IEEE 802.1 Security MACsec Privacy

Don Fedyk – dfedyk@labn.net

Note

- This is an initial meeting slide deck that was presented in the meeting.
- There is a follow-up slide deck that represents a more accurate view of the decisions.

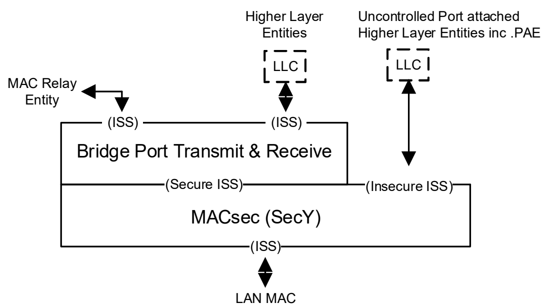
Work Items

- MAC priv SHIM
  - Location/Data path
  - Priorities
  - Working Format
- PrY Shim operation
- Interoperability scenarios
- AE YANG Model
- Towards a first draft
- Other?

Terminology – work in progress

- MAC Privacy Protection
- MAC Privacy Protection Entity PrY
- MPDU - MACsec Protocol Data Units
- MPPDU – MAC Privacy PDU – strawman term
- MPPCI – MAC Privacy PDU Component Identifier –strawman term
- Privacy Channel – Similar to Secure Channel for MACsec
- Aggregation frame, (Super-frame) – synonym for MPPDU

MACsec Shim adding PrY

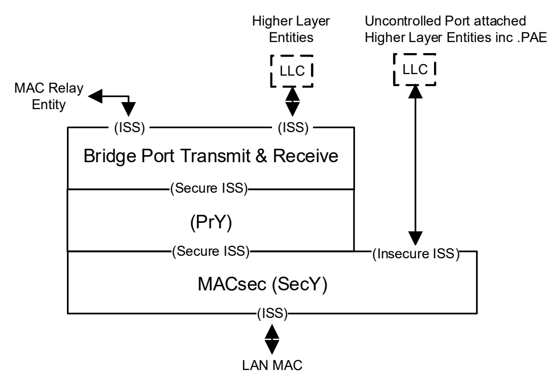


IEEE 802.1AE Figure 11-5— VLAN-unaware MAC Bridge Port with MACsec

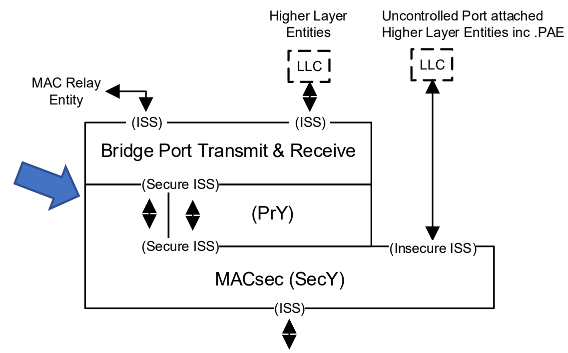
IEEE 802.1AE today

Main Assumption

All Secure traffic uses PrY



Alternative Some Secure traffic uses PrY

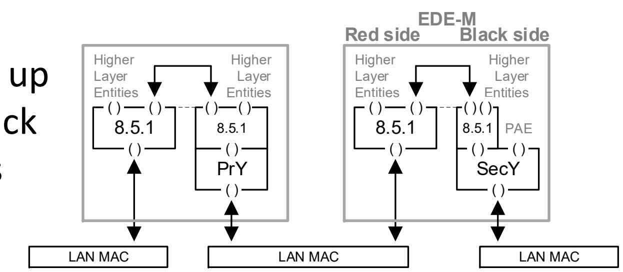


Can be achieved by using separate relays each with there own MACsec

PrY shim by itself on separate a relay

Small Distance
In same switch,
collocated
device.

Move up
so Stack
Aligns



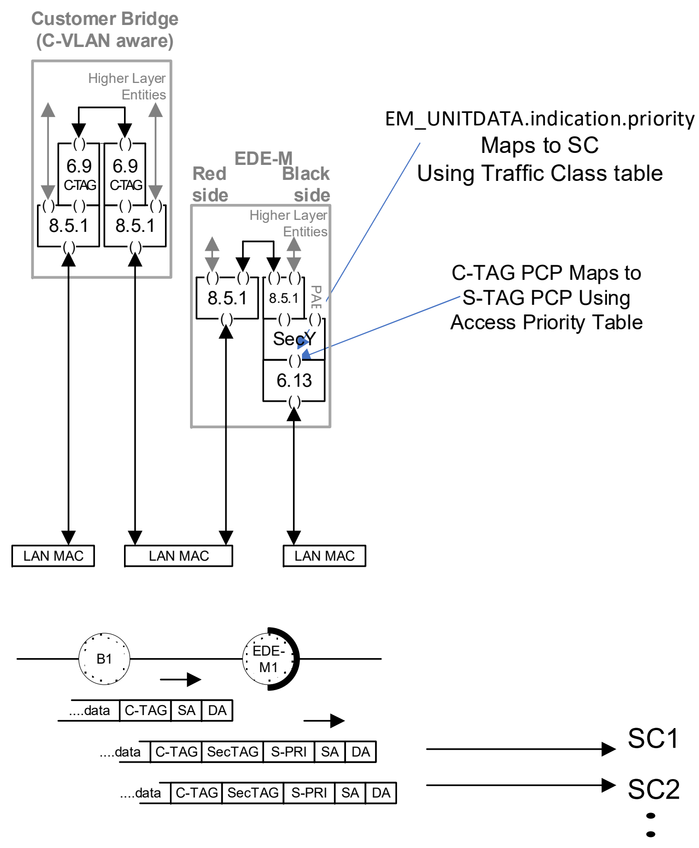
Allows full reuse of EDEs

All traffic
through the
PrY Shim

Mixture of PrY
and other
traffic through
the PrY Shim

Priority 802.1AE

- Currently there are two Mappings
  - Secure Channel (SC) by Traffic class from EM_UNIDATA.indication.priority
  - VLAN tag PCP& DE on SC frame by using Access priority mapping.
- MACsec is packet by packet this ordering is both withing the frame and outside the frame.
- It allows controlling frame ordering by SC and within an SC.



Priority and Ordering 802.1AEdk

With Aggregation we can have EM_UNIDATA.indication.priority map to a privacy channel (PC) but:

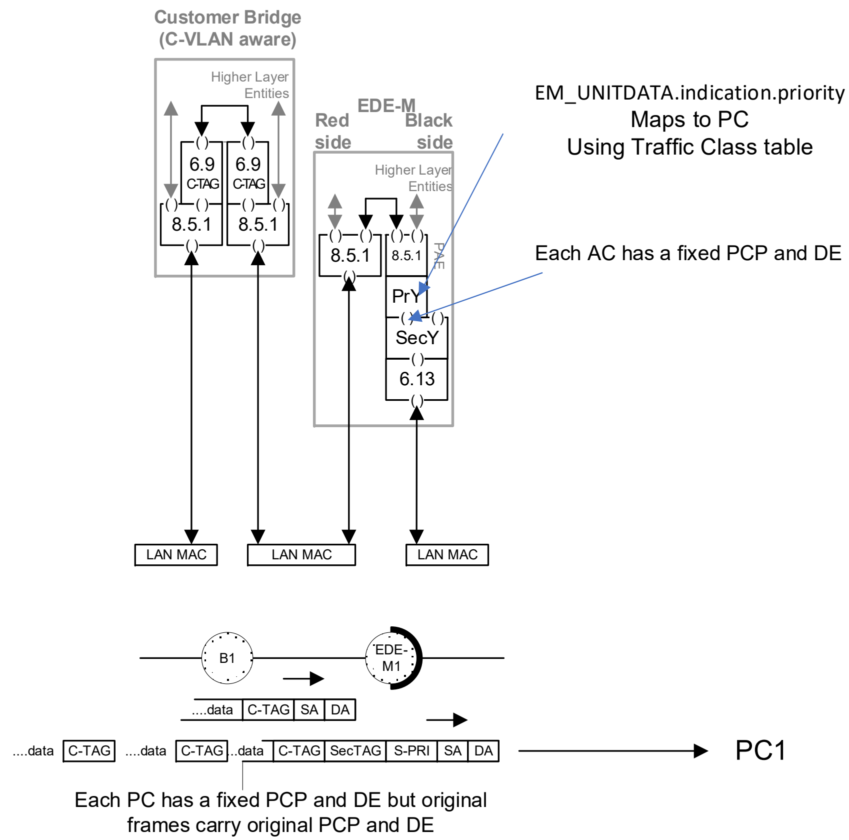
- If MAC privacy is frame-by-frame no aggregation – that would be OK
- But for MAC Privacy aggregation uses a fixed bandwidth to compute frame intervals. If we have multiple aggregated channels, we loose bandwidth efficiency because the bandwidth must be divided over the channels or we compromise privacy with variable rates.
- We have a second option that we can choose priority mapping into the aggregated frame. This uses one privacy channel, but frames are packed by priority into the frame. (both full and fragmented)
- If we use fragmentation, then higher priority (numerically lower) frames can preempt fragments within and privacy channel.

