Windows 10

Windows 10 IoT Core Azure Connectivity and Security

Published July 27, 2016
Version 1.0

Table of Contents

Introduction

The Internet of Things is gaining more momentum with millions of devices that connect to each other and the cloud. This enables powerful scenarios for users and enterprises but also presents challenges in securing devices and cloud services. Compromised devices can cause devastating consequences. Secure IoT devices are essential for the overall success of the Internet of Things. Thus, Microsoft has designed Windows 10 IoT with security in mind.

One key area of IoT security is the potential vulnerability in the connection between a device and its cloud service. Currently, IoT device developers often must place security measures, such as access keys, directly into their code in order for their devices to use cloud services, such as Azure. Developer access to this information increases the risk of accidental exposure by the developer and also opens vulnerabilities for malicious attacks. It is essential that security measures on devices remain confidential and devices have strong, trusted identities that enable their connection to cloud services. Providing these capabilities as part of the Windows 10 IoT platform helps developers build devices that can operate securely and with trust.

This document discusses security as a platform capability and offers guidance to developers creating strong devices and solutions for a trustworthy Internet of Things.

Device identities

Connecting devices and services raises an important question: *Can I trust that the information comes from the right source and can only be read by the intended recipient?* It is essential that the identity of the receiving device or service is trusted even if the data sent is not confidential. For example, if a process decision depends on data from a temperature sensor, then security measures must ensure the cloud service only accepts information from the trusted sensor and that the sensor has not been compromised. In some cases, malicious attackers may spoof the identity of devices like the temperature sensor. Thus, security measures must work to prevent the cloud service from receiving data from a stolen identity.

While in the "Internet of Humans" identity authentication this is more or less solved through a wide range of technologies and human authentication (e.g. secret questions, picture recognition), these techniques are not practical for the Internet of Things due to lack of human involvement in IoT communications and the resource constrains of IoT devices.

Device manufacturers, system integrators, and service providers must offer a platform that provides a trusted interaction between devices and services. This will drive user and enterprise confidence in the secure future of IoT.

Building security into the platform

Generally, today's device development practices call for the device code to handle keys. In some cases, the keys may be directly accessible by the code, which creates a security vulnerability.

With the new Windows 10 Anniversary Update, Microsoft will provide key storage in a secure location not accessible by code running on the device and will provide secured operations as part of the Windows 10 platform. Therefore, the code no longer needs to handle keys.

Top imperatives for secure key storage:

- Keys need to be stored securely on the device and need to resist duplication. Physical access to the device must not permit access to secret keys.
- The cryptographic process must be executed in a trusted environment.
- The data process/transmission code must not be able to duplicate secret keys.

Today, Microsoft has created a better method to address these imperatives – Now the software architecture to connect and communicate with Azure services is designed to separate the cryptographic processing and key storage from the data processing and submission. This new architecture allows developers to build devices that use Azure web services without the need for knowledge of, or access to, cryptographic processing and secret information. They can trust that the platform will handle this information.

Security as a platform feature

In the Windows 10 Anniversary Update, Microsoft uses Trusted Platform Module (TPM) technology to handle security relevant tasks in Windows 10 IoT Core. TPM is driven by the Trusted Computing Group (TCG) and is an industry-wide accepted technology for cryptographic processing and secured storage of secrets.

There are three types of TPMs:

1. Discrete TPM (dTPM), which is a silicon module separate from other system elements,

2. TPMs implemented in firmware (fTPM) in a secured operating environment on chip, and

3. Software TPMs (sTPM), which exist only for development purposes and are not considered secure.

The TCG's latest release is TPM 2.0. It evolved from TPM 1.2 with a redesigned architecture, enhanced cryptography and solution authorization algorithms, simplified TPM management, and more. TPM 1.2 is widely used in PCs today. Today, Windows 10 IoT requires TPM 2.0. Windows 10 IoT Enterprise offers backward compatibility with TPM 1.2; however, Windows 10 IoT Core does not offer backward compatibility. More information on how to use TPMs with Windows IoT Core can be found on the TPM on Windows IoT Core website. TPM hardware options will be discussed later in this document.

