Uncontrolled Ports

Mick Seaman

This note explores the relationships between the .1X ‘Y’ function’, the secure media that support it, the transmission of key agreement or key exchange frames, and the ‘Y’ provided by the P802.1AE MAC Security Entity (the ‘SecY’). An architectural extension that provides multiple secure virtual ports for desktop devices without changing the P802.1AE MACsec specification is described.

1. The .1X ‘Y’

The ‘Y’ function provided by .1X is best described as a shim†1 that provides a secured Controlled Port and an unsecured Uncontrolled Port, using a Common Port provided by an underlying service.

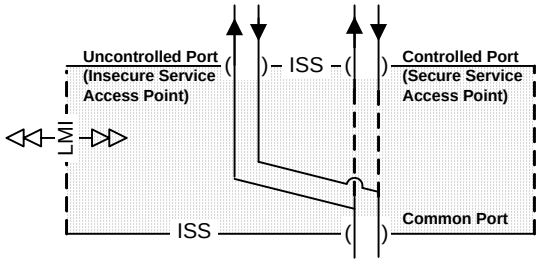

Figure 1—The .1X ‘Y’ function

2. Using .1X with .11i

When .1X-2004 is used with .11i it is the latter that provides the security by cryptographically protecting frames. The .1X shim simply controls the connectivity provided to the Controlled Port. The .11 MAC provides a portValid status parameter†2 for the Common Port. When portValid is set the MAC is cryptographically protecting frames, so communication through the Controlled Port is permitted. The .1X shim controls the port’s MAC_Operational status to signal that connectivity to its client.

This is not the only way that the ‘Y’ function of Figure 1 can be implemented†3, and .1X also describes other functions†4. To identify the ‘Y’ implementation used with .11i and described in clause 6 of .1X specifically, this note uses the term PAC (Port Access Controller)†5.

The .1X/.11 interface stack is illustrated in Figure 2 and uses a combination of these approaches.

The .1X Port Access Entity (PAE) includes (or interfaces to) an EAP component that acquires a Pairwise Master Key (PMK)†6. Proof of possession of the PMK by both the .1X Supplicant and the Authenticator constitutes proof of mutual authentication.

The PMK is handed (through the LMI) to key agreement functions within the .11 MAC. These are somewhat complicated by their historical development and their support for group keys amongst the ‘virtual’ Common Ports that represent each .11 station’s association with an access point. However their overall purpose is simply to agree session keys (similar to the MACsec’s SAKs), to start using these keys to protect transmission, and to then assert portValid.

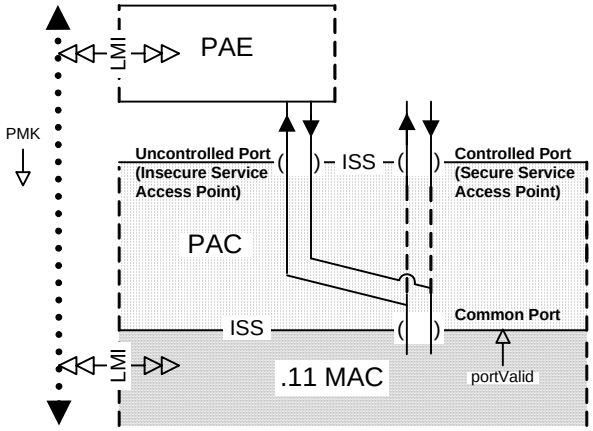

Figure 2—.1X with .11

3. A general problem

This model appears to work fine with .11i, but there is a general problem in applying it in general to cryptographic sublayers that protect frames.

Since frames from the PAC’s Uncontrolled Port are not distinguished from Controlled Port frames at the Common Port, they are also protected when portValid is true. So the Controlled Port, if operational, provides connectivity to the same set of peers as the Uncontrolled Port. This is convenient if the goal is to avoid discrepancies between the connectivity provided to those entities attached to Uncontrolled Port and those attached to the Controlled Port when both can communicate. However it is not possible for an entity attached to the Uncontrolled Port to explore additional or alternate connectivity while the Controlled Port is providing service. Moreover any fault with the underlying service can affect attempts to use the PAC’s Uncontrolled Port to control its parameters. Protocol exchanges that configure the underlying service either have to be carried out by the entity providing that service, or the Controlled Port has to be made inoperable (either immediately or after initial attempts to acquire new parameters have failed) so that unconstrained connectivity is made available through the PAC’s Uncontrolled Port.

If the underlying service does not provide connectivity checking and discovery functions then the result of the architecture in Figure 2 is that the system’s connectivity is a property of its history, rather than the current network configuration, for an indeterminate period. This sort of behavior usually makes debugging network management operations difficult.

---

†1 802.1X-2004 is missing some technical detail, but the intent is clear.

†2 This is a status parameter of the interface (like MAC_Operational) rather than information communicated through the LMI because the parameter value needs to be synchronized with requests and indications at the interface. LMI information is only loosely coupled to service invocations.

†3 The SecY provides an alternate implementation, as will be described.

†4 Such as the PAE.

†5 PAE would have been the obvious name, but is already taken. We are going to need a name, so better suggestions would be welcome.

†6 A certain amount of data is cryptographically bound to the PMK and used with it, the entire package is called the PMKSA (802.11i 8.4.1.1).

4. Using .1X with .1AE MACsec

The MAC Security Entity (SecY) specified in P802.1AE provides both Controlled and Uncontrolled Ports. The SecY only applies cryptographic protection and validation to frames from users of the Controlled Port. This means that:

a) frames sent and received by users of the Uncontrolled Port are exactly the same as they would be if MAC Security was not present, this provides continued interoperability with existing systems

b) the SecY's Uncontrolled Port can always be used to discover potential connectivity

c) a PAE attached to the Uncontrolled Port may have more choices than are possible with 802.11i, where the association to be secured has already been chosen.

The proposed .1X/MACsec interface stack is illustrated in Figure 3. The relationship of the SecY and the KaY is described in P802.1AE Clause 10 and Figure 10-2.

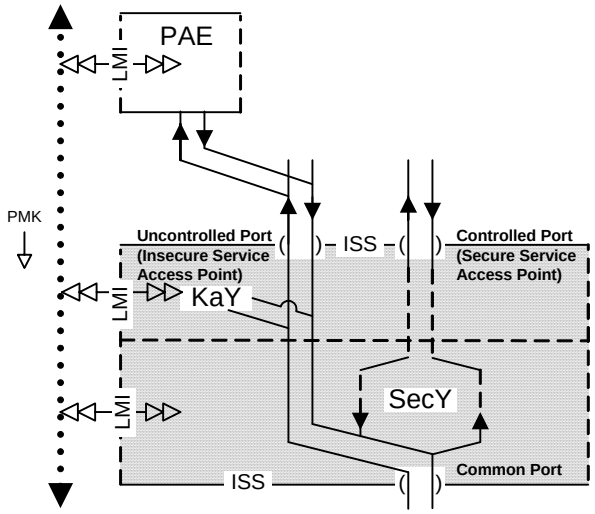

Figure 3—.1X with MACsec

5. MACsec extensions and .1X

P802.1AE recommends the use of a single CA (secure connectivity association) on a physical LAN. This is so network connectivity does not change with security. If infrastructure connectivity varied with security, then security would be eventually turned off and kept off just to stabilize the network topology.

However there is intermittent interest in the use of multiple CAs on a single physical LAN to support connectivity to desktop devices. In this case a bridge or other intermediate system supporting such devices would create a number of on demand "virtual ports" each participating in a separate CA. The traffic for different CAs could be distinguished by a multiplexing field (such as an EPON LLID) and assigned by the bridge†1. Equally it is possible to use the SCI in the SecTAG to perform the multiplexing. This option allows a fairly compact encoding. Frames transmitted by each desktop device could use the bit in the SecTAG which uses the source MAC address to define the SCI field — this keeps the SecTAG at 8 octets and requires no additional multiplexing field. In the other direction the full 16 octet SecTAG would be required (presuming that a bridge is used to attach the desktop devices to the rest of the network) but again there is no additional multiplexing field.

The interface stack in each desktop device is close to that shown in Figure 4.

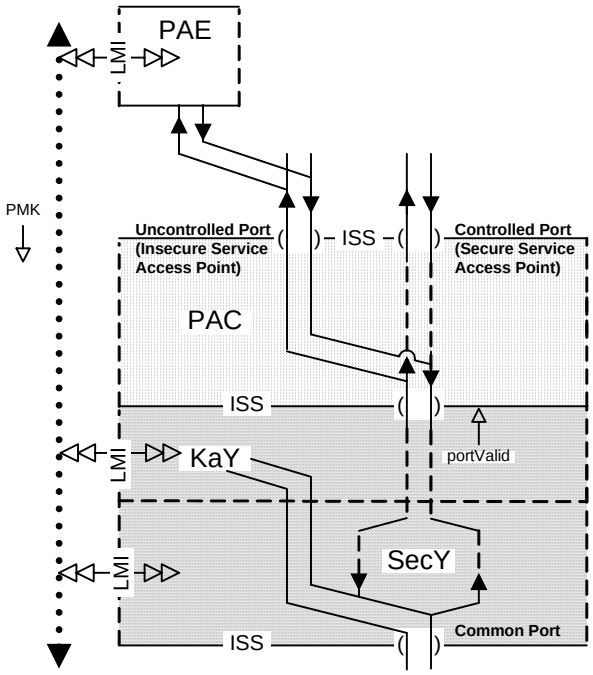

Figure 4—Nearly but not quite

This stack is very similar to that for .11i with .1X, and is nearly but not quite right. It complicates the SecY since that now has to be capable of passing insecure traffic through its Controlled Port, and requires the operation of a SecY to SecY control protocol that is synchronous with the data traffic in order to avoid any exposure around the setting of portValid. Further care has to be taken to ensure that PAEs can talk even if the PN number space is exhausted — without risking the reuse of a secure nonce. A better solution for point-to-point CAs is to use a stack closer to Figure 3, but with KaY adding a tag header to couple the Uncontrolled and Controlled Port communications. The stack for a desktop device is shown in Figure 5.

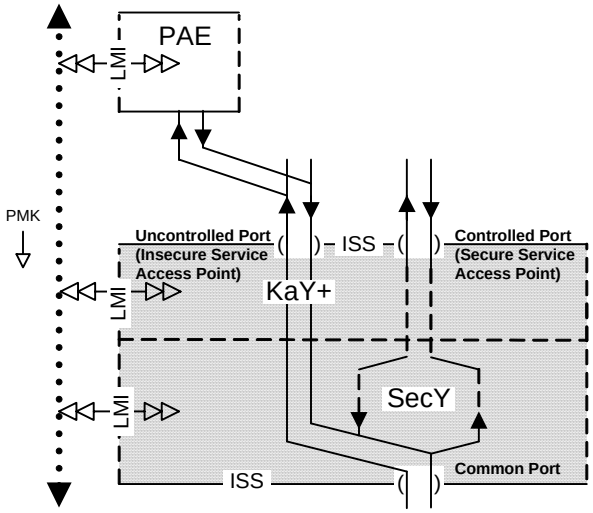

Figure 5—.1X with MACsec and a virtual port

†1Using a VLAN identifier for such a purpose is not recommended because a number of the attached devices might need to be attached to the same VLAN once authorized. Overloading the VLAN ID complicates this an introduces a need for tag translation and management of pools of spare VLAN IDs.

A good way to make this stack work with no chance of confusion as to how to pair the traffic for an Uncontrolled Port with that for its Controlled Port is for the KaY+ function to use a variant of the SecTAG which the SecY will not attempt to process to label the Uncontrolled traffic. My favorite is to use the currently illegal combination 1,1 for the E and C bits. The only impact on the SecY behavior envisaged in P802.1AE/D3.0 should be to separate the management counts for such frames from those for other illegal frames.

The corresponding interface stack in the system with the multiple virtual ports is depicted in Figure 5.

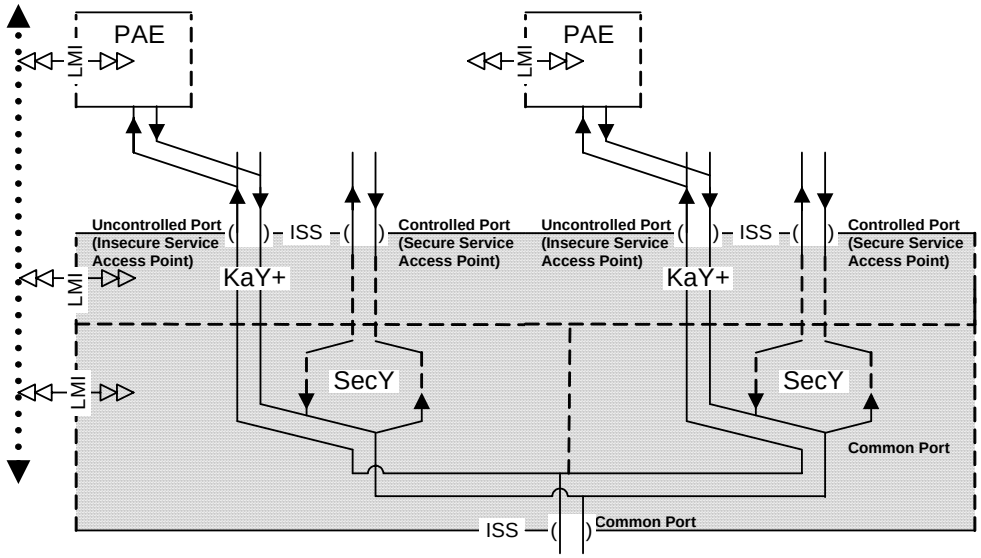

Figure 6—System with multiple secure virtual ports

The figure really comprises a single KaY, with some of the functionality of the single port system duplicated within it, and multiple independent SecYs, each with the functionality specified in P802.1AE/D3.0. Each virtual port presents an Uncontrolled Port/Controlled Port pair.

6. Another desktop alternative

The previous section assumes that separate controllable virtual ports are really wanted for desktop devices, and that the only way that these devices will be permitted to talk involves frame relay through the bridge or other ‘head end’ system. An alternative which can be more readily deployed with existing desktop switches is to support the formation of a group CA, probably comprising the desktop system and an IP phone registered to the user of the desktop. Then the PC can configure the phone, or serve as a local directory/call monitor/answering machine etc.