID is extracted from the SecTAG. The 16-bit MacSecID value is overlayed on the 16-bit LSB of the SecTAG.PN field. MacSecID is used as one of the fixed key fields of the MFP_EFP_KEY_TABLE lookup key.

The EFP Parser supports single VLAN Tag or Double VLAN Tag packets. Three Outer TPID values (TPID1, TPID2 and TPID3) and one Inner TPID value are defined to identify the outer and inner VLAN Tags. TPID1 corresponds to QTAG, and TPID2 corresponds to STAG. Both of them can be used as inner VLAN Tags if Q-in-Q is enabled. The VLAN Tag is defined as follows:

Table 11: VLAN Tag Format

Field Name	***Bits***	***Description***
TPID	31:16	VLAN Tag protocol Identifier
User_priority	15:13	3-bit user priority
CFI	12	Canonical format indicator
VID	11:0	VLAN identifier

One special TPID value is also defined to identify the special VLAN Tag which may carry proprietary information for the special applications. The Special VLAN Tag is always treated as one of the Outer VLAN Tags (without a user-priority field). The VTAG1 is specified by one of the three egress outer TPID registers, or the egress special TPID register for the special VLAN Tag. With all outer VLAN Tags, the priority is from higher to lower in the order of TPID2, TPID1, TPID3, and Special_TPID. VTAG2 is specified by the egress inner TPID register or TPID1 and TPID2 registers. QTAG_TPID and STAG_TPID are programmed to TPID1 and TPID2 registers, respectively. If q-in-q is disabled for the double tag checking, the EtherType of the packet is the TPID of the VTAG2. The parser encodes the final VLAN Tag parsing result to packet_vlan_tag_format defined in the below table:

Table 12: Packet_VLAN_Tag_Format Encoding

Packet VLAN Tag Format	***Value***	***Description***
No VLAN Tag	4’b0000	Untagged packet.
Single Inner VLAN Tag	4’b0001	Single VLAN Tag. It’s an Inner Tag.
Single Outer VLAN Tag	4’b0100	Single VLAN Tag. It’s a Special VLAN Tag.
Single Outer VLAN Tag	4’b0101	Single VLAN Tag. It’s an Outer TPID1, which is used as QTAG.
Single Outer VLAN Tag	4’b0110	Single VLAN Tag. It’s an Outer TPID2, which is used as STAG.
Single Outer VLAN Tag	4’b0111	Single VLAN Tag. It’s an Outer TPID3.
Double VLAN Tag	4’b1000	Double VLAN Tag. The Outer Tag is a Special VLAN Tag.
Double VLAN Tag	4’b1001	Double VLAN Tag. The Outer Tag is Outer TPID1 which is used as QTAG.
Double VLAN Tag	4’b1010	Double VLAN Tag. The Outer Tag is Outer TPID2 which is used as STAG.
Double VLAN Tag	4’b1011	Double VLAN Tag. The Outer Tag is Outer TPID3.

Besides the TPID enable bit for each TPID register, there is a user priority (UP) enable bit for each TPID register except the special TPID and the inner TPID register. If VTAG1 matches one of the TPID registers and UP is enabled for the TPID, the PRI bits are extracted from VTAG1 and mapped to the configurable value if this TPID corresponds to a QTAG_TPID (i.e. TPID1). If VTAG1 doesn’t match any of the programmed TPIDs or the matched TPID doesn’t have UP enabled, a programmed default UP is used as the UP of this packet.

The EFP Parser decodes the 2 bytes data followed by the VLAN Tag. If it is larger than or equal to a configurable length value for EII, the packet is classified to an EII packet. The 2-byte data is the EtherType. Otherwise, it is a length field of the LLC/SNAP packet.

The LLC header looks like this:

Table 13: LLC Header Format

DSAP	SSAP	Control
8 bits	8 bits	8 or 16 bits

The SNAP header looks like this:

Table 14: SNAP Header Format

DSAP	SSAP	Control	3 Octet OUI	Comments
AA	AA	03	0x00_00_00	Zero-OUI SNAP

Any packets with DSAP/SSAP/Control/OUI that do not match the specified value in the SNAP header are classified to the LLC packets. The EtherType of LLC packets will be {DSAP, SSAP}.

Two separate UDFs of four bytes each are supported. The UDF (defined in Table 15) is parsed from the first 64 bytes of the packet starting at the beginning of the packet.

Table 15: UDF Definition

Field Name	Bits	Description
Valid	[7]	UDF Valid.
Reserved	[6]	Reserved.
OFFSET	[5:0]	Offset of the user-defined field starting from the beginning of the packet.

EFP Key Selection

The EFP lookup key consists of the fixed fields, the selectable L2 fields parsed from the packet, and the user defined fields (UDF) extracted at the configurable offsets from the first 64 bytes of the packet. The 234-bit MFP_EFP_KEY_TABLE lookup key is defined in Table 16.

Table 16: Lookup Key of MFP_EFP_KEY_TABLE

Field Name	Bits Field	Descriptions
direction	233	0:ingress 1:egress
Port_num	232:229	Reserved.
SecTAG_Status	228:227	SecTAG status. 0b00: frame does not contain a SecTAG. 0b01: frame contains a SecTAG. 0b1x: reserved
Frame_format	226:225	Type of Ethernet frame. 0:Ethernet II packet (LENTYPE>=CP_ETYPE_MAX_LEN). 1:SNAP packet (aa-aa-03-00-00-00). 2:LLC packet (LENTYPE < CP_ETYPE_MAX_LEN and !SNAP). 3:Reserved.

Table 16: Lookup Key of MFP_EFP_KEY_TABLE (Cont.)

Field Name	Bits Field	Descriptions
Vlan_tag_status	224:221	Type of VLAN Tags found on frame. 0x0:untagged packet. 0x1:Single VLAN Tag. It's Inner Tag. 0x4:Single VLAN Tag. It's ST-VLAN Tag. 0x5:Single VLAN Tag. It's Outer TPID1 (i.e. QTAG). 0x6:Single VLAN Tag. It's Outer TPID2 (i.e. STAG). 0x7:Single VLAN Tag. It's Outer TPID3. 0x8:Double VLAN Tag. Outer Tag is ST-VLAN Tag. 0x9:Double VLAN Tag. Outer Tag is Outer TPID1 (i.e. QTAG). 0xa:Double VLAN Tag. Outer Tag is Outer TPID2 (i.e. STAG). 0xb:Double VLAN Tag. Outer Tag is Outer TPID3. others: Reserved.
Pkt_type	220:219	Type of packet 00–Non-MACsec packet. 01–MACsec packet.
Reserved	218:217	–
EFP_KEY_SLICE2	216:200	This slice of key is configurable by the Egress_SLICE2_SEL register. If Egress_SLICE2_SEL=1, EFP_KEY_SLICE2 is defined by Table 20: “EFP_KEY_SLICE2_1,” on page 59. Otherwise, it’s defined by Table 19: “EFP_KEY_SLICE2_0,” on page 59.
EFP_KEY_SLICE1	199:136	This slice of key is configurable by the Egress_SLICE1_SEL register. If Egress_SLICE1_SEL=1, EFP_KEY_SLICE1 is defined by Table 18: “EFP_KEY_SLICE1_1,” on page 58. Otherwise, it’s defined by Table 17: “EFP_KEY_SLICE1_0,” on page 58.
SecTAG.TCI_AN	135:128	SecTAG TCI and AN fields (i.e. SecTAG.TCI+SecTAG.AN).
ETHERTYPE	127:112	EtherType for Ethernet II/SNAP packets. It’s DSAP+SSAP for LLC packets.
OUTER_TAG	111:96	Outer VLAN Tag (i.e. PRI+CFI+VID). For the ST_VLAN Tag, the format is {ST_VLAN_reserved[15:14], C, E, Flow_Identifier[11:0]}.
SA	95:48	MAC source address.
DA	47:0	MAC destination address.

Table 17: EFP_KEY_SLICE1_0

Field Name	Bits Field	Description
SCI	199:136	SecTAG SCI field

Table 18: EFP_KEY_SLICE1_1

Field Name	Bits Field	Description
UDF1	199:168	The 4B user-defined field. It’s defined by register Egress_UDF1.

