Ethernet Data Encryption devices

Mick Seaman

An Ethernet Data Encryption device (EDE) is a frame forwarding device with two physical ports that uses IEEE 802.1Q, 802.1AE, and 802.1X standards to provide integrity and confidentiality for frames forwarded on network hops open to attack. One port (the 'red side') receives and transmits frames that are not protected, while frames transmitted and received on the other (the 'black side') are protected by MACsec. In some cases the required EDE functionality goes beyond that easily determinable from the existing standards, and I have been asked to comment on what can or should be done. The purpose of this note is to canvas additional 802.1 opinion, looking for problems and issues that need to be explicitly addressed.

---

1. Introduction

Much as there is a desire for EDEs to be a simple as possible, the exclusion of the unnecessary has long been a principle of 802.1 standardization. Time and again experience and analysis has shown the need to include what many would wish to leave out of a start-from-scratch-simple-as-possible design with a single object in view. In specifying EDEs, as an identifiable class of devices, we are therefore less concerned with starting from scratch than we are with clarifying how standards are to be used. In particular there have been significant additions to IEEE 802.1Q (PBBNs, for example) not covered by IEEE 802.1AE-2006's discussion of systems and networks. The following should help to clarify this point.

MACsec is added to an 802.1Q bridge by adding an MACsec ISS shim ('SecY') to one or more interface stacks. When communication to the nearest bridge (of any type) is to be protected, the SecY is naturally added at the bottom of the stack: MACsec protection occurs as a last (in the 802.1 world) data frame transformation prior to transmission, and MACsec validation of the frame and recovery (from encryption) of the original frame as a first step after reception. The group MAC address used for authentication and key agreement exchanges that provide the shared secret keys used by these communicating SecYs is the 'reserved address' allocated by 802.1Q to this restricted scope—so these exchanges are not confused by the fruitless participation of distant systems.

When the 'hop' to be protected by MACsec crosses a PBN or PBBN the necessary arrangements are not so clear. 802.1AE-2006 included the necessary specification for port-based interfaces to PBNs, but not tagged service interfaces. Section 5 of my note MACsec hops[1] discusses some of these additional use cases. This note expands upon but does not repeat that discussion, which is should be read first by anyone who is not intimately familiar with 802.1AE and the discussions that took place during its development.

Much of the discussion concerns protecting traffic between customer systems as it transits a provider network. 802.1Q uses the terms 'customer' and 'provider' in a quite general sense, denoting administrative or architectural separation of operational concerns. In this note the provider side of an EDE is always the 'black' side, where customer traffic needs to be protected, and the customer side is always the 'red' unsecured side.

EDE functionality is described using 802.1's general component-oriented architectural philosophy: as far as possible new systems comprise components that have already been defined. A 'black box' principle applies—if a new system can be specified as equivalent to a validly connected network of existing systems and components, then it is a valid system—for who can tell whether the 'box' contains the new system or that network fragment? At the same time, the usual rule that only the externally observable behavior of the system matters applies[2].

[1] http://www.ieee802.org/1/files/public/docs2013/ae-seaman-macsec-hops-0626-v03.pdf

[2] And as usual network (system) management necessarily ensures, by using an externally visible operational model of the system to structure and explain the effects of network management, close adherence to the 'behaves as if' rule, much as that gives wide implementation flexibility.

2. Taxonomy

Given the desire to maximize the use of existing standards, two types of EDE and EDE deployment can be distinguished. In the first, MACsec capability is integrated within an existing (or replacement) bridge so that bridge becomes the EDE. In the second, EDEs are added in a way that is largely transparent to the existing bridges in the network. The distinction, though not necessarily clear cut, is readily apparent when protecting a customer's connectivity across a PBN providing tagged service interfaces. A solution of the first 'integrated' type adds MACsec to the (bottom of) the Provider Edge Port of the PEB's C-VLAN component. See PEB3 in Figure 1 (borrowed from MACsec hops). A solution of the second 'transparent' type, takes the existing interface (C-tagged or S-tagged) to the PBN as a given. In both cases the functionality provided by each EDE is readily identified by the type of the tag that it uses on the unsecured ('red side') and the type that it emits on the secure ('black side'). The 'integrated' EDE just discussed (PEB3 in Figure 1) is an EDE-CS. It accepts C-tagged frames from the customer, and transmits S-tagged frames into the provider network. The two 'transparent' EDE types are EDE-CC and EDE-SS respectively.

