IoT Hub Device Provisioning Service security concepts

IoT Hub Device Provisioning Service is a helper service for IoT Hub that you use to configure zero-touch device provisioning to a specified IoT hub. With the Device Provisioning Service, you can auto-provision millions of devices in a secure and scalable manner. This article gives an overview of the *security* concepts involved in device provisioning. This article is relevant to all personas involved in getting a device ready for deployment.

Attestation mechanism

The attestation mechanism is the method used for confirming a device's identity. The attestation mechanism is also relevant to the enrollment list, which tells the provisioning service which method of attestation to use with a given device.

> [!NOTE] IoT Hub uses "authentication scheme" for a similar concept in that service.

Device Provisioning Service supports the following forms of attestation:

- **X.509 certificates** based on the standard X.509 certificate authentication flow.
- **Trusted Platform Module (TPM)** based on a nonce challenge, using the TPM standard for keys to present a signed Shared Access Signature (SAS) token. This form of attestation does not require a physical TPM on the device, but the service expects to attest using the endorsement key per the TPM spec.
- **Symmetric Key** based on shared access signature (SAS) Security tokens, which include a hashed signature and an embedded expiration. For more information, see Symmetric key attestation.

Hardware security module

The hardware security module, or HSM, is used for secure, hardware-based storage of device secrets, and is the most secure form of secret storage. Both X.509 certificates and SAS tokens can be stored in the HSM. HSMs can be used with both attestation mechanisms the provisioning supports.

> [!TIP] We strongly recommend using an HSM with devices to securely store secrets on your devices.

Device secrets may also be stored in software (memory), but it is a less secure form of storage than an HSM.

Trusted Platform Module

TPM can refer to a standard for securely storing keys used to authenticate the platform, or it can refer to the I/O interface used to interact with the modules implementing the standard. TPMs can exist as discrete hardware, integrated hardware, firmware-based, or software-based. Learn more about TPMs and TPM attestation. Device Provisioning Service only supports TPM 2.0.

TPM attestation is based on a nonce challenge, which uses the endorsement and storage root keys to present a signed Shared Access Signature (SAS) token.

Endorsement key

The endorsement key is an asymmetric key contained inside the TPM, which was internally generated or injected at manufacturing time and is unique for every TPM. The endorsement key cannot be changed or removed. The private portion of the endorsement key is never released outside of the TPM, while the public portion of the endorsement key is used to recognize a genuine TPM. Learn more about the endorsement key.

Storage root key

The storage root key is stored in the TPM and is used to protect TPM keys created by applications, so that these keys cannot be used without the TPM. The storage root key is generated when you take ownership of the TPM; when you clear the TPM so a new user can take ownership, a new storage root key is generated. Learn more about the storage root key.

X.509 certificates

Using X.509 certificates as an attestation mechanism is an excellent way to scale production and simplify device provisioning. X.509 certificates are typically arranged in a certificate chain of trust in which each certificate in the chain is signed by the private key of the next higher certificate, and so on, terminating in a self-signed root certificate. This arrangement establishes a delegated chain of trust from the root certificate generated by a trusted root certificate authority (CA) down through each intermediate CA to the end-entity "leaf" certificate installed on a device. To learn more, see Device Authentication using X.509 CA Certificates.

Often the certificate chain represents some logical or physical hierarchy associated with devices. For example, a manufacturer may:

- issue a self-signed root CA certificate
- use the root certificate to generate a unique intermediate CA certificate for each factory
- use each factory's certificate to generate a unique intermediate CA certificate for each production line in the plant
- and finally use the production line certificate, to generate a unique device (end-entity) certificate for each device manufactured on the line.

To learn more, see Conceptual understanding of X.509 CA certificates in the IoT industry.

Root certificate

A root certificate is a self-signed X.509 certificate representing a certificate authority (CA). It is the terminus, or trust anchor, of the certificate chain. Root certificates can be self-issued by an organization or purchased from a root certificate authority. To learn more, see Get X.509 CA certificates. The root certificate can also be referred to as a root CA certificate.

Intermediate certificate

An intermediate certificate is an X.509 certificate, which has been signed by the root certificate (or by another intermediate certificate with the root certificate in its chain). The last intermediate certificate in a chain is used to sign the leaf certificate. An intermediate certificate can also be referred to as an intermediate CA certificate.

End-entity "leaf" certificate

The leaf certificate, or end-entity certificate, identifies the certificate holder. It has the root certificate in its certificate chain as well as zero or more intermediate certificates. The leaf certificate is not used to sign any other certificates. It uniquely identifies the device to the provisioning service and is sometimes referred to as the device certificate. During authentication, the device uses the private key associated with this certificate to respond to a proof of possession challenge from the service.

Leaf certificates used with an Individual enrollment entry have a requirement that the **Subject Name** must be set to the registration ID of the Individual Enrollment entry. Leaf certificates used with an Enrollment group entry should have the **Subject Name** set to the desired device ID which will be shown in the **Registration Records** for the authenticated device in the enrollment group.

To learn more, see Authenticating devices signed with X.509 CA certificates.

Controlling device access to the provisioning service with X.509 certificates

The provisioning service exposes two types of enrollment entry that you can use to control access for devices that use the X.509 attestation mechanism:

- Individual enrollment entries are configured with the device certificate associated with a specific device. These entries control enrollments for specific devices.
- Enrollment group entries are associated with a specific intermediate or root CA certificate. These entries control enrollments for all devices that have that intermediate or root certificate in their certificate chain.

When a device connects to the provisioning service, the service prioritizes more specific enrollment entries over less specific enrollment entries. That is, if an individual enrollment for the device exists, the provisioning service applies that entry. If there is no individual enrollment for the device and an enrollment group for the first intermediate certificate in the device's certificate chain exists, the service applies that entry, and so on, up the chain to the root. The service applies the first applicable entry that it finds, such that:

- If the first enrollment entry found is enabled, the service provisions the device.
- If the first enrollment entry found is disabled, the service does not provision the device.
- If no enrollment entry is found for any of the certificates in the device's certificate chain, the service does not provision the device.

This mechanism and the hierarchical structure of certificate chains provides powerful flexibility in how you can control access for individual devices as well as for groups of devices. For example, imagine five devices with the following certificate chains:

- *Device 1*: root certificate -> certificate A -> device 1 certificate
- *Device 2*: root certificate -> certificate A -> device 2 certificate
- *Device 3*: root certificate -> certificate A -> device 3 certificate
- *Device 4*: root certificate -> certificate B -> device 4 certificate
- *Device 5*: root certificate -> certificate B -> device 5 certificate

Initially, you can create a single enabled group enrollment entry for the root certificate to enable access for all five devices. If certificate B later becomes compromised, you can create a disabled enrollment group entry for certificate B to prevent *Device 4* and *Device 5* from enrolling. If still later *Device 3* becomes compromised, you can create a disabled individual enrollment entry for its certificate. This revokes access for *Device 3*, but still allows *Device 1* and *Device 2* to enroll.