Table 18: EFP_KEY_SLICE1_1 (Cont.)

Field Name	Bits Field	Description
UDF0	167:136	The 4B user-defined field. It's defined by register Egress_UDF0.

Table 19: EFP_KEY_SLICE2_0

Field Name	Bits Field	Description
Reserved	216	–
SecTAG.PN[15:0]	215:200	The lower 16 bits of SecTAG PN field.

Table 20: EFP_KEY_SLICE2_1

Field Name	Bits Field	Description
Reserved	216	–
INNER_TAG	215:200	Inner VLAN Tag (i.e.PRI+CFI+VID)

MFP Search Engine Arbiter

As EFP shares the same search engine as IFP, the MFP Search Engine Arbiter performs the round robin algorithm to select EFP lookup key request and IFP lookup key request. The winner is marked with a label in port ID to the MFP Search Engine. The MFP Search Engine takes different actions depending upon the source of the lookup key request. After the MFP Search Engine generates an action for the lookup request, the MFP Search Engine Arbiter distributes it to EFP_EBUF if EFP is the winner. Otherwise, it is sent to IFP_EBUF.

MFP Search Engine

The MFP Search Engine consists of a TCAM which implements both MFP_EFP_KEY_TABLE and MFP_IFP_KEY_TABLE. MFP_EFP_KEY_TABLE is defined in Table 21.

Table 21: MFP_EFP_KEY_TABLE Definition

Field Name	Bits Field	Description
KEY_VALID	468	Indicates this entry is valid. 1: Valid 0: Invalid.
KEY_MASK	467:234	Per-bit key comparison enable. 1: Compare 0: Ignore.
KEY	233:0	234-bit lookup key generated by EFP Key Selection.

The TCAM size is 128 x 234 with 234 bits lookup key, 234 bits key mask and a valid bit to indicate whether this entry is valid or not. Even though MFP_EFP_KEY_TABLE and MFP_IFP_KEY_TABLE have different definitions, they share the same TCAM. The software has the flexibility to allocate the dynamic memory space for MFP_IFP_KEY_TABLE and MFP_EFP_KEY_TABLE, respectively. If the lookup key matches any of the entries in the TCAM, the packet is classified as belonging to the flow defined by the entry. The action associated with the flow is applied to the packet. If more than one entry is matched, the first match is picked up.

The MFP_EFP_ACTION_TABLE is defined in Table 22.

Table 22: MFP_EFP_ACTION_TABLE Definition

Field Name	Bits Field	Description
POLICY	[31:27]	[31]–If set, the packet is controlled. The only exception is the KaY frame. If the non-MACsec packet is received from the system-side device and the packet is programmed to be a KaY frame (C=0,E=1) either by C and E bits in this table or from the C and E bits in the special VLAN Tag, this bit needs to be set for the egress engine to insert a dumb SecTAG with C=0, E=1, SC=1, PN=0 and SCI=0. The packet is still sent out as the uncontrolled port traffic. [30] If bit [31] is set, selects between C and E bits from the special VLAN Tag/ SecTAG or from the EFP Action Table. 1'b1–Use C and E bits from packet. 1'b0–Use C and E bits from EFP Action Table. [29]–If bit [31] is set, indicates whether to use the SC and SCI fields from the incoming packet's SecTAG. The SCI_Index always comes from the EFP Action Table. 1'b0–Use TCI.SC bit and SecTAG.SCI field from packet if present in the outgoing packet. 1'b1–Use SC bit and SecTAG.SCI field from the SC Table as the outgoing packet's SecTAG.SCI and SecTAG.TCI field. [28]–If bit [31] is set, indicates whether to use the ES and SCB fields from the incoming packet's SecTAG. 1'b0–Use TCI.ES bit and TCI.SCB bit from packet if present in the outgoing packet. If it's a non-MACsec packet, TCI.ES and TCI.SCB comes from the egress SC Table. 1'b1–Use TCI.ES bit and TCI.SCB bit from the egress SC Table as the outgoing packet's SecTAG.TCI.ES and SecTAG.TCI.SCB field. [27] Indicates block/drop the packet. 1'b0–Pass. If bit 31 is 0, the packet will be treated as uncontrolled packet and passed to the line as is. Otherwise, the packet will be a controlled port packet and look up the EGRESS_SC_INDEX_TABLE for the packet protection processing. 1'b1–Drop the packet. The packet will be dropped no matter bit 31 is 0 or 1.
C	[26]	C bit in SecTAG TCI.
E	[25]	E bit in SecTAG TCI.

Table 22: MFP_EFP_ACTION_TABLE Definition (Cont.)

Field Name	Bits Field	Description
V	[24]	V bit in SecTAG TCI. If the packet is a SecTAG packet, the outgoing packet's SecTAG.TCI.V comes from the packet. Otherwise, use V bit in EFP Action Table.
REDIR_DBUG_ FIFO	[23]	1'b0: Normal mode. 1'b1: Redirect to debug capture FIFO.
Reserved	[22:17]	Reserved bits
SCI_Index	[16:10]	If redir_dbug_fifo =1, this field is a match index to be carried in the reason code otherwise lower 4-bits is the index to the SC Table.
ACTION	[9:6]	ACTION_TYPE: [9]: When set, remove the outer Special VLAN tag. This action can be set only when the packets contain the Special VLAN Tag (i.e. match vlan_tag_status). In addition, this action is only applied to the controlled port packets. [8]: Reserved [7]: When set, override the lower 12-bits of SCI with the VID value of the outer VLAN tag. [6]: When set, override the lower 12-bits of SCI with the VID value of the inner VLAN tag.
Reserved	[5:0]	Reserved bits.

If both bit 31 and bit 30 are set in MFP_EFP_ACTION_Table, the value of the C/E bits come from the packet. If the packet is a SecTAG packet, C and E come from SecTAG.TCI.C and SecTAG.TCI.E. If the packet is a special VLAN Tagged packet, C and E come from bit 13 and bit 12 of the VLAN tag.

If bit 7 is set, the MFP Search Engine provides the 12-bit VID from the outer VLAN Tag to ESEC. Otherwise, if bit 6 is set, the MFP Search Engine provides the 12-bit VID from the inner VLAN Tag to ESEC.

As ESEC needs user priority (UP) for per-UP MIB counters and per-UP MTU check, the MFP Search Engine puts UP decoded from the packet VLAN Tag in the reserved bits 19:17 of the action data for all packets including ESEC2fp_up_only packets: (i.e., packets when SP_MASTER_Ctrl.ControlledPortEnabled = 0), ESEC retrieves UP from the action_data even though the MFP_EFP_KEY_TABLE lookup is a miss.

If MFP_EFP_ACTION_TABLE has a double-bit ECC error, the MFP Search Engine uses the reserved bit 20 in the action data to notify ESEC to treat the packet as a failed packet. The packet is dropped like CRC error packets.

In addition, EFP tells ESEC whether the packet is a SecTAG packet or not via reserved bit 0 of the action data. This bit is independent on the MFP_EFP_KEY_TABLE lookup hit or miss.

EFP EBUF

EFP EBUF receives the action data from the MFP Search Engine Arbiter. The action data is sent to ESEC with MFP_EFP_KEY_TABLE match or mismatch information. If it is a match, the action_data and the other sideband signals are valid. Otherwise, the packet is treated as an EFP mismatch packet, the default programmed action either drops the packet or treats the packet as an uncontrolled port packet.

MFP MIB

The MFP MIB implements a register file to store all MFP flow counters. A 48-bit counter is associated with each TCAM table entry (128 counters per TCAM). The MFP Search Engine generates a TCAM hit entry index for each packet classified. They are sent to MFP MIB to update the counter associated with the TCAM entry index. If the counter overflows, it stays at all 1s if the programmable register SECY_CFG_GLB_MIB_Ctrl cnt_saturate_dis is not set. Otherwise, it rolls over to 0. The MFP MIB supports the CPU read and write operation. For the read operation, the counter is cleared on read if the programmable register cnt_rd_clr_dis is not set. Otherwise, the counter are not cleared. There is a global counter clear register called clr_mib_cnt. If this signal is asserted, all MFP counters are cleared to 0.