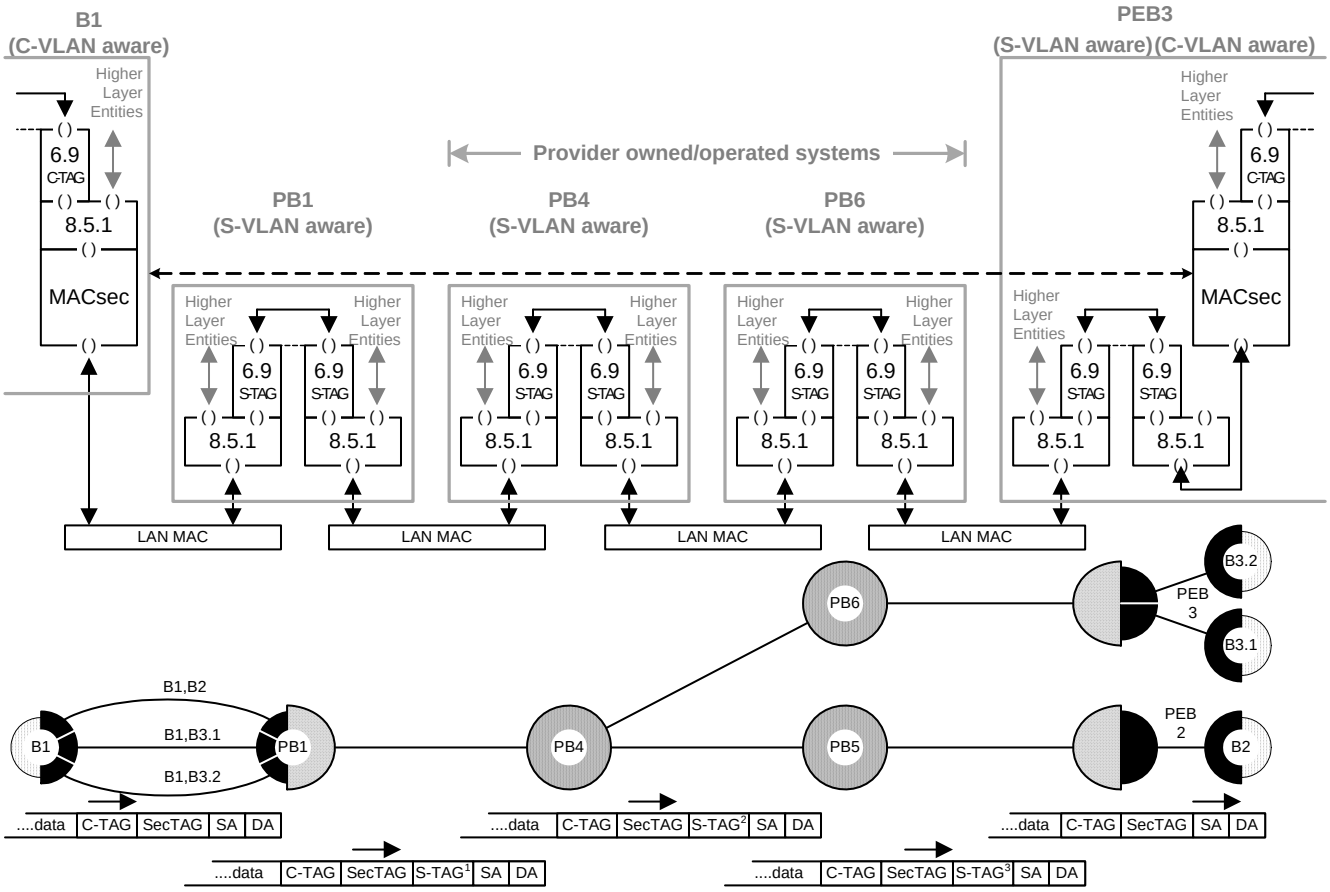

Figure 1—Interface stacks and paths for PBN tagged service interfaces

The useful EDE types are (so far as I understand the possibilities):

- EDE-T (6)

  A TPMR based EDE, that does not recognize or add tags (other than the MACsec SecTAG on the black side). It peers with (connecting black side to black side) another EDE-T.

- EDE-M

  A two-port VLAN-unaware MAC Bridge, with MACsec on one (black side) port. Differs from an EDE-T in the extent to which it is visible to other protocols. Can peer with one or more other EDE-Ms, or with VLAN-aware or unaware bridges

- EDE-CS

  A Provider Edge Bridge (PEB) with integrated MACsec capability, as discussed above.

- EDE-CC (4)

  A transparent EDE, as introduced above, that is intended for connection to a PBN's C-tagged or port-based service interface. It can peer with (black to black, over the PBN) one or more EDE-CCs, or (with minor configuration restrictions) EDE-CSs or PEBs with MACsec capability.

- EDE-SS (5)

  A transparent EDE, as introduced above, that is intended for connection to a PBN's S-tagged or port-based service interface. It can peer with (black to black, over the PBN) one or more EDE-SSs, or (with configuration restrictions) EDE-CSs or PEBs with MACsec capability.

  An EDE-SS can also be used to protect frames that have been I-tagged by a Backbone Edge Bridge (BEB) prior to submitting them to the S-tagged interface of a PBN or PBBN. In this configuration the original customer MAC addresses will be carried in the I-TAG and will be protected (encrypted if MACsec confidentiality has been selected). In this context it might be called an EDE-BB.

- EDE-SIS

  This simply a name for a PEB packaged together with an EDE-SS in a restricted configuration.

Additional EDE types might be identified, but there seems little value to simply labeling any bridge (particularly bridges with multiple physical ports) as EDEs if they implement MACsec in the 'usual' place, i.e. at the bottom of the media independent portion of a port's interface stack. There are characteristics that (it is hoped) broadly distinguish EDEs, with their two physical port restriction, from MACsec capable bridges in general. After discussing these characteristics (3), this note reviews the architecture and use of each of the EDE types named above, starting with the EDE-CC, to contrast its functionality with that of the EDE-CS introduced by reference to PEB3 in Figure 1.

3. EDE characteristics

While this note holds out little prospect of an EDE constituting a significant reduction of the functionality of a standard bridge (see 1), the restriction to two physical ports allows simplifications already described in 802.1Q, such as not learning from MAC source addresses. In particular a 'transparent' EDE does not have to participate as a network node in some protocols (see MACsec hops 6.1). More significantly, the two port restriction and transparency allows functionality that is widely used but not standardized (complex filters, traffic management etc.) and the associated management controls to be off-loaded to other devices. This helps to reduce the tendency to make the EDE just another platform for running arbitrary network code—making the validation of the code that it does run a feasible task[3].

4. EDE-CC

An EDE-CC secures frames submitted to a C-tagged provider service interface, and is modeled as comprising two C-VLAN components, connected by internal ports (See Figure 2). In the simplest configuration (see below for an extended discussion) there is one internal connection for each C-VID in use, so MACsec keys can be agreed for the particular set of provider service access points supported by the provider service instance selected by that C-VID.

Since the C-VID, and possibly all of the tag control information (TCI)—VID, priority, and discard eligible indicator, has to be integrity protected to ensure the attached customer network functions correctly it has to be included after the SecTAG, even if the PBN is providing a C-tagged interface. This protected C-TAG will also be encrypted if confidentiality is being provided for the rest of the data in the frame. The copy (in the simple configuration) of the C-TAG prepended to the frame by the provider side component allows the provider's PEB to select the correct provider service instance (and will be reflected in the S-TAG).

A frequent suggestion is that this outer C-TAG should be omitted, and the PEB instructed to 'skip over' the SecTAG and read the following C-TAG. This suggestion ignores a number of important points:

[3] A discussion of the ways in which unprivileged code can compromise the security provided by theoretically separate and contained privileged code is well beyond the scope of this note, but if you have never focused on this subject you would be surprised.

