# Overview of Azure certificates

100 XP

5 minutes

As mentioned previously, Transport Layer Security (TLS) is the basis for encryption of website data in transit. TLS uses *certificates* to encrypt and decrypt data. However, these certificates have a lifecycle that requires administrator management. A common security problem with websites is having expired TLS certificates that open security vulnerabilities.

Certificates used in Azure are **x.509 v3** and can be signed by a trusted certificate authority, or they can be self-signed. A self-signed certificate is signed by its own creator; therefore, it is not trusted by default. Most browsers can ignore this problem. However, you should only use self-signed certificates when developing and testing your cloud services. These certificates can contain a private or a public key and have a thumbprint that provides a means to identify a certificate in an unambiguous way. This thumbprint is used in the Azure configuration file to identify which certificate a cloud service should use.

## Types of certificates

Certificates are used in Azure for two primary purposes and are given a specific designation based on their intended use.

1. **Service certificates** are used for cloud services
2. **Management certificates** are used for authenticating with the management API

### Service certificates

Service certificates are attached to cloud services and enable secure communication to and from the service. For example, if you deploy a web site, you would want to supply a certificate that can authenticate an exposed HTTPS endpoint. Service certificates, which are defined in your service definition, are automatically deployed to the VM that is running an instance of your role.

You can upload service certificates to Azure either using the Azure portal or by using the classic deployment model. Service certificates are associated with a specific cloud service. They are assigned to a deployment in the service definition file.

You can manage service certificates separately from your services, and you can have different people managing them. For example, a developer could upload a service package that refers to a certificate that an IT manager has previously uploaded to Azure. An IT manager can manage and renew that certificate (changing the configuration of the service) without needing to upload a new service package. Updating without a new service package is possible because the logical name, store name, and location of the certificate is in the service definition file, while the certificate thumbprint is specified in the service configuration file. To update the certificate, it's only necessary to upload a new certificate and change the thumbprint value in the service configuration file.

### Management certificates

Management certificates allow you to authenticate with the classic deployment model. Many programs and tools (such as Visual Studio or the Azure SDK) use these certificates to automate configuration and deployment of various Azure services. However, these types of certificates are not related to cloud services.

## Using Azure Key Vault with certificates

You can store your certificates in Azure Key Vault - much like any other secret. However, Key Vault provides additional features above and beyond the typical certificate management.

- You can create certificates in Key Vault, or import existing certificates
- You can securely store and manage certificates without interaction with private key material.
- You can create a policy that directs Key Vault to manage the life cycle of a certificate.
- You can provide contact information for notification about life-cycle events of expiration and renewal of certificate.
- You can automatically renew certificates with selected issuers - Key Vault partner x509 certificate providers / certificate authorities.

Automating certificate management helps to reduce or eliminate the error prone task of manual certificate management.

---

**Next unit: Protect your network**

Continue >