The MFP MIB provides the status update to EFP register whenever there is a single bit or a double bit ECC error detected for MFP MIB memory.

EFP Register

The EFP register implements all EFP registers. It receives the request from MFP and acknowledges when the read/write operation is completed.

The EFP register implements the single bit ECC error counter and double bit ECC error counter for all MFP memories. Those counters can be cleared by the global register clr_other_cnt. If the counter overflows, it stays at all 1s if the programmable register cnt_saturate_dis is not set. Otherwise, it rolls over to 0. For the CPU read operation, the counter is cleared on read if the programmable register cnt_rd_clr_dis is not set. Otherwise, the counter is not cleared.

In addition, the EFP register implements the ESEC2fp_up_only packet counter and the MFP_EFP_KEY_TABLE miss counter. The EFP register also generates all MFP interrupt events, such as double bit ECC error per MFP memory.

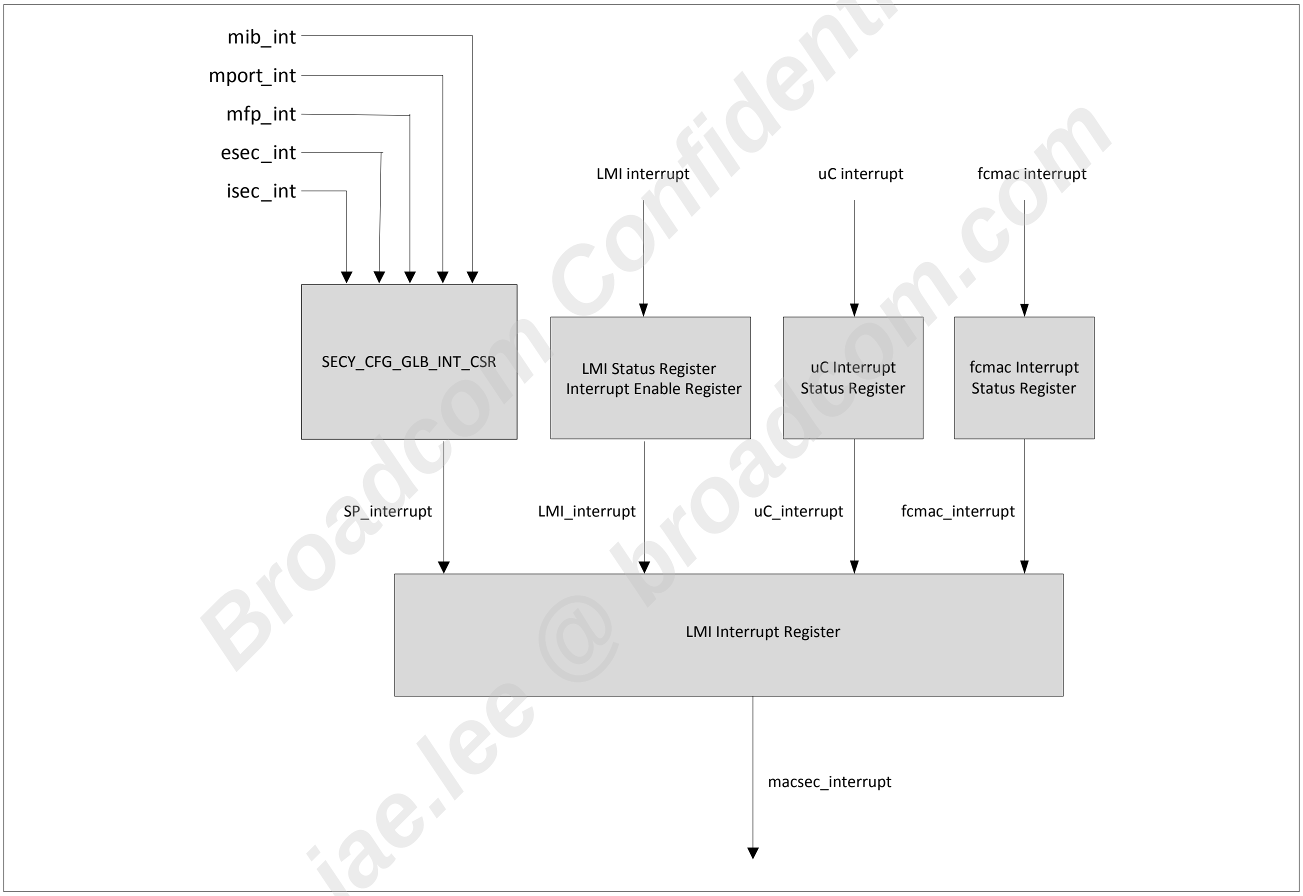
SecY Management

This section discusses the management mechanisms deployed for SecY.

Interrupt

LMI is the top-level of the MACsec interrupt hierarchy. It collects the interrupts from all internal blocks, as well as the LMI interrupt. These interrupt events are shown in LMI Interrupt register, LMI_Interrupt, and sent to the chip top-level as MACsec interrupt and ORed with the physical layer interrupts before outputting to the interrupt pin.

Figure 6: Interrupt Hierarchy



The LMI interrupt events are shown in the Status register Port_LMI_Status. They are cleared on read. Software can enable them via LMI Port Interrupt register, Port_LMI_Interrupt_EN.

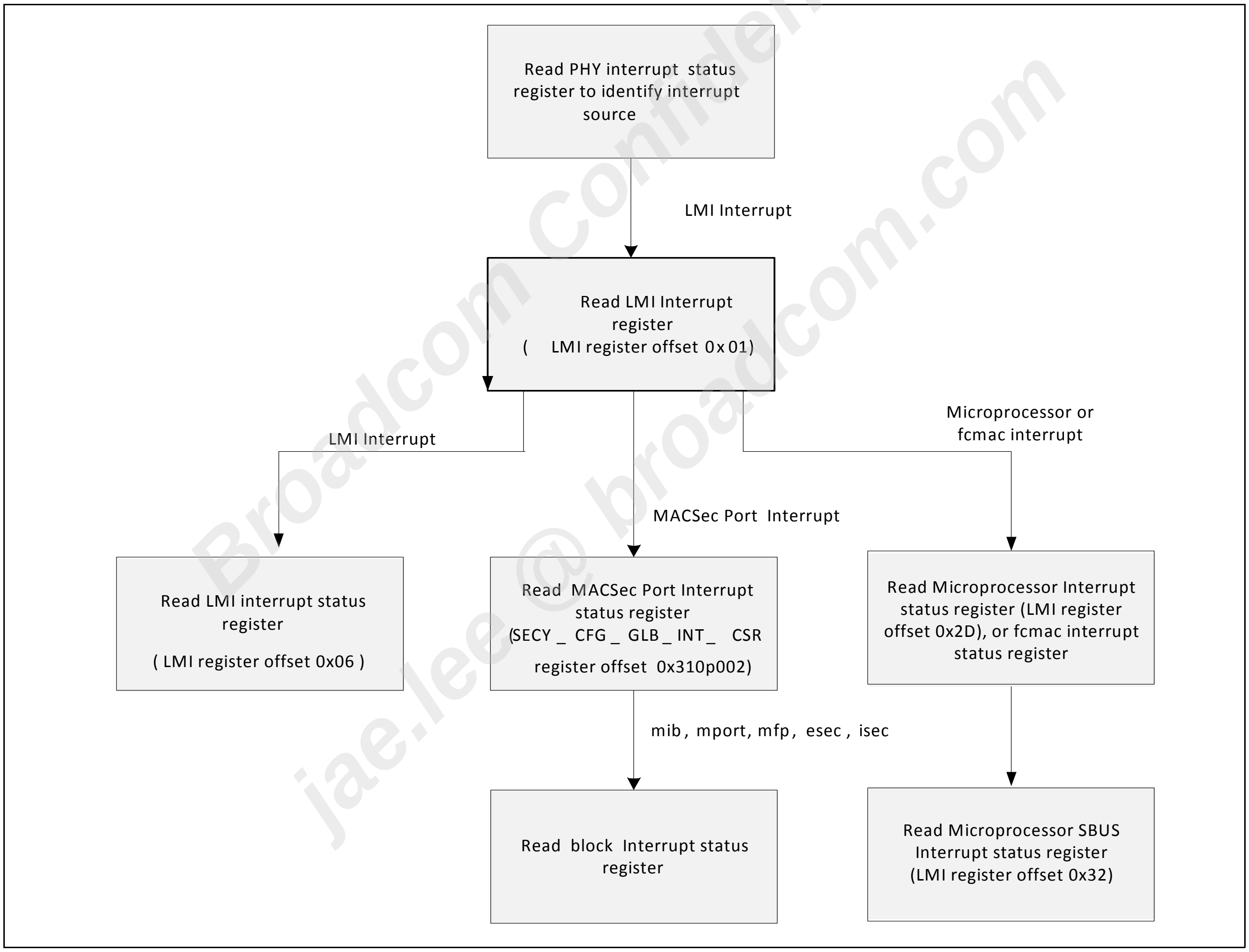
Interrupt Enable register, LMI_Interrupt_Enable, needs to be configured to enable the MACsec port interrupt and LMI interrupt as needed. If LMI interrupt is enabled, the LMI Port Interrupt register needs to be configured to enable LMI interrupt events to generate the interrupt.

