Using the Windows 8 Platform Crypto Provider and Associated TPM Functionality

Stefan Thom
Senior Software Development Engineer and Security Architect for IEB, Microsoft

Rob Spiger, Senior Security Strategist
Agenda

- Windows 8 TPM Scenarios
- Hardware Choices with Windows
- Windows 8 TPM Provisioning
- Platform Crypto Provider
- TPM Base Services
- Boot Measurements and Attestation
- PCP ToolKit
Attestation – Measurements are recorded during the boot process and proven to a 3rd party

BitLocker Drive Encryption™ - The OS volume is encrypted and booting it requires TPM boot measurements match and an optional PIN is provided during boot

Platform Crypto Provider
- Certificates can be created with their private key bound to the TPM
- 3rd party applications can use the TPM like any other Key Storage Provider in Windows 8

Virtual Smart Card
- The TPM acts like a permanently inserted Smart Card.

3rd Party Applications
- Applications send TPM commands to the device through the TPM Base Services API in Windows
Windows 8 support for applications using TPM

- Direct TPM Commands
  - Win32_TPM Class
  - Key Storage Provider
  - TPM Provisioning
- Internal TPM 1.2/TPM 2.0 Abstraction Layer
- TPM Base Services API
- Microsoft TPM Stack (resource virtualization & power management)
- TPM 1.2
- TPM 2.0

TPM Applications
- