PKI Planes in Cisco APIC-EM (1.4.x) Tech Note

Revised: February 28, 2017,

PKI Planes in Cisco APIC-EM 1.4.x

Effective management of the Cisco APIC-EM PKI requires an understanding of the mechanisms that secure various types of network connection. This topic concerns itself primarily with PKI-based controller and device connections, describing other kinds of connections only for purposes of comparison and contrast with PKI-secured connections. Detailed descriptions of non-PKI connections are outside the scope of this discussion.

The Cisco APIC-EM provides PKI-based connections in several distinct PKI planes.

- **Controller PKI Plane:** HTTPS connections in which the controller is the server in the client-server model, and the controller's server certificate secures the connection. The controller's server certificate can be self-signed (default) or issued by an external CA (recommended.)
- **Device PKI Plane:** DMVPN connections between devices in the control plane of the network, bilaterally authenticated and secured by the device ID certificates of both devices that participate in the connection. These DMVPN tunnels secure data-plane traffic as it travels between network devices. A private CA provided by the APIC-EM controller (the **Device PKI CA**) manages these certificates and keys.
- **Grapevine Service PKI Plane:** The grapevine root manages this internal PKI plane that secures communications between Grapevine services in a multi-host cluster; the Grapevine Service PKI Plane is not externally accessible, so it is not discussed further here.

Table 1: PKI Planes in Cisco APIC-EM

	Authentication	Encryption	Use Case
Controller PKI Plane: external caller initiates connection to controller			
HTTPS	Caller presents username+password or caller presents service ticket; Controller presents server certificate	Yes	REST client, including Cisco Network Plug N Play (PnP) mobile app or Cisco Prime Infrastucture
HTTPS	One-way: controller presents its server certificate	Yes	Cisco Network Plug N Play (PnP) provisioning workflows
Device PKI Plane: device-to-device connections			
DMVPN	Bilateral authentication via IKEv2 using device ID certificates/keys issued by the private CA that the APIC-EM controller provides (the **Device PKI CA**). Device ID certificates and keys secure device-to-device connections between IWAN-managed devices.	Yes	DMVPN connections between devices for the secure exchange of data-plane traffic
Grapevine Service PKI Plane: connections between grapevine services			
HTTPS	Connections between grapevine services	Yes	System use only. Not accessible to external callers.

In the default configuration of Cisco APIC-EM, the Device PKI CA is a root CA; there is no parent CA above it. This configuration of the Device PKI CA is known as **rootCA mode**. Note that rootCA mode applies ONLY to the Device PKI CA; the Device PKI CA has nothing to do with the Controller PKI plane.

Optionally, version 1.4.x of Cisco APIC-EM provides the ability for the Device PKI CA to use a CA certificate that has been issued by an external CA. This configuration of the Device PKI CA is known as **subCA mode**. Regardless of mode, the Device PKI CA never interacts directly with the external CA and no automated management of the Device PKI CA's CA certificate ever occurs. Again, regardless of mode, only the Device PKI CA manages the certificates and keys that secure device-to-device connections in the Device PKI plane; an external CA, if used, never has access to these certificates and keys.

In the rare event that an external CA revokes the CA certificate of the Device PKI CA, a user who has `ROLE_ADMIN` in scope `ALL` must replace this certificate manually. Note that doing so requires re-configuration of the cluster and manual deprovisioning of devices that use certificates and keys issued under the old CA certificate; there is no other workflow for replacing the Device PKI CA's CA certificate.

To understand subCA mode, and APIC-EM PKI implementation in general, simply remember two points that the remainder of this topic explores further:

- subCA mode affects ONLY the CA certificate of the Device PKI CA. It does not affect ANY other certificates or keys.
- The APIC-EM provides NO automated interactions with any external CA.

Overview of Secure Connections in Cisco APIC-EM v. 1.4.x

Two independent PKI planes (**Controller PKI Plane** and **Device PKI Plane**) secure two main categories of PKI-based connection. The APIC-EM controller also supports other types of secure connection that do not use PKI.

Note The grapevine root manages an internal PKI plane that secures communications between Grapevine services in a multi-host cluster; the **Grapevine Service PKI Plane** is not externally accessible, so it is not discussed further here.

PKI-Based Connections

All HTTPS connections to the APIC-EM use the **Controller PKI Plane**. Device-to-device connections use the **Device PKI Plane**, which is completely separate from the **Controller PKI Plane**.

Controller PKI Plane: Externally Initiated HTTPS Connections to the Controller

When the controller responds to a request for an HTTPS session, it is the server in a client-server model that uses PKI to secure the connection. In response to the request for an HTTPS session, the controller presents its server certificate. Therefore, externally initiated HTTPS connections to the controller take place in the **Controller PKI Plane**.

HTTPS requests can come from devices in the control plane of the network or they can come from NB REST API callers. The controller never initiates HTTPS connections to devices.

Device PKI Plane: DMVPN Connections Between IWAN-Managed Devices

A separate PKI plane secures the Dynamic Multipoint VPN (DMVPN) connections that IWAN-managed devices form amongst themselves for the secure exchange of data-plane traffic. This **Device PKI Plane** is managed by a private CA that the APIC-EM controller provides (the **Device PKI CA**.)

By default, the Device PKI CA runs as a root CA; in this mode (known as **rootCA mode**), the CA certificate of the Device PKI CA is the apex of the certificate chain for device certificates. Optionally, the Device PKI CA can be configured to use an externally issued CA certificate (so-called **subCA mode**), which subordinates the Device PKI CA to the external CA.

- In the default configuration (rootCA mode), an external CA cannot manage the certificates and keys that secure the Device PKI Plane.
- In subCA mode, a user who has `ROLE_ADMIN` in scope `ALL` must manually upload to the private CA a CA certificate issued by an external CA. In subCA mode, the private CA does not interact directly with the external CA, no automated management of the private CA's CA certificate occurs, and the external CA still cannot manage any of the certificates or keys that the private CA issues to IWAN-managed devices.

  While in subCA mode, if you use the same CA to manage the Device PKI CA's CA certificate and the controller's server certificate, the respective certificate chains of the Device PKI plane and the Controller PKI plane have a common ancestor, but no use case takes advantage of this ancestry. For more information, see Device-to-Device DMVPN Connections, on page 8.

Grapevine Service PKI Plane: Connections Between Grapevine Services

The grapevine root manages this internal PKI plane that secures communications between Grapevine services in a multi-host cluster; the **Grapevine Service PKI Plane** is not externally accessible, so it is not discussed further here.

Non-PKI Secure Connections

The controller also supports the following secure connections that do not use PKI.

Controller-Initiated Non-PKI Secure Connections to Devices

Controller-initiated secure connections to devices can use SSH or Authenticated SNMPv3. These connections are authenticated, but they do not use a CA; therefore, these connections do NOT take place in the Controller PKI Plane.

- **SSH from the controller:** When the controller initiates an SSH connection to a device, the controller presents a shared secret (username/password pair) and the device presents its public key to create a secure connection.
- **Authenticated SNMPv3:** Authentication-enabled SNMPv3 uses a shared secret to establish trust between the controller and the device. When the controller uses authentication-enabled SNMPv3 to initiate a connection to a device, it presents a username and password that a trusted administrator supplied out-of-band to both the controller and the device:

  The admin supplied credentials to the device by creating on the device a login account that the controller can use for discovery purposes.
  The admin supplied credentials to the controller by creating discovery credentials on the controller. These credentials enable the controller to supply a valid username/password pair to log in to the device for discovery purposes.

  If the device accepts the username and password that the controller presents, then the controller trusts the device. (Note that SNMPv3 can be configured not to authenticate the connection; if so, the connection is not secured and it is outside the scope of this discussion of secure connections. Optionally, authenticated SNMPv3 connections can also be encrypted.)