TPMs provide a wide range of cryptographic functionality. On Windows 10 IoT Core the TPM is used for storing the Azure secret keys and processing requests to create tokens to access Azure assets.

Using TPMs gives OEMs the opportunity to create identities for their devices in Azure and attach services – even if the OEM has not determined the final software that will be executed on its devices. Using TPMs also moves the secret key storage, and therefore the Azure identities, of devices out of storage volumes and into platform-independent hardware storage. This creates platform-independent identities for the devices. Platform-independent Azure identities can withstand a wipe, a reimaging of the OS, or a change in the platform. Additionally, software updates on the devices will not impact the identities of these devices. Moving the key store in this way allows a separation of the provisioning process from the software development process, as follows:

> During the **provisioning process**, the keys are transferred from the Azure account to the device. This can be done in a controlled environment and limits the number of users that have direct access to those secrets.
>
> During the **software deployment process**, the client code can be deployed and executed without any developer knowledge of keys needed to access Azure resources. Instead, the code will use the keys provisioned on the device, but will not have access, or the need to access, those keys in order to communicate with Azure resources.

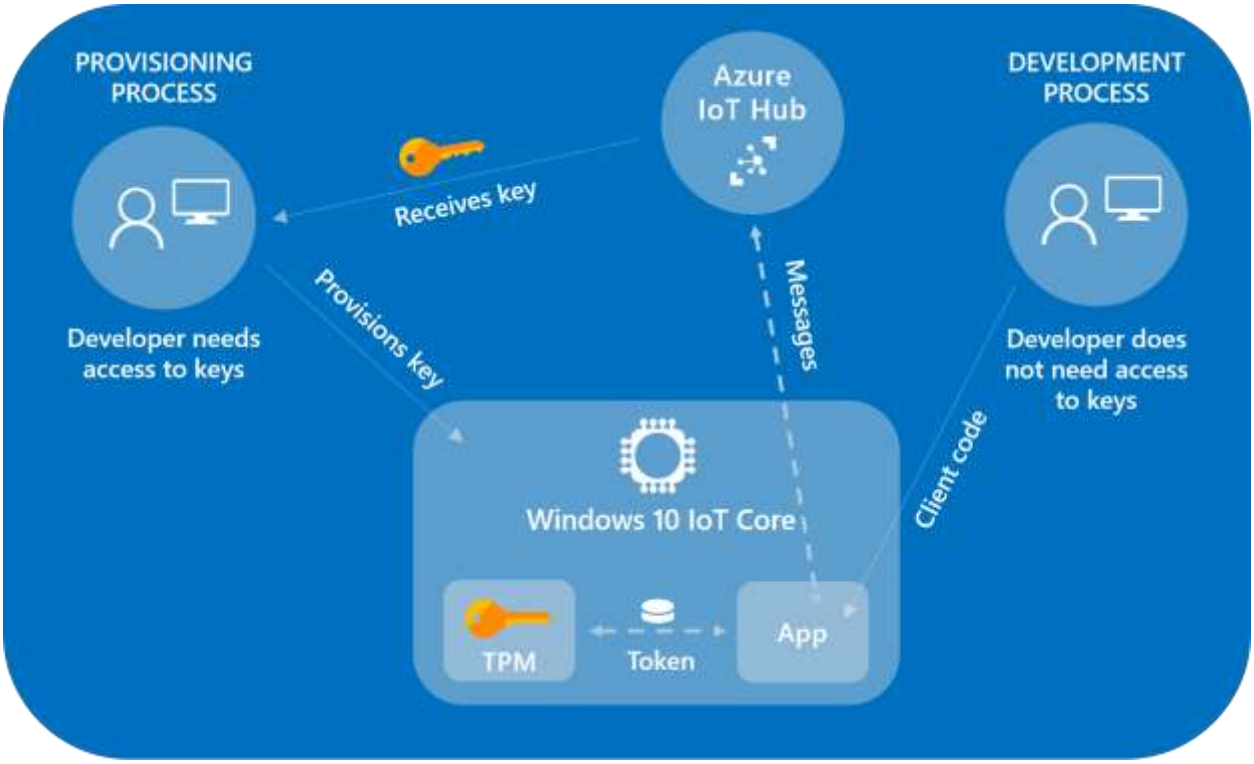

Figure 1. The provisioning process and the software deployment process

Provisioning process

Provisioning is an important process in the lifecycle of a device, in which the device receives an Azure identity, which is essentially comprised of (1) the information contained in the connection string, such as the URL; (2) a device identity; and (3) a secret key. During provisioning, secrets