Big question is how many Traffic classes to support?

- Minimum 1
- Maximum?

Also how many priority levels into a single privacy channel?

- Minimum 2
    - With two you can always map all to one.
- Maximum ?



Multiple Priorities in an aggregated Frame.

- Goal: To keep scheduling of frames as simple as possible.
- Proposal: Use simple priority-based scheduling.
- Higher priority frames are always packed in aggregation frames first.
  - Provided there is room remaining in the frame.
  - Subject to head of the line blocking – high priority can block lower priority frames forever. Should ensure the rate of high priority is less than PC bandwidth by some margin.
- This argues for one level of preemption in the frame.
  - More levels complicates the buffering and scheduling.

Controls per traffic class.

- Only allow 1 mode per PrY instances
  - What if we allowed a mode per traffic class

  Possible but – Configuration on a Per PC basis
  - TC 0 variable size frames on PC 0
  - TC 1 Fixed frames minimum rate PC 1
  - TC 2-3 Fixed frames variable rate PC 2
- In above PC 0 still has to share bandwidth with PC 1 and PC 2.
- Bandwidth is still sub optimal.
- Priority mappings become per PC
- See how it would be configured
- Note the above could be supported with the current format as long as the TCs are unique across all PCs. In other words a receiver will be able to decode frames whether TC are on one PC or assigned to specific PCs.

Two privacy channels

- By separating frames into PCs with different PCP & DE bits MACsec can map to two SCs and/or support different scheduling for frames.
- When privacy channels are fixed size this causes inefficient use of the fixed frames with fixed timing.
- Fixed frames with fixed timing is essential to privacy.
- But traffic mapping to the PCs is not bandwidth aware so one PC may be sending padded frames while the other is congesting.
- Propose allow minimum 2 PCs for the case of variable sized frames (no aggregation). Discussion Point.

Fragmentation

- Frame format is illustrated later (a format that supports a large number fragments).
- Fragmentation occurs for several reasons:
  1. A frame is larger than the aggregation frame.
     - Must fragment or drop.
  2. A frame is larger than the remaining space in the aggregation frame.
     - May fragment - improves efficiency and delay.
  3. A maximum fragment size that is smaller than the aggregation frame is imposed to force fragmentation of any frame larger than the fragmentation size. This facilitates frame preemption of lower priority frames by higher priority frames.
  4. Minimum size Fragment (64 bytes) except for Aggregation frame end.
  5. Minimum size Aggregation frame (256 bytes?).

Fragmentation recommendations

Single PC with one priority : All TCs map to one class

- Only fragment packets when aggregation frame is > x bytes and < frame size.
  - X proposed minimum is 7 bytes. (Fragment header 6 bytes + one data byte).
  - Implementations can be less aggressive

Single PC with preemption: High and low priority

- Fragment low priority at some nominal maximum fragment size.
  - Frame is broken up even if it would fit completely.
  - Implementations can vary this to suit because the frame is self describing.
- Strictly Prioritize high priority frames before any low priority frames or fragments.
  - Implementation Scheduling has flexibility.
- Fragment high priority frames if the frame will not fit in the remaining PC frame.
  - Implementations can choose.