If a MACsec interrupt was generated, software needs to read the LMI Interrupt register to find out if the interrupt is generated from the LMI or the SECPORT. If the interrupt originates from LMI, read the LMI status register to identify the interrupt source. If the interrupt is in the SECPORT, software reads the SECY_CFG_GLB_INT_CSR register to find out which block asserts the interrupt. The possible sources are:

- ISEC
- ESEC
- MFP
- MPORT
- MIB

Reads the interrupt status register of the reporting block to identify the interrupt source. Interrupts are clear when the interrupt status register of the reporting block is read. Figure 7 illustrates how an LMI interrupt is handled.

Figure 7: Interrupt Handling Flow Diagram


Read PHY interrupt status register to identify interrupt source
LMI Interrupt
Read LMI Interrupt register (LMI register offset 0x01)
LMI Interrupt
Microprocessor or fcmac interrupt
MACSec Port Interrupt
Read LMI interrupt status register (LMI register offset 0x06)
Read MACSec Port Interrupt status register (SECY _ CFG _ GLB _ INT _ CSR register offset 0x310p002)
Read Microprocessor Interrupt status register (LMI register offset 0x2D), or fcmac interrupt status register
mib, mport, mfp, esec, isec
Read block Interrupt status register
Read Microprocessor SBUS Interrupt status register (LMI register offset 0x32)

SA Management

SA Lifetime

Each security association key (SAK) used to protect traffic in a security association has a lifetime due to security concerns with regard to counter mode encryption. The lifetime of the key is determined by a 32-bit packet number (PN). The PN is embedded in the SecTAG and it is incremented monotonically for every packet transmitted with the same SAK until it reaches its saturation value of ‘0xFFFFFFFF’. Afterwards, the SAK needs to be refreshed.

The SecY module uses interrupt to inform the host that the SAK has expired. A configurable threshold is used to allow enough time for the host to perform a new round of key refreshment. When the PN number reaches the threshold level, the corresponding SAK is considered soft expired. When the PN number reaches the saturation value, the corresponding SAK is considered hard expired. A soft-expired SAK continues to be applied to MACSec packet processing. A configurable interrupt can be generated when the soft-expiration condition is detected. A hard-expired SAK could be invalidated immediately. A separate configurable interrupt is used for reporting hard expiration. For any SAK expiration, the corresponding SA index is latched into the interrupt status register of the corresponding port so that the host can determine which key to refresh promptly. Related registers are:

- SECY_CFG_GLB_INT_CSR
- SP_Ingress_SA_status0
- SP_Ingress_SA_status1
- SP_Ingress_PN_THD
- SP_Ingress_Ctrl
- SP_ingress_STAT
- SP_Egress_SA_status0
- SP_Egress_SA_status1
- SP_Egress_PN_THD
- SP_Egress_Ctrl
- SP_Egress_STAT

For 10G links, the lifetime of the SA is approximately 4.8 minutes based on the maximum packet rate of 14.88 Mpps.

The SA management of MACSec is also designed to support the notion of non-interrupting service for both transmitting and receiving packets. This is achieved by support more than one SAK for each SC. One key is used for actively protecting the traffic. The other SAKs are used for backup. The software can setup more than one backup SAKs at a time and configure the hardware in Auto_AN_Switch mode through SP_Master_Ctrl and SP_Egress_AUTO_AN_SWITCH_CTRLregister. On transmit, when the current AN’s NEXT_PN reaches 0xFFFFFFFF, the hardware automatically switches to the backup SAK. This reduces the frequency that the software has to be interrupted to setup a new SAK.

The 802.1AE standard requires the support of overlapping usage the two SAKs of the same SC (active and backup) for a minimum of 0.5 second due to packet reordering over the network. The SecY module doesn’t enforce this time limit. However, it does allow two receiving SAKs to be used overlapping each other.

The transmit SAKs are identified by the active AN number programmed into the CA memory for each SC. The AN number is two bits wide. The host changes the AN number in order to switch to a new SAK. This provides minimum switch-over time required by the standard. The receiving SAKs are identified by the AN number embedded in the SecTAG of the packet.

Global Timer

A per-port timer based on the SECY_CFG_GLB_TimeTick register and the SECY_CFG_GLB_PreScale register is provided to the SecY module. The initial value of the time tick and pre-scaling factor can be configured through these two registers. The timer is used as a reference time tick to track the start and stop time of the SA when it is applied to protect the traffic. The default value of the Pre-Scale register sets the time tick to 1s intervals based on a 156.25 MHz clock. The timer is mainly used to timestamp the SA for management purposes.

Auto AN Switch

The register field, SP_MASTER_Ctrl.Auto_AN_Switch, can turn on the feature of Auto AN Switch, which means that AN is automatically switched to the next available SA after the current SA is expired. The AN number in the EGRESS_SC_INDEX_TABLE entry indicates which SA is currently in use within this SC. At egress, the AN number can be programmed to automatically switch to the next available SA upon the current SA expired. Note that if PN rollover mode is enabled in EGRESS_SA_ATTRIBUTE_TABLE, there is no auto switch on the AN number. If this field is set, it can globally enable all SCs auto AN switch function. If Auto_AN_Switch is not set, another register, SP_EGRESS_AUTO_AN_SWITCH_CTRL, can individually enable each of the SCs auto AN switch function.

In two SAs/SC mode, the auto AN switch must be: 0->1->2->3->0->..., although the starting AN can be either one, the next AN must be the immediately following one (e.g., if the current AN is 0, the next one has to be AN 1).

In four SAs/SC mode, the auto AN switch can jump to the next valid AN (e.g., if the current AN is 0, AN 1 is invalid, AN 2 is valid, upon AN 0 expires, AN 2 is used).

Summary

MACsec has a vast set of registers. This document just describes the usage of the popular ones. Refer to *MACsec and Fiber Channel over Ethernet (FCoE) Features* (Reference [1] on page 9) for more information about each register.

Appendix A: Pseudo Code of MACsec Registers Programming

The following initialization procedure is used when the control port is disabled and enabled in the following two subsections.