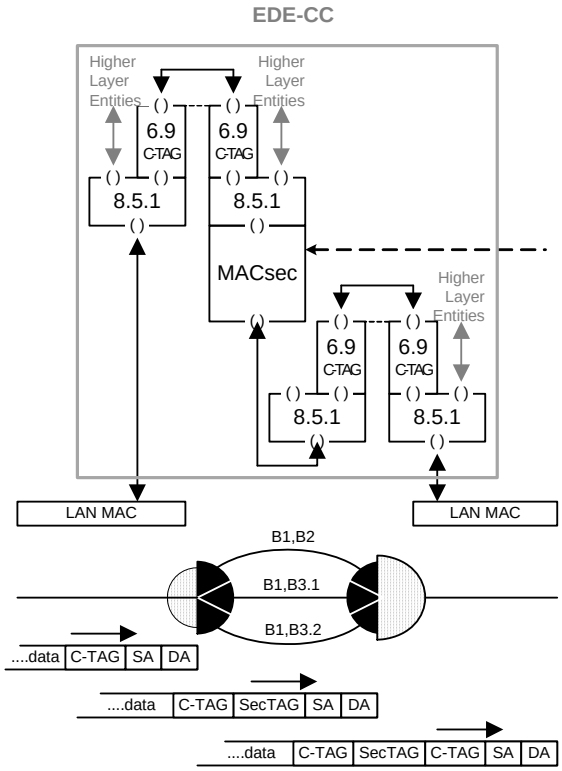

Figure 2—EDE-CC architecture

a) The PEB now has to understand the format of the SecTAG (at a minimum) though it has no need to process it. The peer PEB that delivers the frame back to the customer also has understand how to skip and act on the inner TAG.

b) If confidentiality is being provided, this has to be 'offset' so the C-TAG remains in clear. However it proves very difficult to balance the desire of networking equipment to see as much of the frame as possible and of the customer to maintain confidentiality. The latest MACsec Cipher Suites do not support a confidentiality offset. While the Default Cipher suites included the capability it has not (as far as I am aware) been used, and was designed to expose those very fields that need to be protected when https is used (see MACsec hops).

c) This 'skipping' approach is not possible for an EDE-SS or EDE-CS, unless used by all the provider equipment in the network. Even then it is unduly restrictive. The network can reasonably police and mark down the priority bits in an S-tag, and set the discard eligible bit. Configuring the MACsec validation process to selectively validate fields in tags that have to be integrity protected is a complex task. A simpler and robust approach is to provide the network with its own prepended copy that can then be modified, translated, or removed as needed. The last of these operations (removal) is permitted by 802.1Q. It supports hub-and-spoke deployments where a central site uses a tagged interface to select amongst service interfaces to simple remote sites that only require a port-based interface.

Figure 3 shows the interface stack along a network path through EDE-CCs. The peer relationships between the MACsec SecYs (marked M), and between the attached customer bridges, B1 and B2, is highlighted.

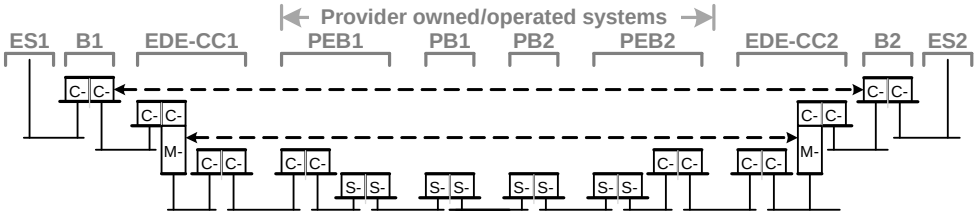

Figure 3—Using EDE-CCs attached to a provider network

In simple configurations, as mentioned above, the C-TAG presented to PEB by the EDE-CC will be the same as that received from the attached customer bridge: the provider-side C-VLAN component uses the configured PVID of each internal port to assign the C-VID of the prepended tag and the priority component of the tag is conveyed by the internal medium. This approach, which makes use of the existing properties of the component, also allows a number of C-VLANs to be bundled together and can hide their C-VID values from the PEB, just as a PEB can bundle C-VIDs into S-VIDs. Neither of C-VLAN components needs to learn from the source addresses of frames, and optimization already provided by 802.1Q (clause 8.7.2) so this bundling is independent of any address allocation conflicts.

5. EDE-SS

The architecture of an EDE-SS is very similar to that of an EDE-CC—so much so that it is not worth duplicating Figure 2. C-VLAN components are simply replaced by S-VLAN components, and C-tags by S-tags.

The EDE-SS might be used by a customer that wished to provide his own C-VLAN to provider service instance multiplexing, using his own (separate) PEB. Figure 4 shows the interface stack along a network path through EDE-SSs. .

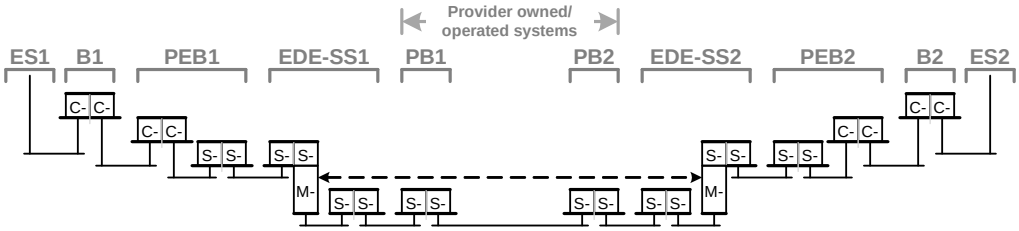

Figure 4—EDE-SSs attached to a provider network

Of course the provider could equally support the connection between attached EDE-SSs with a Provider Backbone Network (PBBN), encapsulating the customer MAC addresses in backbone MAC addresses. The customer could also use an EDE-SS with his own Backbone Edge Bridge (BEB) to encapsulate (and encrypt) his own addresses over a PBN or PBBN. Figure 5 shows the end to end path across a PBN. Note that there is no difference between a B-tag and an S-tag except context. .

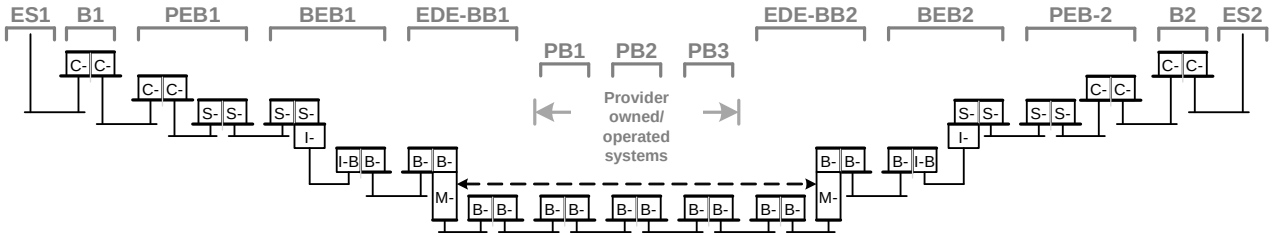

Figure 5—Customer address encapsulation across a PBN

802.1Q does not currently specify a C-tagged customer interface to a PBN. It should be possible to do that with a small amount of standards work, allowing the PEB and BEB functions provided by the customer in Figure 5 to be performed by a single two component system. Equally any unnecessary complexity of such single system PEB-BEB combination using current standards need only appear in the its use of existing, standard, management—the implementation only has to exhibit the required external behavior. It does not actually have to forward a frame internally in a way that corresponds to the model of that behavior.

6. EDE-T

<<Not much to say, though a short discussion of the group address to be used by the PAE might be in useful, as might a reference to the LLDP, CFM, and link aggregation arrangement. A discussion of the reachability or how to reach the EAP AS when multiple layers of MACsec are in place might also be useful. Somewhere in this note as a whole the distinction (introduced by the ESS specification) between Pre-shared Keys and Pre-placed Keys also merits discussion.>>

7. EDE-M

<<Discussion of whether it is worth identifying this as a separate class. Discussion of group addresses, visibility etc.>>