Do not fragment option

- For a receiver that does not support reassembly.
- Implications
  - All frames must be less that the aggregated frame size + overhead.

  For large frames this has impacts:
  - Forced to wait for a large aggregation frame if the space is too small.

  For small frames (typically high priority) impact is small.
  - Small frames may block larger frames by using up enough space to prevent lower priority frame admission.
  - Implies that single priority should be used in the PC since small high priority frames could waste aggregation space if they were allowed to fill the aggregation frame first.

Other Fragmentation considerations

- Once mapped to an PC
  - All frames within a Traffic class must be ordered
  - Ordering between Traffic classes is flexible
    - Could choose to fill a frame with a lower priority fragment when a higher priority frame won't fit in the remaining aggregation frame.
- Fragments are assumed to be received in order per PC &Traffic class
  - If a out of order fragment is received in a traffic class, the frame can be discarded (or we must wait how long?)
  - If a fragment is lost – this is an out of order fragment indicated by the sequence number gap.
  - A fragmented frame can only be complete if all the fragments initial bit, last bit and all fragments in sequence have been received.

Late addition of frames

- No Data condition:
  - Idle frames carry padding.
- Late addition allows a frame to be added as long as the current PC frame has space left even if transmission of that frame has begun.
- Explicit PADs of a minimum length are always filling frames unless the remaining length is less the minimum pad length.
  - Explicit PAD allows the addition of a frame after some padding
  - A receiver must process all explicit pads in case there is a frame or a fragment somewhere in the frame.
- Implicit pads are used when there is no data or the bytes left are less than the Explicit pad minimum length.
  - Implementations have flexibility here.

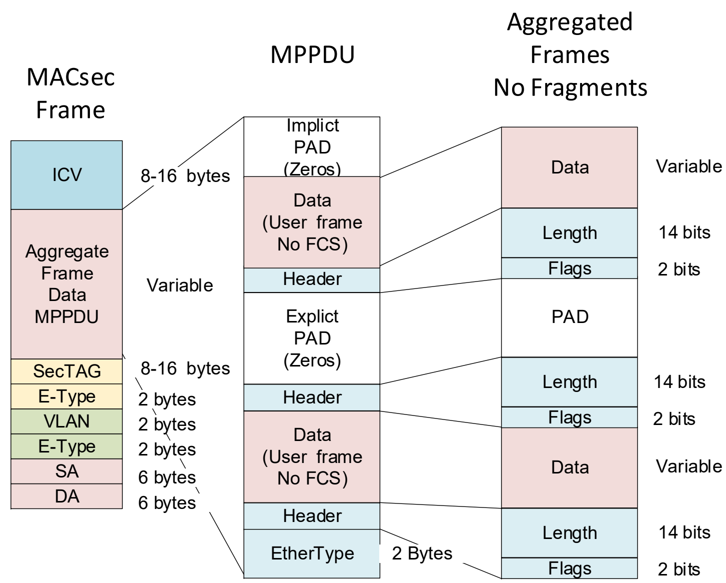
Explicit PAD size tradeoffs

Frame Bytes/ Speed bits	100M	1G	10G	100G	400G	1T
512	128	256	256	512	512	512
1024	128	256	512	512	512	1024
1500	128	256	512	512	512	1500
3000	256	512	512	1000	1000	1000
6000	256	512	512	1000	2000	3000
9000	256	512	512	1000	3000	3000