Initialization Procedure When Control Port Disabled

```
#################################################################################
The following file is the procedure to program MACsec core when Control Port
# is disabled. This script can be used for any port.
# The key steps are:
#### Step 1. Program SFI/Copper/SGMII and XFI/SGMII/QSGMII registers to
####         establish the 10G/1G link. Enable Port 0 MACSEC
#### Step 2. Program line-side XMAC Registers for 10 Gbps
#### Step 3. Program line-side UNIMAC Registers for non-10 Gbps
#### Step 4. Program switch-side/system-side XMAC registers for 10 Gbps
#### Step 5. Program switch-side/system-side UNIMAC registers for non-10 Gbps
#### Step 6. Initialize MACsec functions
#### Step 7. Start sending the traffic in both directions
################################################################################

################################################################################
#### Step 1. Program SFI/Copper/SGMII and XFI/SGMII/QSGMII registers to
####         establish the 10G/1G link. Enable Port 0 MACSEC
#### wait 10ms

##########################################################
#### Step 2. Program line-side XMAC Registers for 10 Gbps
##########################################################
#### write XMAC XMAC_CTRL register, enable Rx and Tx, clear soft reset
write 0x01100000 0x00000003

#### write XMAC XMAC_TX_CTRL register, append CRC, enable padding
write 0x01100004 0x100000C810

#### write XMAC XMAC_RX_MAX_SIZE register,16KB max Rx packet size
write 0x01100008 0x00003FE4

#### write XMAC XMAC_RX_LSS_CTRL register, enable external LF/RF
write 0x0110000A 0x00000004

#### write XMAC XMAC_PAUSE_CTRL register, enable RX and TX PAUSE
write 0x0110000D 0x0007C000

#### write XMAC XMAC_MACSEC_CTRL register, enable the CRC corruption for the TX error packets
write 0x01100024 0x00000002

##########################################################
#### Step 3. Program line-side UNIMAC Registers for non-10 Gbps
##########################################################
#### write UNIMAC COMMAND_CONFIG: SW_RESET=1
write 0x00100202 0x014020d8
```

```
#### write UNIMAC COMMAND_CONFIG: RX_ENA=1, TX_ENA=1, ETH_SPEED=2, PROMIS_EN=1, CRC_FWD=0,
PAUSE_FWD=0, ENA_EXT_CONFIG=1
write 0x00100202 0x01c0001b

#### write UNIMAC FRM_LENGTH
write 0x00100205 0x3fe4

#### write UNIMAC TX_IPG_LENGTH
write 0x00100217 0xc

#### write UNIMAC MACSEC_CNTRL register,enable CRC corruption on TxERR packets
write 0x001002C5 0x2

##########################################################
#### Step 4. Program switch-side/system-side XMAC registers for 10 Gbps
##########################################################
#### write XMAC XMAC_CTRL register,enable Rx and Tx, clear soft reset
write 0x01101000 0x00000003

#### write XMAC XMAC_TX_CTRL register, append CRC, enable padding
write 0x01101004 0x100000C810

#### write XMAC XMAC_RX_CTRL register, RECEIVE_18_BYTE_PKTS=1, RUNT_THREHOLD=0x12
write 0x01101006 0x00000092C

#### write XMAC XMAC_RX_MAX_SIZE register,16KB max Rx packet size
write 0x01101008 0x00003FE4

#### write XMAC XMAC_RX_LSS_CTRL register, enable external LF/RF
write 0x0110100A 0x00000004

#### write XMAC XMAC_PAUSE_CTRL register, enable RX and TX PAUSE
write 0x0110100D 0x0007C000

#### write XMAC XMAC_MACSEC_CTRL register, enable the CRC corruption for the TX error packets
write 0x01101024 0x00000002

##########################################################
#### Step 5. Program switch-side/system-side UNIMAC registers for non-10 Gbps
##########################################################
#### write UNIMAC COMMAND_CONFIG: SW_RESET=1
write 0x00101202 0x014020d8

#### write UNIMAC COMMAND_CONFIG: RX_ENA=1, TX_ENA=1, ETH_SPEED=2, PROMIS_EN=1, CRC_FWD=0,
PAUSE_FWD=0, ENA_EXT_CONFIG=1 RUNT_FILTER_DIS=1
write 0x00101202 0x41c0001b

#### write UNIMAC FRM_LENGTH
write 0x00101205 0x3fe4

#### write UNIMAC TX_IPG_LENGTH
write 0x00101217 0xc

#### write UNIMAC MACSEC_CNTRL register,enable CRC corruption on TxERR packets
write 0x001012C5 0x2
```

```
##########################################################
#### Step 6. Initialize MACsec functions
##########################################################
#### clear the RMON counters, MACSEC MIB counters and MFP MIB counters
write 0x03100003 0x3

#### write MPORT_LN_XPORT_RSV_MASK to set line side MAC packet RX error mask
write 0x02100036 0x30054

#### write MPORT_SW_XPORT_RSV_MASK to set switch/system side MAC packet RX error mask
write 0x02100037 0x10054

#### write MPORT_LN_XPORT_STAT_UPDATE_MASK to set line side MAC statistic mask
write 0x02100038 0x3045D

#### write MPORT_SW_XPORT_STAT_UPDATE_MASK to set switch/system side MAC statistic mask
write 0x02100039 0x1045D

#### initialize MFP Key Table (MFP_EFP_KEY_TABLE and MFP_IFP_KEY_TABLE:128 entries and 59 bytes per
entry)
write 0x08200000-0x0820007f 0x0

#### initialize Ingress SC Index Table (INGRESS_SC_INDEX_TABLE:16 entries and 32 bytes per entry)
write 0x0b100000-0x0b10000f 0x0

#### initialize Egress SC Index Table (EGRESS_SC_INDEX_TABLE:16 entries and 9 bytes per entry)
write 0x0c100000-0x0c10000f 0x0

##########################################################
#### Step 7. Start sending the traffic in both directions
##########################################################
```