Externally Initiated Non-PKI Secure Connections to the Controller

SSH to the controller: When an external caller (such as an administrative remote terminal session) initiates an SSH connection to the controller, the controller presents its host public RSA or ECDSA key. Requests for SSH sessions come from administrators opening remote console sessions with the grapevine root. Network devices never initiate SSH connections to the controller.

Controller-to-Controller Secure Tunneling

APIC-EM controllers in a multi-host cluster can use a secure, encrypted channel for communicating amongst themselves. This communications infrastructure is not accessible by means of any API or user interface. This security mechanism is not PKI-based; it uses IPSec tunnels secured by private keys that the grapevine root manages. For more information, see "Configuring IPSec Tunneling for Multi-Host Communications" in the *Cisco Application Policy Infrastructure Controller Enterprise Module Administrator Guide*.

PKI Planes in Cisco APIC-EM 1.4.x

The APIC-EM maintains multiple separate PKI planes. Each PKI plane secures a particular set of connections:

- **Controller PKI Plane: Client-initiated HTTPS connections to the controller**

  When an external caller initiates an HTTPS connection to the controller, the controller presents its server certificate. Such connections include the following:

  - Logins to the APIC-EM GUI via HTTPS
  - Grapevine API calls (HTTPS on port 14141, redirected to port 443)
  - Invocations of the NB REST API via HTTPS

    When a NB REST API caller initiates an HTTPS connection to the controller to invoke a NB REST API or to download a file (such as a device image, a configuration, and so on) the controller (server) presents its server certificate to the caller (client) that requested the connection.

  Note that controller-initiated connections to devices do NOT take place within the Controller PKI Plane. Even if the connections use SSH or SNMPv3, no CA manages the keys involved, so the connection is not considered to be PKI-based. The controller may initiate connections to devices for purposes that include discovery, managing tags, pushing policy to devices, or interacting with devices on behalf of a REST caller. For compatibility with older devices, discovery can optionally use the TELNET protocol, which is insecure and therefore outside the scope of this PKI discussion.

- **Device PKI Plane: Device-to-device DMVPN connections**

  IWAN-managed control-plane devices form Dynamic Multipoint VPN (DMVPN) connections among themselves for the secure exchange of data-plane traffic. A private Certificate Authority (CA) provided by the Cisco APIC-EM (the **Device PKI CA**) provisions the certificates and keys that secure these DMVPN connections. The PKI broker service manages these certificates and keys as directed by an admin in the IWAN GUI or as directed by a REST caller that uses the `/certificate-authority` and `/trust-point` NB REST APIs.

  The private CA can run in rootCA mode (default) or subCA mode:

  - **In rootCA mode (default)**, the Device PKI CA is the root CA and there is no parent CA above it. Its CA certificate cannot be replaced. It cannot be a sub-CA or intermediate CA to any external CA.
  - **In subCA mode**, the Device PKI CA uses a CA certificate that was issued by an external CA. However, this relationship does not enable the external CA to perform any sort of automated management of any PKI items on the controller or on the device network.

- **Grapevine Service PKI Plane: Connections between grapevine services**

The grapevine root manages an internal PKI plane that secures communications between Grapevine services in a multi-host cluster; the **Grapevine Service PKI Plane** is not externally accessible, so is not discussed further here.

Regardless of whether the Device PKI CA operates as a root CA or as a sub CA, the following rules always apply:

- The Device PKI CA ***never*** interacts directly with the external CA.
- No automated management of the Device PKI CA CA certificate ever occurs. If the external CA revokes the CA certificate of the Device PKI CA, a user who has `ROLE_ADMIN` in scope `ALL` must learn of this out-of-band and must replace the subCA certificate manually.
- The external CA cannot manage the certificates and keys that the Device PKI CA issues to secure device-to-device connections between IWAN-managed devices. These certificates and keys are always managed ONLY by the Device PKI CA.
- The use of subCA mode does not alter the behavior of the `/certificate-authority` and `/trust-point` NB REST APIs. For example, if you use the NB REST API to revoke the CA certificate of the Device PKI CA, the controller does NOT call out to the external CA.

It is also important to understand that the Device PKI CA ***never*** manages the controller's server certificate.