A table like this Numbers TBD

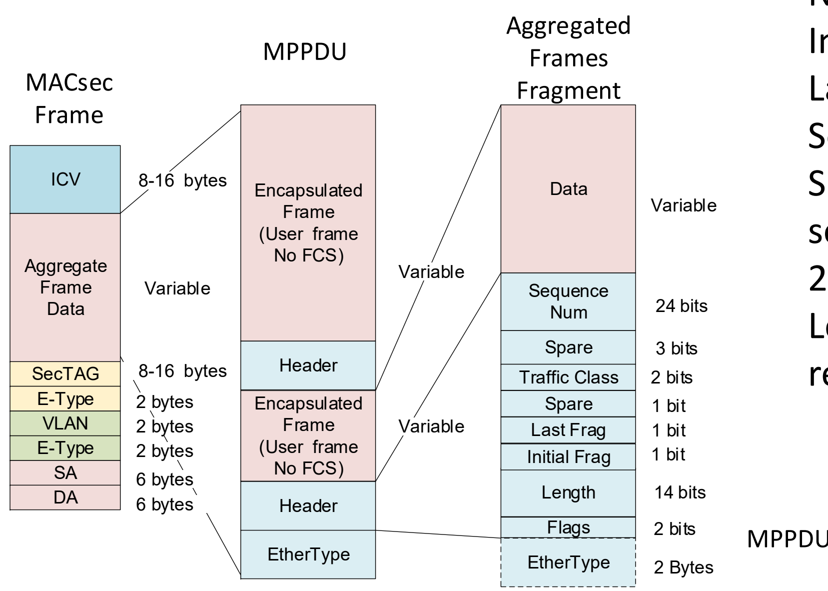
Trade off delay versus work interval.
Standard should allow configuration flexibility – implementations can choose.

MAC Privacy Packet Data Unit (MPPDU) No fragments



DA	Destination Address
SA	Source Address
E-Type	Ethernet Type
VLAN	Virtual Local Area Network
SecTAG	Security TAG

MPPDU with Fragments

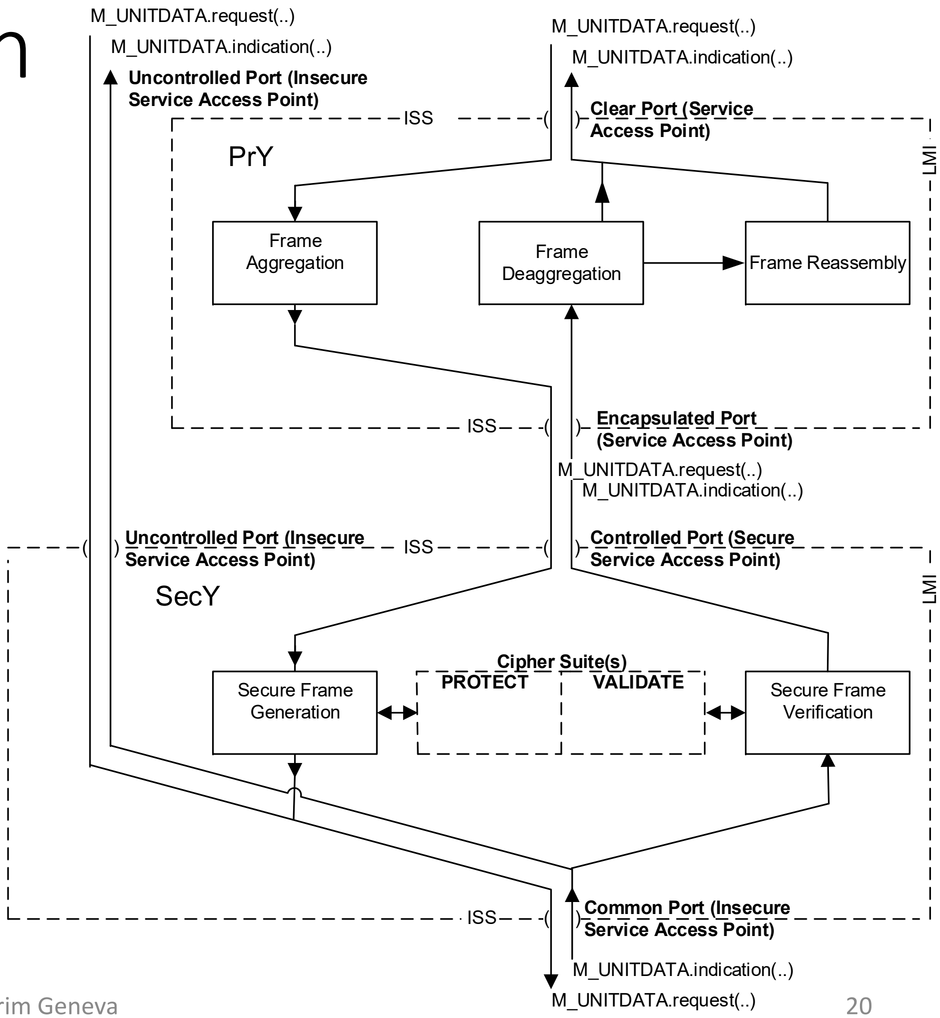


DA	Destination Address
SA	Source Address
E-Type	Ethernet Type
VLAN	Virtual Local Area Network
SecTAG	Security TAG

New format
Initial Fragment Flag
Last Fragment Flag
Sequence number per fragment
Suggest 3 bits spare added to sequence Number
2 bits traffic class
Leave 1 spare but for future? Or relegate to sequence number?

MAC PrY Shim operation

- Encapsulated Port address
  - (None) Same as Common Port
  - When remote
    - Unicast SA and DA mac ?
    - VLAN ? ?
    - Multicast ?



PrY Management

- Privacy modes – enabled/disabled
- Fragmentation – enabled/disabled
  (Remote side must be Fragmentation capable – Need a way to exchange this)
- Frame-rate-mode
  - variable-size-frames
  - fixed-size-minimum-rate
  - fixed-size-fixed-rate
- Maximum-reassembly-time
  - Time to wait for fragments
- Max-per-sec-bitrate
  - Rate in bps
- Min-per-sec-bitrate
  - Rate in bps
- Frame Size in octets
  - Maximum frame size (MPPDU size + PC frame header)
- Maximum-aggregation-time
  - Maximum time to wait to fill an aggregation frame
- Explicit Pad size

Frame Interval Timing Interval fixed-size-minimum-rate

$$\text{Interval fixed-size-minimum-rate} = \frac{\text{Min-per-sec-bitrate}}{\text{Frame Size in bits}}$$

QueueDelay = <total number of octets in the queue> * 8 / < MPPDU Rate in bps>

If Queue Delay > Maximum Aggregation time Then:

$$\text{Interval fixed-size-minimum-rate} = \frac{\text{Max-per-sec-bitrate}}{\text{Frame Size in bits}}$$

Until IQueue Delay < Maximum Aggregation time

Frame Interval Timing fixed-size-fixed-rate

$$\text{Interval fixed-size-fixed-rate} = \frac{\text{Max-per-sec-bitrate}}{\text{Frame Size in bits}}$$

QueueDelay = <total number of octets in the queue> * 8 / < MPPDU Rate in bps>

If Queue Delay > Maximum Aggregation time, then:

Discard Frames until Queue Delay < Maximum Aggregation

Pry Management for Priorities (discussion)

- Maximum Fragment size
  - Could be same as Explicit PAD size.
- Strict Priority
  - Traffic classes serviced by priority
- Fragmentation on all priorities?
  - Should each priority have a fragmentation option? Suggest No
  - There is no restriction from the format point of view.
  - Efficiency is better if fragmentation is supported.
  - Implementations are free to put local restrictions.

Interoperability Capabilities

- Fragmentation Capability
  - Both sides of a PrY must agree on the capability to fragment frames.
  - How is this exchanged?
    - Could leverage KaY for collocated situations
    - Could exchange in band messages
- Traffic class Capability
  - Both sides must support the signaling of all traffic class bits. (2 bits, 4 values)
  - A PrY should not reorder frames within a traffic class
  - There is no requirement that a PrY must order frames between traffic classes
  - Fragmentation is per traffic class and buffers must be capable of handling the traffic classes.

Interoperability Capabilities

- Number of privacy channels
  - Minimum 1
  - Maximum 4 ?
    - Usefulness for aggregated frames diminishes after 1
    - For variable sized frames 4-8 makes sense
    - If we used Traffic class = 4 and Traffic class per privacy channel sum must be less than 4 then Traffic class could map to