Initialization Procedure When Control Port Enabled

```
##############################################################################################
# This file provides the procedure to program MACsec core when Control Port is enabled
# The script is only used for BCM84756 in which MACsec register/memory addresses can be
# applied to any port.
# Test 1: IXIA->Line Side PHY->MACsec Ingress Flow->Switch Side SerDes (XFI)->IXIA
#         Generate the BPDU packet. The BPDU packet will be classified to be the uncontrolled
#        port packet via Ingress Management Packet Classification. The packet data will be sent out
#         AS IS
# Test 2: IXIA->Line Side PHY->MACsec Ingress Flow->Switch Side SerDes (XFI)->IXIA
#         The untagged (i.e.non-macsec) packet with MAC_DA=0x000123456789 will miss both Ingress
#         Management Packet Classification and Ingress_SC_INDEX_TABLE lookup. But it hits
#         MFP_IFP_KEY_TABLE rule 0. The packet will be sent out with a special VLAN Tag inserted
# Test 3: IXIA->Switch Side SerDes (XFI)->MACsec Egress Flow->Line Side PHY->IXIA
#         Generate the BPDU packet. The BPDU packet will be classified to be the uncontrolled
#         port packet. The packet data will be sent out AS IS.
# Test 4: IXIA->Switch Side SerDes (XFI)->MACsec Egress Flow->MACsec Ingress Flow->Switch Side SerDes
(XFI)->IXIA
#       The non-macsec packets (i.e.untagged packets) with MAC_DA=0xAABBCCDDEEFF will be classified
to be the controlled
#        port packets. The packets will be encrypted and loopbacked to MACsec Ingress. The packets
will be classified to
```

```
#         be the controlled port packets and be decrypted. (The generated packet from IXIA is non-macsec packet and received
#         packet is expected to be the plain text packet with SecTAG preserved. SecTAG.PN will be replaced with 0xEEEE)
#
# The key steps are:
#### Step 1. Program SFI/Copper/SGMII and XFI/SGMII/QSGMII registers to
####         establish the 10G/1G link.
####         Enable Port 0 MACSEC
#### Step 2. Program line-side XMAC at 10 Gbps
#### Step 3. Program line-side UNIMAC at non-10 Gbps


#### Step 4. Program switch-side/system-side XMAC at 10 Gbps
#### Step 5. Program switch-side/system-side UNIMAC at non-10 Gbps
#### Step 6. Initialize MACsec functions
#### Step 7. Ingress Flow Setup
#### Step 8. Egress Flow Setup
#### Step 9. Start sending the traffic in both directions

##############################################################################################

#### Step 1. Program SFI/Copper/SGMII and XFI/SGMII/QSGMII registers to
####         establish the 10G/1G link.
####         Enable Port 0 MACSEC

#### wait 10ms

##########################################################
#### Step 2. Program line-side XMAC at 10 Gbps
##########################################################
#### write XMAC XMAC_CTRL register,enable Rx and Tx, clear soft reset
write 0x01100000 0x00000003

#### write XMAC XMAC_TX_CTRL register, append CRC, enable padding
write 0x01100004 0x100000C810

#### write XMAC XMAC_RX_MAX_SIZE register,16KB max Rx packet size
write 0x01100008 0x00003FE4

#### write XMAC XMAC_RX_LSS_CTRL register, enable external LF/RF
write 0x0110000A 0x00000004

#### write XMAC XMAC_PAUSE_CTRL register, enable RX and TX PAUSE
write 0x0110000D 0x0007C000

#### write XMAC XMAC_MACSEC_CTRL register, enable the CRC corruption for the TX error packets
write 0x01100024 0x00000002

##########################################################
#### Step 3. Program line-side UNIMAC at non-10 Gbps
##########################################################
#### write UNIMAC COMMAND_CONFIG: SW_RESET=1
write 0x00100202 0x014020d8

#### write UNIMAC COMMAND_CONFIG: RX_ENA=1, TX_ENA=1, ETH_SPEED=2, PROMIS_EN=1, CRC_FWD=0,
PAUSE_FWD=0, ENA_EXT_CONFIG=1
```

```
write 0x00100202 0x01c0001b

#### write UNIMAC FRM_LENGTH
write 0x00100205 0x3fe4

#### write UNIMAC TX_IPG_LENGTH
write 0x00100217 0xc

#### write UNIMAC MACSEC_CNTRL register,enable CRC corruption on TxERR packets
write 0x001002C5 0x2

##########################################################
#### Step 4. Program switch-side/system-side XMAC at 10 Gbps
##########################################################
#### write XMAC XMAC_CTRL register,enable Rx and Tx, clear soft reset
write 0x01101000 0x00000003

#### write XMAC XMAC_TX_CTRL register, append CRC, enable padding
write 0x01101004 0x100000C810

#### write XMAC XMAC_RX_CTRL register, RECEIVE_18_BYTE_PKTS=1, RUNT_THREHOLD=0x12
write 0x01101006 0x0000092C

#### write XMAC XMAC_RX_MAX_SIZE register,16KB max Rx packet size
write 0x01101008 0x00003FE4

#### write XMAC XMAC_RX_LSS_CTRL register, enable external LF/RF
write 0x0110100A 0x00000004

#### write XMAC XMAC_PAUSE_CTRL register, enable RX and TX PAUSE
write 0x0110100D 0x0007C000

#### write XMAC XMAC_MACSEC_CTRL register, enable the CRC corruption for the TX error packets
write 0x01101024 0x00000002

##########################################################
#### Step 5. Program switch-side/system-side UNIMAC at non-10 Gbps
##########################################################
#### write UNIMAC COMMAND_CONFIG: SW_RESET=1
write 0x00101202 0x014020d8

#### write UNIMAC COMMAND_CONFIG: RX_ENA=1, TX_ENA=1, ETH_SPEED=2, PROMIS_EN=1, CRC_FWD=0,
PAUSE_FWD=0, ENA_EXT_CONFIG=1 RUNT_FILTER_DIS=1
write 0x00101202 0x41c0001b

#### write UNIMAC FRM_LENGTH
write 0x00101205 0x3fe4

#### write UNIMAC TX_IPG_LENGTH
write 0x00101217 0xc

#### write UNIMAC MACSEC_CNTRL register,enable CRC corruption on TxERR packets
write 0x001012C5 0x2


##########################################################
#### Step 6. Initialize MACsec functions
```

```
##########################################################
#### clear the RMON counters, MACSEC MIB counters and MFP MIB counters
write 0x03100003 0x3

#### write MPORT_LN_XPORT_RSV_MASK to set line side MAC packet RX error mask (10G)
write 0x02100036 0x30054

#### write MPORT_SW_XPORT_RSV_MASK to set switch/system side MAC packet RX error mask (10G)
write 0x02100037 0x10054

#### write MPORT_LN_XPORT_STAT_UPDATE_MASK to set line side MAC statistic mask (10G)
write 0x02100038 0x3045D

#### write MPORT_SW_XPORT_STAT_UPDATE_MASK to set switch/system side MAC statistic mask (10G)
write 0x02100039 0x1045D

#### initialize MFP Key Table (MFP_EFP_KEY_TABLE and MFP_IFP_KEY_TABLE:128 entries and 59 bytes per
entry)
write 0x08200000-0x0820007f 0x0

#### initialize Ingress SC Index Table (INGRESS_SC_INDEX_TABLE:16 entries and 32 bytes per entry)
write 0x0b100000-0x0b10000f 0x0

#### initialize Egress SC Index Table (EGRESS_SC_INDEX_TABLE:16 entries and 9 bytes per entry)
write 0x0c100000-0x0c10000f 0x0

#### enable the control port
#### SP_MASTER_Ctrl CONTROLLEDPORTENABLED=1 IngressStoreForward=0 EgressStoreForward=0
DropFailedPkts=1 UnmatchedPacketPolicy=0 sa_mode=1 en_zero_err_pkt=1 Ingress_SecTAG=2
write 0x03100100 0x00594197

###############################################################################
#### Step 7. Ingress Flow Setup
###############################################################################
#### Ingress Management Packet Classification (enable MAC_DA[47:4]=0x01_80_c2_00_00_0 for the bypass
action)
#### SP_Ingress_Pre_Psr_ctrl: CTRL_MACDA_HARDCODED_VAL0_EN=2
write 0x0310010E 0x00000200_00000000

###############################################################################
# Secure Channel 0 (SC0) for Ingress
###############################################################################
#### INGRESS_SC_INDEX_TABLE (32 bytes)
#### VALID=1 CIPHER_SUITE_PROTECTION=1 AN0=0 AN1=1 SCI=0xAACC0000 PKT_TYPE=1 EGR_INGR_LB=1
KEY_BYTE_MASK=0xFF KEY_BIT_MASK=0x8C00
write 0x0B100000 0x0458c000_000FF840_00000000_00000000_00000000_00000000_00000000_AACC0000

#### INGRESS_SA_ATTRIBUTE_TABLE for the Ingress Decryption (20 bytes)
#### VALID=3 NEXT_PN=1 SECTAG_MODE=0x2 ReplayControl=1
write 0x08100000 0x34000000_00000002_00000001_00000000_00000000

#### INGRESS_SA_KEY_TABLE for the Ingress Decryption (16 bytes)
#### SAK=0x5927
write 0x08108000 0x00000000_00000000_00000000_00005927

#### Program the length value for EII packet
#### Ingress_CP_ETYPE_MAX_LEN CP_ETYPE_MAX_LEN=0x600
```

```
write 0x00200106 0x600

#### Program Ingress Speical TPID
#### Ingress_Special_TPID Ingress_Special_TPID=0x8808 Valid=1
write 0x00200101 0x18808

#### MFP_IFP_KEY_TABLE (59 bytes)
#### enable DA, sectag_status and direction in the comparison
#### IFP Rule 0:KEY_VALID=1 DA=0x000123456789 sectag_status=0 (non-macsec packet) DIRECTION=0
(Ingress)
#### Please note IFP rules can be put in any place of the table which is shared with EFP. The entry#
is 0 of MFP rules
write 0x08200000
0x01860000_00000000_00000000_00000000_00000000_00000000_03ffffff_fffffc00_00000000_00000000_000000
00_00000000_00000000_00000001_23456789

#### MFP_IFP_ACTION_TABLE (4 bytes)
#### IFP Rule 0 ACTION:POLICY=0x3 (add a 4-byte special VLAN Tag with VLAN ID=0)
#### packets will not be dropped as an untagged packet in the strict mode and a special VLAN Tag is
inserted after MAC_SA
#### Please note IFP actions can be put in any place of the table which is shared with EFP. The
entry# is 0 of MFP actions
write 0x09200000 0x30000000

#### INGRESS_SC_ACTION_TABLE for the MACSECID (2 bytes)
#### MACSECID_VLAN=0xEEEE
write 0x0A100000 0xEEEE

###############################################################################
#### Step 8. Egress Flow Setup
###############################################################################
#### Program the length value for EII packet
#### Egress_CP_ETYPE_MAX_LEN CP_ETYPE_MAX_LEN=0x600
write 0x00200008 0x600

#### MFP_EFP_KEY_TABLE (59 bytes)
#### enable DA, SecTAG_Status and direction in the comparison
#### EFP Rule 0:KEY_VALID=1 DA=0xAABBCCDDEEFF SecTAG_Status=0 (non-macsec packet) DIRECTION=1
(Egress)
#### Please note EFP rules can be put in any place of the table which is shared with IFP. The entry#
is 1 of MFP rules
write 0x08200001
0x01860000_00000000_00000000_00000000_00000000_00000000_03ffffff_fffffe00_00000000_00000000_000000
00_00000000_00000000_0000AABB_CCDDEEFF

#### MFP_EFP_ACTION_TABLE (4 bytes)
#### EFP Rule 0 ACTION:POLICY=0x16 TCI_C=1 TCI_E=1 TCI_V=0 SCI_INDEX=0
#### packets will be classified to be the controlled port packets and loop up the 1st entry of
EGRESS_SC_INDEX_TABLE
#### Please note EFP actions can be put in any place of the table which is shared with IFP. The
entry# is 1 of MFP actions
write 0x09200001 0xB6000000

###############################################################################
# Secure Channel 0 (SC0) for Egress
###############################################################################
#### EGRESS_SC_INDEX_TABLE (9 bytes)
```

```
#### VALID=1 CIPHER_SUITE_PROTECTION=1 SCI=0xAACC0000 ALWAYS_INCLUDE_SCI=1 USE_ES=0 USE_SCB=0 AN=0
write 0x0C100000 0x00A4_00000000_AACC0000

#### EGRESS_SA_ATTRIBUTE_TABLE for the Egress Encryption (16 bytes)
#### VALID=3 NEXT_PN=1 Loopback=1
write 0x09100000 0xC8000000_00000001_00000000_00000000

#### EGRESS_SA_KEY_TABLE for the Egress Encryption (16 bytes)
#### SAK=0x5927
write 0x09108000 0x00000000_00000000_00000000_00005927

##########################################################
#### Step 9. Start sending the traffic in both directions
##########################################################
```