For more information, see Device-to-Device DMVPN Connections, on page 8.

HTTPS Connections to the Controller

When an external caller initiates an HTTPS connection to the Cisco APIC-EM controller, that connection takes place in the PKI plane that the controller's server certificate secures (the Controller PKI Plane.) In this client-server model, the controller is the server that presents its certificate to the client (REST client or Web browser) to establish a trusted connection. (If the controller is behind a firewall, its gateway-proxy certificate participates in the trust chain.)

The certificate that the controller (or proxy) presents can be self-signed or CA-issued. If the certificate is CA-issued, a native Cisco caller may refer to the trustpool bundle to establish trust with the CA. A PnP-managed Cisco device downloaded the trustpool bundle from the controller as part of the Cisco Network Plug N Play (PnP) workflow that provisioned the device. A non-PnP Cisco device can also be configured to use the trustpool bundle. Non-Cisco devices or callers cannot use the trustpool bundle "as-is" but they can establish trust with trustpool CAs by means of their own certificate chain.

Upon establishing trust with the controller and any required CAs, the caller can use HTTPS to invoke NB REST APIs on the controller, such as those which provide one-time downloads of configuration files, certificates, keys, and so on. For example, the PnP mobile application may initiate HTTPS with the controller for the purpose of provisioning network devices, but the deployment of configuration files to the devices by the controller takes place over a controller-initated SSH connection that is NOT within the Controller PKI Plane.

Security-conscious callers typically would not connect to a controller that presents an expired or revoked server certificate, although it is possible for them to do so at their own risk.

Expiration of the Controller's Server Certificate

The APIC-EM controller or a network device does not need the assistance of a CA to determine whether a certificate has expired. The expiration date of the certificate is contained in the certificate itself; the device or controller simply compares the certificate's expiration date with current system time.

The APIC-EM controller warns administrators of the impending expiration of its server certificate. This warning appears in the controller GUI only. The controller provides no automated management of this certificate; a user who has `ROLE_ADMIN` in scope `ALL` must take explicit action to replace the server certificate before it expires. This administrator can use the **Controller Settings** panel in the GUI or the `/certificate` NB REST API to replace the server certificate. Similarly, this admin can use the GUI or the `/proxy-certificate` NB REST API to replace the proxy certificate.

Revocation of the Controller's Server Certificate

The controller's server certificate can be self-signed (default) or CA-issued (recommended.)

- A self-signed certificate can't be revoked in the true sense of the word: without the use of an external Certificate Authority, there is no mechanism for communicating the invalid status of the certificate to those who have established a trust relationship with that certificate. However, the self-signed certificate can be deleted or replaced, breaking the chain of trust that had been established with the old certificate.
- The CA-based revocation workflow applies to CA-issued certificates only. In this workflow, a trusted Certificate Authority may revoke a certificate, communicate the revoked status of that certificate to other members of its trust domain, and perhaps supply a valid replacement certificate.

The workflow that results from the CA-based revocation of the controller's server certificate varies according to the context in which the CA interacts. Use cases to consider include non-PnP devices, NB REST API callers, PnP-managed devices, the APIC-EM controller itself and invalidation of an intermediary CA certificate that is part of the trustpool bundle.

Non-PnP Devices and NB REST Callers

If configured to do so, a non-PnP network device can contact the appropriate CA to learn of the revocation of a CA-issued server certificate by means of the Certificate Revocation List (CRL) or Online Certificate Status Protocol (OSCP). However, if the device is not configured to perform a revocation cross-check with an external CA, the device cannot determine whether the external CA has revoked that certificate. As a result, the device may trust a certificate that an external CA has revoked.

Note This example describes non-PnP devices that communicate with an external CA, not PnP-managed devices that interact with the Device PKI CA. Non-PnP devices cannot communicate with the Device PKI CA, and the Device PKI CA does not accept OSCP requests.