are exchanged between the Azure cloud and the device. There is a risk that the key information could become compromised during this process. Currently, Azure IoT Hub uses symmetric keys that could become an ongoing security concern if compromised during provisioning. There are several ways to mitigate this risk:

A. Symmetric key exchange. A symmetric key exchange is conducted in a secure environment that is closely monitored. The transport of the key from the cloud to the device, or vice versa depending who issues the key, must be secured from end-to-end. Windows 10 IoT does not provided tools for a symmetric key exchange; however, developers can create their own environments and tools to move symmetric keys between the TPM and the cloud.

B. Asymmetric key exchange. Using asymmetric keys will mitigate the risk of compromising keys during exchange because only the public portion of the key leaves the device to be exchanged in the cloud. Asymmetric keys enable a secure key exchange between two instances (i.e., a device and a cloud service), but there is still the risk of false identity if the key pair can be accessed and moved to another device. Thus, an additional security measure is needed to assure the key pair cannot leave the device. Option C below discusses using TPMs as this additional security measure.

C. TPM-generated asymmetric key exchange. A developer can use a TPM to generate the asymmetric key pair, as shown in Figure 2. The device-side key cannot be moved to another device and the Azure service interacts with that key as it would interact with an asymmetric key generated in any method. Each TPM has an endorsement certificate that can be verified by the Azure service. The endorsement certificate enables the Azure service to validate that the public key and the matching private key where created by this TPM. This allows the service to verify the history of the key pair and the location of the private key.

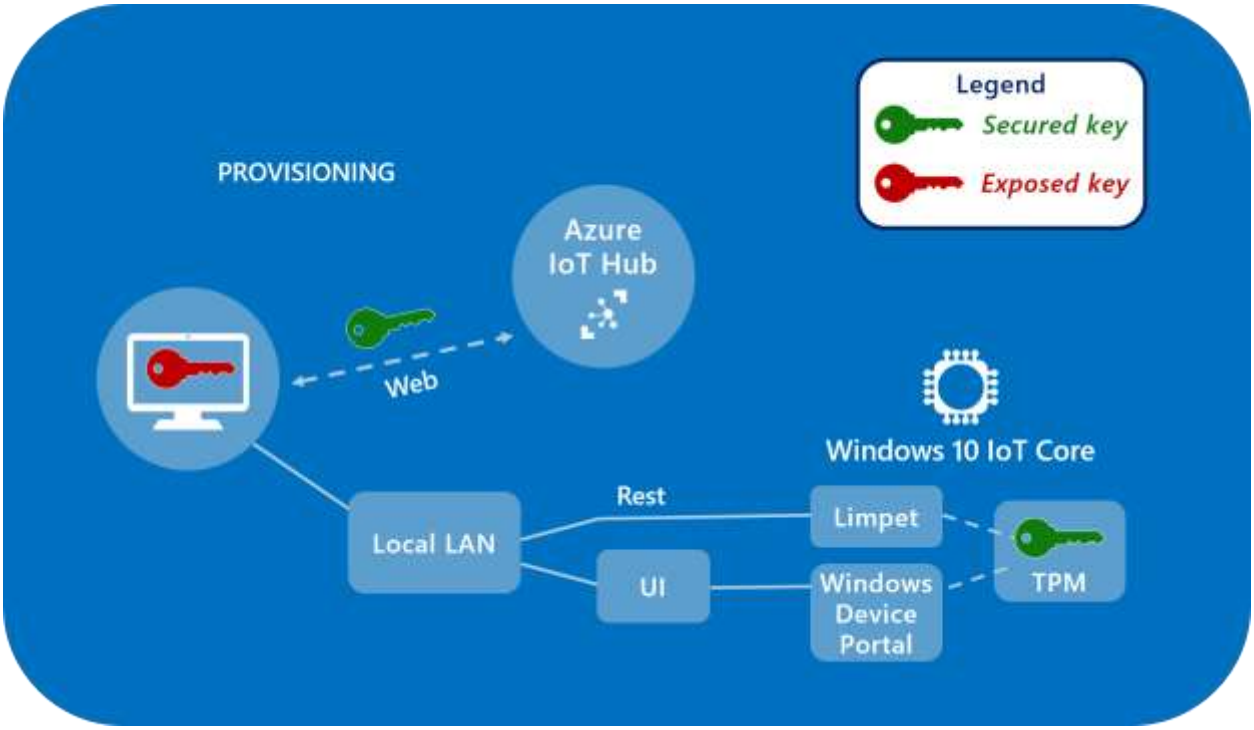

Figure 2. TPM-generated asymmetric key exchange

Software Architecture - TPM Stack and Tools

Microsoft provides a TPM stack for Windows 10 IoT Core. The stack consists of a TPM driver and an API adaption layer. There are several tools available to use this this stack and help developers build applications and provision TPMs with keys, as further described below.

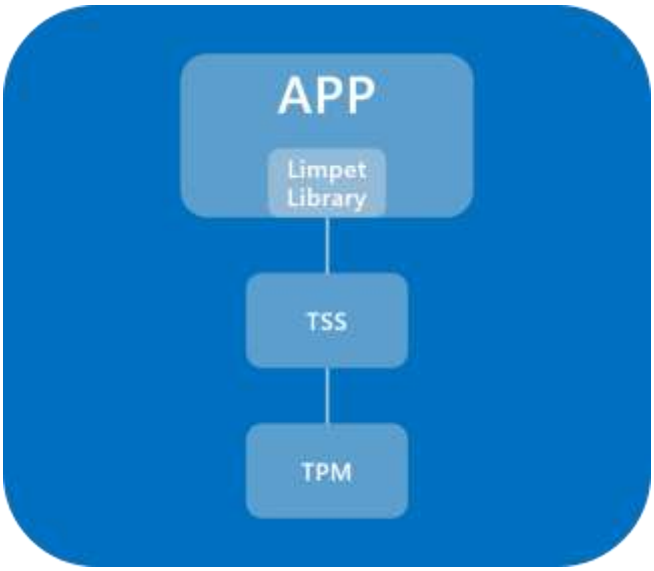

Figure 3. Architecture overview

TPM Software Stack

Microsoft offers a TPM Software Stack (TSS). The software stack can be configured to work with any TPM 2.0 through ACPI tables. Microsoft provides several tables for TPMs that are currently on the market.

Any TCG compatible SPI or I2C TPM is supported in the Windows 10 IoT Core Anniversary Update. Currently, there are ACPI tables available for TPM from the following manufacturers: Infineon, NationZ, Nuvoton, and STMicro. Any new TPMs can also be supported through this mechanism. The latest information on ACPI tables can be accessed here.

The stack gives Windows applications access to the TSS 2.0 APIs. The source of the stack is available on GitHub.

Limpet

Limpet is a collection of functions to simplify the access to TPM functionality. It helps developers to better utilize cryptographic functionality in their application. The development tools and samples released by Microsoft use this library. The code is available on GitHub.

TPM simulator

On several of the developer platforms a discrete or firmware TPM might not be available yet; however, developers would like to begin prototyping. In these circumstances, Microsoft empowers developers by providing a TPM simulator for the purpose of development and prototyping. The simulator provides many of the same cryptographic processing capabilities as a real TPM, but without security features, such as secure storage and tamper protection. Therefore, the TPM simulator is only appropriate in a prototyping environment. It is not secure and must not be used in a production environment.

The TPM simulator enables developers to build and validate their code independent from TPM hardware. The tool gives OEMs the flexibility to test their applications and later choose what TPMs to use in the production versions of their devices.

Windows 10 IoT Core Dashboard

The Windows 10 IoT Core Dashboard is a tool for developers and Makers to setup devices and for prototyping and development based on the Windows 10 IoT Core reference platforms, such as Raspberry Pi, Dragon Board, or Minnow Board Max. Microsoft extended the tool to help developers connect their devices to Azure. The Windows 10 IoT Dashboard provides the following functionality:

- Apply settings for TPM hardware configurations (ACL tables)
- Install the TPM simulator if no discrete or firmware TPM is available
- Validate that the TPM has the correct capabilities
- Connect to the user Azure account
- Create Azure IoT Hub and Azure device if needed
- Provision the device with the user's Azure keys
- Remove Azure keys

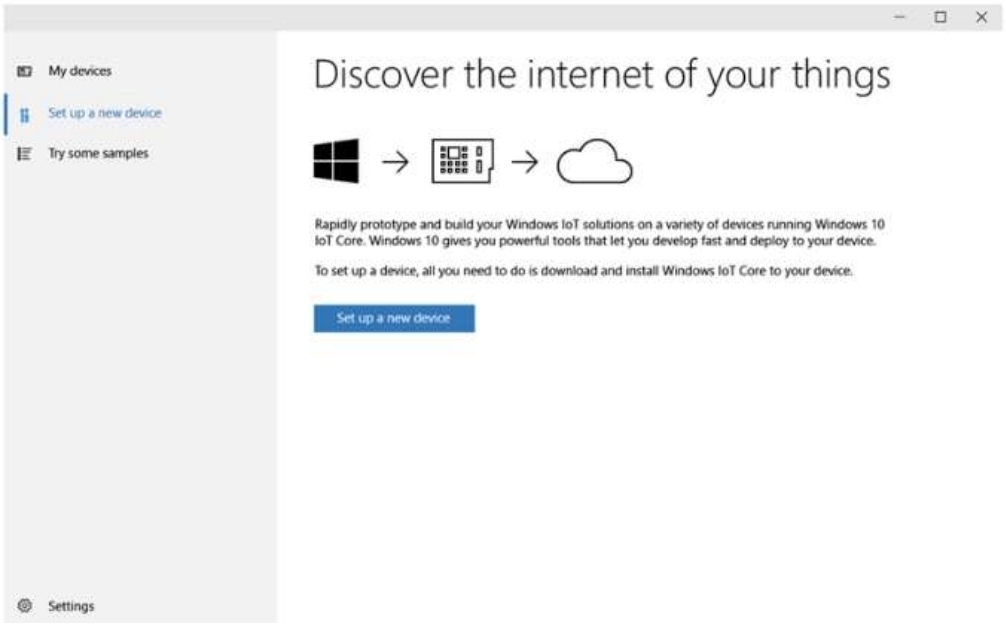

Figure 4. Windows 10 IoT Core Dashboard

Windows Device Portal

The Windows Device Portal (WDP) is a web configuration tool available on the Windows 10 IoT Core reference platforms. Similar to the Windows 10 IoT Core Dashboard described above, the WDP was extended to allow provisioning of Azure keys. The WDP gives developers more flexibility to store keys on the TPM. The picture below shows the TPM configuration screen.

Figure 5. TPM configuration screen in the Windows Device Portal

Visual Studio Connected Service for Azure IoT Hub

The first version of the Visual Studio Connected Service for Azure IoT Hub was released in early 2016. It is a Visual Studio extension that helps developers to quickly build Azure IoT Hub connected applications; however, it was still the developer's responsibility to handle and store the secret keys to authenticate to Azure IoT Hub. The newly updated version will alleviate this responsibility by leveraging the updated APIs and software stack to create code that use the TPM infrastructure.

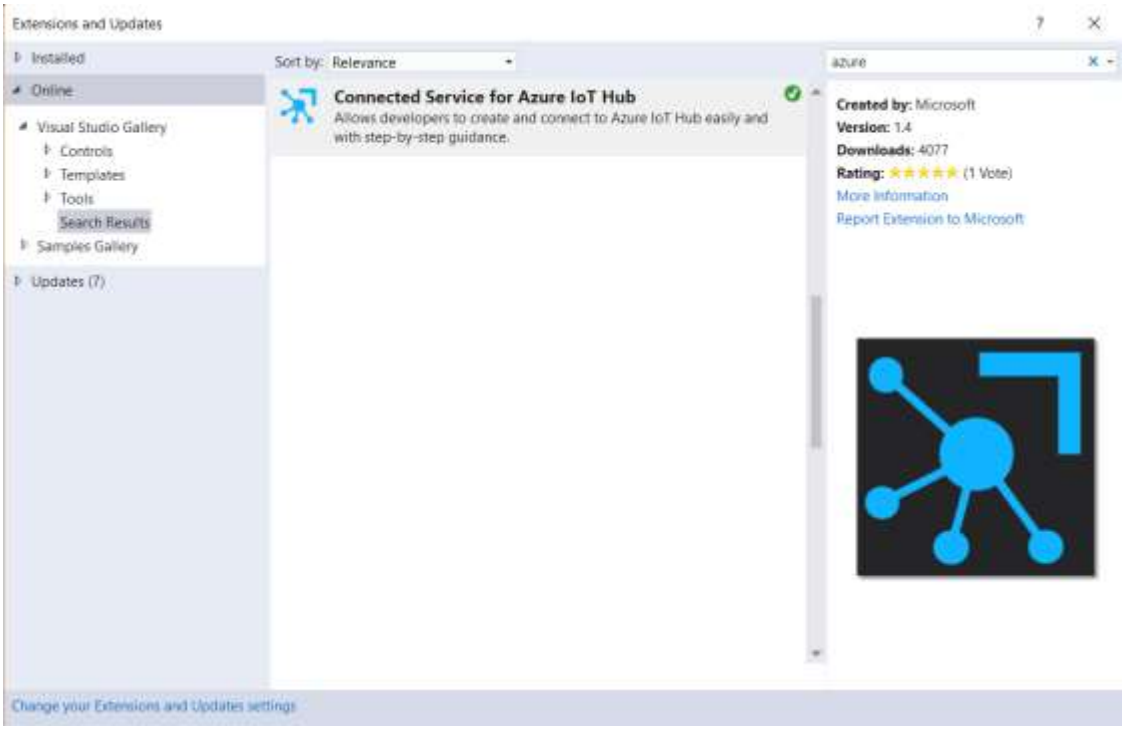

Figure 6. Connected Service for Azure IoT Hub

Conclusion

Microsoft empowers device developers and Makers to build the next generation of secured connected devices for the future of the Internet of Things. IoT device manufacturers must offer their customers a platform that provides a trusted interaction between devices and services. Thus, IoT devices should be built with security in mind during all phases of prototyping and development. With Windows 10 IoT Core, developers can create connected devices with security as a platform capability. Microsoft offers the technology and developer tools to design devices that harness the latest TPM technology to provide safer Azure connectivity that is essential for the Internet of Things.