Appendix B: Fixed-Latency Configuration

```
###########################################################################################
##### Step 1. Program SP_MASTER_CTRL to enable cut-through mode
#####             IngressStoreForward = 0
#####             EgressStoreForward = 0
###########################################################################################
modify 0x03100100 IngressStoreForward = 0, EgressStoreForward = 0

###########################################################################################
##### Step 2. Program MPORT_EGRESS_TS_DELAY and MPORT_INGRESS_TS_DELAY to constant delay
###########################################################################################
if ( 10G )
{
    write 0x02100034 180
    write 0x02100035 180
}
else if ( 1G )
{
    write 0x02100034 300
    write 0x02100035 300
}
else if ( 100M )
{
    write 0x02100034 700
    write 0x02100035 700
}
else {
    write 0x02100034 1420
    write 0x02100035 1420
}

###########################################################################################
##### Step 3. Program XMAC register XMAC_MACSEC_CTRL and UNIMAC register MAC_CNTRL on both
#####          the line-side and the system-side.
#####             MACSEC_TX_LAUNCH_EN = 1
#####             MACSEC_TX_CRC_CORRUPT_EN = 1
###########################################################################################
if ( 10G )
{
    write 0x01100024 0x3         # XMAC line-side
    write 0x01101024 0x3         # XMAC system-side
}
else {
    write 0x001002c5 0x3         # UNIMAC line-side
    write 0x001012c5 0x3         # UNIMAC system-side
}
```

Appendix C: Procedure to Set Up Large MTU in MACsec

```
##############################################################################################
#### The following is the procedure to set up MTU to 16206 (0x3F4E) bytes in the MACsec core
##############################################################################################
#### Step 1. Program SP_INGRESS_MTU0_1, SP_INGRESS_MTU2_3, SP_INGRESS_MTU4_5, SP_INGRESS_MTU6_7
####          MTU = 0x3f4e
##############################################################################################
write 0x0310012e 0x3f4e3f4e
write 0x0310012f 0x3f4e3f4e
write 0x03100130 0x3f4e3f4e
write 0x03100131 0x3f4e3f4e

##############################################################################################
#### Step 2. Program SECY_CFG_CNTMAXSIZE register
####          CNTMAXSIZE = 0x3f4e
##############################################################################################
write 0x03100004 0x00003f4e

##############################################################################################
#### Step 3. Program SP_MTUx registers (egress)
####          MTU = 0x3f4e
##############################################################################################
write 0x04100906 0x00003f4e
write 0x04100907 0x00003f4e
write 0x04100908 0x00003f4e
write 0x04100909 0x00003f4e
write 0x0410090a 0x00003f4e
write 0x0410090b 0x00003f4e
write 0x0410090c 0x00003f4e
write 0x0410090d 0x00003f4e

if ( 10 Gbps )
{
    ##########################################################################################
    #### Step 4A. Program XMAC_CTRL registers
    ####          RX_EN = TX_EN = 1
    ##########################################################################################
    write 0x01100000 0x0000000000000003  # line-side
    write 0x01101000 0x0000000000000003  # system-side

    ##########################################################################################
    #### Step 5A. Program XMAC_TX_CTRL registers
    ####          TX_PREAMBLE_LENGTH = 0x8
    ####          AVERAGE_IPG = 0xC
    ####          PAD_THRESHOLD = 0x40
    ####          CRC_MODE = 0x3 # CRC mode is determined by the inputs pins on the system side
    ##########################################################################################
    write 0x01100004 0x000000100000c803 # line-side
    write 0x01101004 0x000000100000c803 # system-side
    ##########################################################################################
    #### Step 6A. Program XMAC_RX_MAX_SIZE registers
    ####           RX_MAX_SIZE = 0x3fe4
    ##########################################################################################
    write 0x01100008 0x0000000000003fe4  # line-side
    write 0x01101008 0x0000000000003fe4  # system-side
}
else {
    ##########################################################################################
    #### Step 4B. Program COMMAND_CONFIG registers
    ####           NO_LGTH_CHECK = 1
    ####           ENA_EXT_CONFIG = 1   # speed/duplexity determined by external pins
    ####           PAUSE_FWD = 1        # forward pause frames to user application
    ####           PROMIS_EN = 1
    ####           RX_EN = 1
    ####           TX_EN = 1
    ##########################################################################################
    write 0x00100202 0x01400093         # line-side
    write 0x00101202 0x01400093         # system-side

    ##########################################################################################
    #### Step 5B. Program FRM_LENGTH registers
    ####           MAXFR = 0x3f4e
    ##########################################################################################
    write 0x00100205 0x00003f4e         # line-side
    write 0x00101205 0x00003f4e         # system-side
}
```

Appendix D: MACsec Terminology

- **Association Number (AN):** A number that is concatenated with the Secure Channel Identifier to identify a Secure Association.
- **Bounded receive delay:** A guarantee that a frame will not be delivered after a known bounded time, in the case of protocols designed to use the MAC Service this is typically assumed to be less than two seconds.
- **Bridged Local Area Network:** A concatenation of individual IEEE 802 LANs interconnected by MAC Bridges.
- **Cipher Suite:** A set of one or more algorithms, designed to provide any number of the following: data confidentiality, data authenticity, data integrity.
- **Common Port:** An instance of the MAC Internal Sublayer Service used by the SecY to provide transmission and reception of frames for both the controlled and uncontrolled ports.
- **Canonical Format Indicator (CFI):** A 1-bit field. If the value of this field is 1, then the MAC address is in non-canonical format. If the value is 0, then the MAC address is in canonical format. It is always set to zero for Ethernet switches. CFI is used for compatibility between Ethernet and Token Ring networks. If a frame received at an Ethernet port has a CFI set to 1, then that frame should not be bridged to an untagged port.
- **Controlled Port:** The access point used to provide the secure MAC Service to a client of a SecY.
- **Cryptographic key**: A parameter that determines the operation of a cryptographic function such as:
  - The transformation from plain text to cipher text and vice versa
  - Synchronized generation of keying material
  - Digital signature computation or validation.
- **Cryptographic mode of operation:** Also referred to as mode. An algorithm for the cryptographic transformation of data that features a symmetric key block cipher algorithm.
- **Data integrity:** A property whereby data has not been altered in an unauthorized manner since it was created, transmitted or stored.
- **Initialization vector (IV):** A vector used in defining the starting point of an encryption process within a cryptographic algorithm.
- **Integrity:** See data integrity.
- **Integrity check value (ICV):** A value that is derived by performing an algorithmic transformation on the data unit for which data integrity services are provided. The ICV is sent with the protected data unit and is recalculated and compared by the receiver to detect data modification.
- **Key:** See cryptographic key.
- **Key management:** The generation, storage, distribution, deletion, archiving, and application of keys in accordance with a security policy.
- **MACsec Management Interface (MMI):** The interface between a protocol entity in a system and the system management, providing for the exchange of parameters with other system entities that are not attached to the service access points used and provided by the protocol entity.
- **MAC Security Entity (SecY):** The entity that operates the MAC Security protocol within a system.
- **MAC Security TAG (SecTAG):** A protocol header, comprising a number of octets and beginning with an EtherType, that is prepended to the service data unit supplied by the client of the protocol, and is used to provide security guarantees.

- **MAC service data unit (MSDU):** A sequence of zero or more octets that compose the data to be communicated with a single MAC Service request or indication.
- **Man-in-the-middle attack:** An attack on the authentication protocol run in which the attacker is positioned between the claimant and verifier so that the attacker can intercept and alter data traveling between the claimant and verifier.
- **Master key:** A secret key that is used to derive one or more cryptographic keys that are used directly to protect data transfer.
- **Message authentication:** If the message arrives authenticated, the cryptographic guarantee is that the message was not modified in transit and that the message originated from an entity with the proper cryptographic credentials.
- **Mode:** See cryptographic mode of operation.
- **Multipoint:** Involving or potentially involving more than one participant in the role of receiver, or in the role of transmitter, in a single data transfer or set of related data transfers.
- **Nonce:** A non-repeating value, such as a counter, used in key management protocols to thwart replay and other types of attack.
- **Packet number (PN):** A monotonically increasing value used to uniquely identify a MACsec frame in the sequence of frames transmitted using an SA.
- **Plaintext key:** An unencrypted cryptographic key.
- **Port Identifier:** A 16-bit number that is unique within the scope of the address of the port.
- **Point-to-Point Protocol (PPP):** The PPP originally emerged as an encapsulation protocol for transporting IP traffic over point-to-point links. PPP also established a standard for the assignment and management of IP addresses, asynchronous (start/stop) and bit-oriented synchronous encapsulation, network protocol multiplexing, link configuration, link quality testing, error detection, and option negotiation for such capabilities as network layer address negotiation and data-compression negotiation. PPP supports these functions by providing an extensible Link Control Protocol (LCP) and a family of Network Control Protocols (NCPs) to negotiate optional configuration parameters and facilities. In addition to IP, PPP supports other protocols, including the Novell® Internetwork Packet Exchange (IPX™) and Cisco® DECnet.
- **Priority Code Point (PCP):** A 3-bit field which refers to the IEEE 802.1p priority. It indicates the frame priority level from 0 (lowest) to 7 (highest), which can be used to prioritize different classes of traffic (such as voice, video, or data).
- **Protocol data unit (PDU):** A unit of data specified in a protocol and consisting of protocol information and, possibly, user data.
- **Secret key:** A cryptographic key used with a secret key cryptographic algorithm that is uniquely associated with one or more entities and should not be made public.
- **Secure Association (SA):** A security relationship that provides security guarantees for frames transmitted from one member of a CA to the other members. Each SA is supported by a single secret key, or a single set of keys where the cryptographic operations used to protect one frame require more than one key.
- **Secure Association Identifier (SAI):** An identifier for an SA, comprising the SCI concatenated with the Association Number (AN).
- **Secure Association Key (SAK):** The secret key used by an SA.
- **Secure Channel (SC):** A security relationship used to provide security guarantees for frames transmitted from one member of a CA to the others. An SC is supported by a sequence of SAs thus allowing the periodic use of fresh keys without terminating the relationship.

- **Secure Channel Identifier (SCI):** A globally unique identifier for a secure channel, comprising a globally unique MAC Address and a Port Identifier, unique within the system allocated that address.
- **Secure Connectivity Association (CA):** A security relationship, established and maintained by key agreement protocols, that comprises a fully connected subset of the service access points in stations attached to a single LAN that are to be supported by MACsec.
- **Short Length (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 field.
- **Spoofing:** Claiming a fraudulent identity for purposes of mounting an attack.
- **Tag Protocol Identifier (TPID):** A 16-bit field set to a value of 0x8100 to identify the frame as an IEEE 802.1Q-tagged frame.
- **Uncontrolled Port:** The access point used to provide the insecure MAC Service to a client of a SecY.
- **VLAN Identifier (VID):** A 12-bit field specifying the VLAN to which the frame belongs. A value of 0 means that the frame doesn't belong to any VLAN; in this case the 802.1Q tag specifies only a priority and is referred to as a priority tag. A value of hex FFF is reserved for implementation use. All other values may be used as VLAN identifiers, allowing up to 4094 VLANs. On bridges, VLAN 1 is often reserved for management.
- **Wiretapping:** An attack that intercepts and accesses data and other information contained in a flow in a communication system. The term is used to refer to reading information from any sort of medium used for a link or even directly from a node, such as a gateway or subnetwork switch. Active wiretapping attempts to alter the data or otherwise affect the flow; passive wiretapping only attempts to observe the flow and gain knowledge of information it contains.

Appendix E: Acronyms and Abbreviations

Table 23: Acronyms and Abbreviations

AES	***Advanced Encryption Standard***
AN	Association Number
CA	Secure Connectivity Association
CFI	Canonical Format Indicator
CRC	Cyclic Redundancy Check
CTR	Counter mode
DA	Destination Address
EAPOL	EAP over LANs (Extensible Authentication Protocol Over LANs)
EPON	Ethernet Passive Optical Network
ES	End Station
FCS	Frame Check Sequence
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
Gbps	Gigabit per second (1 Gbps is equivalent to 1 000 000 000 bits per second)
ICV	Integrity Check Value
ISS	Internal Sublayer Service
IV	Initialization Vector
KaY	MAC Security Key Agreement Entity
LACP	Link Aggregation Control Protocol
LAN	IEEE 802 Local Area Network
LLC	Logical Link Control (IEEE Std 802.2)
LLDP	Link Layer Discovery Protocol
LMI	Layer Management Interface
MMI	MACsec Management Interface
MAC	Media Access Control
MACsec	MAC Security Protocol
Mbps	Megabit per second (1 Mbps is equivalent to 1 000 000 bits per second)
MIB	Management Information Base
MPDU	MACsec Protocol Data Unit
MSDU	MAC Service Data Unit
MSTP	Multiple Spanning Tree Protocol
NESSIE	New European Schemes for Signatures, Integrity, and Encryption
NIST	National Institute of Standards and Technology
OLT	Optical Line Terminator
ONU	Optical Network Unit
PAE	Port Access Entity
PDU	Protocol Data Unit

Table 23: Acronyms and Abbreviations

AES	Advanced Encryption Standard
PN	Packet Number
PCP	Priority Code Point
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RSTP	Rapid Spanning Tree Algorithm and Protocol
SA	Secure Association
SAI	Secure Association Identifier
SAK	Secure Association Key
SC	Secure Channel
SCB	Secure Channel Broadcast
SCI	Secure Channel Identifier
SecTAG	MAC Security TAG (8 or 16 byte MACsec header)
SecY	MAC Security Entity (The entity that operates MAC Security protocol)
SL	Short Length
SNMP	Simple Network Management Protocol
TPID	Tag Protocol Identifier
VID	VLAN Identifier

Broadcom® Corporation reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design.

Information furnished by Broadcom Corporation is believed to be accurate and reliable. However, Broadcom Corporation does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Connecting everything®

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Phone: 949-926-5000
Fax: 949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com