A device performs a revocation cross-check with the external CA only when its CRL distribution point (CDP) points to the external CA. If the device is not configured to check a CRL or to issue an OSCP request to the external CA, the device simply checks its own internal truststore of valid and/or private CA root certs along with the expiration date of the server certificate that the controller presents to it. If the truststore contains stale revocation data and the certificate is not expired, it is possible for the device to trust a revoked certificate.

The most likely circumstance under which a controller's server certificate might be revoked would be an explicit request by the controller admin to the external CA to revoke the server certificate. The administrator might issue this request if the controller was stolen or if failed hardware was returned to Cisco without having been processed properly for return. In this situation, it is reasonable to assume that the admin issuing the revocation request would know of the revocation and would take steps to generate and install a new, valid, CA-issued replacement server certificate on the controller or on a replacement controller.

Devices configured to interact with the trusted CA that manages the server certificate should continue to work correctly upon installation of another valid, CA-issued server certificate on the controller. Devices that do not interact with the trusted CA might need to be updated manually as necessary to trust the new server certificate; until this update takes place, these devices may refuse to connect to the controller, and REST API requests that involve these devices may fail.

PnP-Managed Devices and IWAN-Managed Devices

PnP-managed devices and IWAN-managed devices never interact with any CA other than the APIC-EM private CA, even when the private CA runs in subCA mode. Therefore, PnP-managed devices cannot learn of the revocation of the controller's server certificate directly from an external CA. Hence, the status of the APIC-EM server certificate is of no direct consequence to such devices. These devices might respond to a change in status of the private CA server certificate, however, as described in Device-to-Device DMVPN Connections, on page 8.

Important The private CA that secures Distributed Multipoint VPN connections does not manage the controller's server certificate. Therefore, it cannot provide revocation status of the controller's server certificate. For more information, see Device-to-Device DMVPN Connections, on page 8.

APIC-EM Controller

Although a CA-issued server certificate can be installed on the APIC-EM controller, the APIC-EM controller itself does NOT interact directly with any external CA; therefore, it has no way to learn of the revocation of its server certificate by an external CA. Note, also, that the controller does not update its server certificate automatically under any circumstances. Replacement of an expired or revoked server certificate requires explicit action on the part of a user who has `ROLE_ADMIN` in scope `ALL`.

Intermediary CA Certificate in the Trustpool Bundle

Invalidation of an intermediary CA certificate in the trustpool bundle is a special case. When the trustpool bundle changes, the controller GUI does display a notifcation to users who have `ROLE_ADMIN` in scope `ALL`, and it provides a button that this type of admin can click to download and install a new trustpool bundle on the controller. However, the means by which network elements get the new trustpool bundle vary according to how the bundle was installed on those devices. Devices not managed by PnP cannot get the trustpool bundle from the controller, but they may be configured to download a new trustpool bundle from the Cisco cloud automatically. PnP-managed devices that got the trustpool bundle from the controller will continue to trust the controller's new intermediary certificate if it has a valid Root CA certificate. The same is true of devices not managed by PnP. Therefore, although best practice recommends manual update of devices with the new trustpool bundle in timely fashion, a change to an intermediary CA is not likely to cause an immediate problem.

Device-to-Device DMVPN Connections

IWAN-managed devices can form Dynamic Multipoint VPN (DMVPN) connections among themselves for the secure exchange of data-plane traffic. The Device PKI CA and the pki-broker service work together to provision the device ID certificates and keys that secure these DMVPN connections. The pki-broker service also exposes a NB REST API that can be used to manage these device ID certificates and keys manually.

When the Device PKI CA runs in rootCA mode (default), the Device PKI CA is not recognized by any external CA as an intermediate CA. Therefore, this internal CA is not a member of the trustpool (`ios.p7b`) bundle that the APIC-EM provides to devices in the Network Plug n Play provisioning workflow. External CAs in the trustpool bundle have no knowledge of the certificates that the controller's internal CA doles out to IWAN devices privately. Certificates issued by the Device PKI CA can be revoked manually by using the `/trust-point` NB REST API that the Cisco APIC-EM controller exposes.

Running the Device PKI CA in subCA mode requires the Device PKI CA to use a CA certificate that is signed by an external CA. However, subCA mode does not enable the Device PKI CA to interact with the external CA, and it does not provide automated management of the Device PKI CA's CA cert. If an external CA revokes the CA certificate of the Device PKI CA, the Device PKI CA cannot learn of this revocation because it never interacts directly with the external CA. Although the revocation of the CA certificate of the private CA invalidates the PKI broker's server certificate, which, in turn, invalidates all device ID certificates that the PKI broker issued, the APIC-EM provides no automated management of the Device PKI CA's CA cert; therefore, it is possible for devices to continue trusting the pki-broker server certificate even when the subCA CA certificate has been revoked by the external CA.

As a result, the use of subCA mode does not change the end-user-visible behavior of the private CA itself. The user who has `ROLE_ADMIN` in scope `ALL` must learn of the revocation of the CA certificate out-of-band and install a new CA certificate in the private CA. Note that revocation of the CA certificate is an extremely rare occurrence, and installation of a new CA certificate in the private CA is a non-trivial task. The controller does not provide a GUI or an API for replacing the subCA certificate. Once subCA mode is enabled, the only way to replace the CA certificate of the Device PKI CA is to do a complete reset that brings the controller back to the default

rootCA mode, and then subsequently redo the conversion to SubCA mode using the new subCA certificate. Before converting the controller back to subCA mode, you must remove all device ID certificates and keys issued to network devices under the previous configuration of the Device PKI CA. The devices must be taken off line before converting the controller to subCA mode with the new subCA certificate, and then all devices will need to be reprovisioned by the PKI broker service using the new configuration of the Device PKI CA.

IWAN-participating devices learn of the revocation of internal CA-issued device ID certificates by means of the CRL distribution point that is the private CA itself. When two devices attempt to create a DMVPN tunnel, they present device ID certificates to each other. To determine whether the certificate presented to it has been revoked, each device polls its CRL. Whenever a device ID certificate is revoked, the private CA generates a new CRL.

A private CA-issued certificate (which is used to secure DMVPN connections) is valid for one year from the date of issue (default) or until an administratively-set expiration date. IWAN-participating devices can attempt automated renewal of a private CA-issued certificate before the certificate actually expires. Expiration or renewal of a private CA-issued certificate generates PKI events that appear in the audit logs.

Intersection of the Device and Controller PKI Planes

The request for a certificate from the Device PKI CA could be viewed as the point at which the two PKI planes intersect, though in different contexts. This request from a device to the controller requires a trust relationship that the **controller's** server certificate guarantees; the controller's server certificate is NOT issued by the Device PKI CA. However, the payload of the response concerns the **device** ID certificate that the Device PKI CA issues to the device as part of the device-provisioning workflow. This device ID certificate protects the DMVPN connections that IWAN-participating devices form among themselves in the Device PKI Plane.

To understand these statements more clearly, let's examine one example of a workflow that intersects both PKI planes. The **POST** `/trust-point` request generates the certificates and keys necessary to enroll a specific device in the trust domain of the Device PKI CA. This trust relationship enables the device to participate in DMVPN connections with other devices that trust the Device PKI CA. Any caller can issue this request to enroll a device in the trust domain that the Device PKI CA manages. For example, you might create a custom REST application that enrolls devices, or an administrator might use this API to enroll devices manually, if necessary. Required arguments to this request identify a specific device (for example, by device serial number as well as its FQDN or IP address). The response to this request cannot be used to enroll any other device, and it is valid for a limited amount of time only.

For illustrative purposes, assume that an external caller has issued the **POST** `/trust-point` request, supplying a serial number, FQDN and IP address that identify a specific device to enroll. The controller receives this HTTPS request in the Controller PKI plane and forwards it to the pki-broker service, which invokes the Device PKI CA in the Device PKI plane to generate certificates and keys intended for use in the Device PKI plane. The response to the request then travels back to the caller in the Controller PKI plane.

Because generation of the certificates bundle completes asynchronously to the original request, the **POST** `/trust-point` response body indicates only whether the request was accepted; if the HTTP response is 202, then the response body contains a task ID that the caller can use to determine when the task has completed. When the **GET** `/task/{taskID}` response body indicates that the task has completed successfully, it contains an `id` element that can be passed as the `trustPointId` argument to the the **GET** `/trust-point/{trustPointId}/config` request. This request returns the information necessary to retrieve and install the PKCS12 bundle that the controller generated for this specific device. This request originates in the Controller PKI plane and returns a payload that enables access to the device ID certificate/key bundle that was generated in the Device PKI plane; upon installation of this bundle in the device, the device (and its certificate/key bundle) reside in the Device PKI plane.

Note The `/trust-point` API provides a number of ways to retrieve trustpoint IDs. For more information, see the REST API documentation.

Thus, the `/trust-point` NB REST API is an example of a device lifecycle-management operation that occurs in the **Controller PKI Plane** that the controller's server certificate protects. Although the **POST** `/trust-point` and **GET** `/trust-point/{trustPointId}/config` requests are protected by the server certificate, the PKI payloads of their responses concern

device ID certificates, keys and CRLs that protect transactions in the **Device PKI Plane**, such as DMVPN tunneling among IWAN devices themselves.

A similar context applies when a device in the Device PKI plane retrieves a new CRL from the Device PKI CA. The device sends a request to the controller, which requests a CRL from the Device PKI CA. Because the device has an existing trust relationship with the Device PKI CA, it can get the CRL from the Device PKI CA directly. One important difference, however, is that the device's request for the new CRL goes to the controller over HTTP rather than HTTPS. Because no certificate secures the connection, this request is not considered to occur in a PKI trust domain, but the workflow is similar in the sense that a request from the Device PKI plane to the controller results in the generation of a PKI construct that is used in the Device PKI plane.

Another example in which the controller acts as an intermediary between an IWAN-managed device and the Device PKI CA is the workflow in which the device's device ID certificate is due to expire and the device requests a new device ID certificate from the Device PKI CA. Again, the device issues an HTTP request to the controller, which instructs the Device PKI CA to generate a new device ID certificate for the device. In this case, however, the current device ID certificate on the device is still valid, so the Device PKI CA can install the replacement device ID certificate directly on the device. Once the new certificate is installed, the Device PKI CA generates a new CRL that revokes the old device ID certificate. Thus, this particular workflow includes two requests that originate in the Device PKI plane, travel to the controller, and result in the installation of a device ID certificate and CRL used in the Device PKI plane.

Summary: PKI Planes in Cisco APIC-EM v. 1.4.x

The following factors govern secure connections to the APIC-EM controller and secure device-to-device connections:

- **PKI for network control plane is completely separate from PKI for connections to APIC-EM controller**

  Two completely independent PKI planes separate the controller's interaction with external callers from device-to-device interactions in the network control plane. (Another PKI plane, the **Grapevine Service PKI Plane**, is not externally accessible.)

  - **Device PKI Plane: DMVPN connections between network devices**

    Devices managed by Cisco Network Plug N Play (**PnP-managed devices**) make device-to-device DMVPN connections autonomously among themselves for the secure exchange of data-plane traffic. A private Certificate Authority (CA) provided by the APIC-EM controller (the Device PKI CA) manages the certificates and keys that secure these connections. The Device PKI CA can operate as a root CA (default) or as a subordinate to an external CA (subCA mode.)

    In rootCA mode, the CA certificate of the Device PKI CA cannot be managed by any external CA, and the Device PKI CA cannot be a sub-CA or intermediate CA to any other CA.

    In subCA mode, the Device PKI CA uses a CA certificate that is issued by an external CA. However, the Device PKI CA does not interact directly with the external CA, and revocation of the CA certificate by the external CA does not result in the automated replacement of any certificates or keys. If a user having `ROLE_ADMIN` in scope `ALL` replaces the revoked CA certificate with a new one, the controller generates a new server certificate for the PKI broker, device ID certificates signed by the old server certificate are revoked, and PnP-managed devices retrieve a new CRL.

    Regardless of mode, neither the private CA nor the controller itself can interact directly with any external CA. PnP-managed devices never interact with any CA other than the private CA. Devices cannot be "hybrid provisioned" to interact with both the Device PKI CA and another CA.

  - **Controller PKI Plane: HTTPS connections to the controller secured by controller's certificate**

    The controller presents its server certificate in response to HTTPS connection requests. This certificate can be self-signed (default) or CA-issued (recommended), but the controller itself does not interact with any external CA.

Note Connections FROM the controller to devices do NOT take place within the controller PKI plane, even if they use SSH or SNMPv3.

- **PnP-managed devices vs. non-PnP devices**

  Devices managed by Cisco Network Plug N Play (**PnP-managed devices**) interact with the Device PKI CA. They do not interact with any other CA, even when the Device PKI CA operates in subCA mode. Therefore, they cannot learn of the revocation status of the Device PKI CA's CA certificate from the external CA.

  Devices not managed by PnP can interact with an external CA to learn of revocation of the controller's server certificate or public key, but the controller itself does not interact with an external CA, nor does it provide any sort of automated management of its server certificate. For example, upon impending expiry of its server certificate that was issued by a valid external CA, the controller does not automatically initiate renewal with the external CA. The controller admin must take explicit action to upload a renewed certificate to the controller. In contrast, as the certificate on a PnP-managed device approaches expiry, the device can renew its certificate automatically with the Device PKI CA before the certificate actually expires.

  The Device PKI CA does not manage the controller's server certificate or anything else in the controller PKI plane. Therefore, PnP-managed devices cannot learn of revocation of the controller's server certificate or public key from the private CA, even when the private CA operates in subCA mode.

  Devices cannot be "hybrid provisioned" to interact with both the Device PKI CA and any other CA. Effectively, there are two classes of devices: those managed by PnP, and those not managed by PnP. (IWAN-claimed devices are PnP-managed devices.)

- **Device-initiated connections vs. controller-initiated connections**

  When a device initiates connection to the controller, it uses HTTP(S) or SNMPv2. Only HTTPS is secure; in response to an HTTPS request, the controller presents its certificate.

  - **If the device is PnP-managed**, it cannot interact with an external CA to learn of revocation of the controller's server certificate. Therefore, unless the certificate has expired (which the device can determine without external assistance), the device accepts the certificate and the connection succeeds. This is because the "CRL distribution point" (CDP) for a PnP-enabled device is the controller's internal private CA itself, which has no knowledge of the controller's server certificate or any other CA.
  - **If the device is not PnP-managed and the device is configured to contact an external CA for certificate revocation status**, the device may reject the controller's server certificate based on information it gets from the CA. For example, if the CA says that the certificate is revoked, or if the certificate is not issued by the particular CA that the device queries about this certificate, the device rejects the connection. If the controller presents a self-signed certificate, the external CA will not recognize the certificate as valid and the device will refuse to connect. However, it is also possible for devices to use CA-signed certificates without performing revocation cross-check; for important details, see Revocation of the Controller's Server Certificate, on page 7.

  When the controller initiates connection to the device, it can use SSH, SNMP or TELNET. Only SSH and SNMPv3 (if configured appropriately) are secure, but controller-initiated connections to devices do NOT take place within the controller PKI plane.

  - **For SSH connections**, the controller presents a username/password pair to the device and the device returns its public host RSA or ECDSA key (not its device ID certificate) to the controller. The shared secret (username/password pair) validates the identity of the controller. For such connections, it is unimportant whether the controller uses a self-signed or CA-issued certificate, because the controller never presents its certificate to the device.
  - **For SNMPv3 connections**, the controller presents a username/password pair to the device. The device opens a connection according to its SNMPv3 configuration. The SNMPv3 protocol provides separate parameters that control the authentication and encryption behavior of the connection. The minimum secure configuration requires authentication; if authentication

is enabled, the option to encrypt the connection is also available. For more information, see "Configuring SNMPv3" in the *Cisco Network Visibility Application for APIC-EM User Guide*.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands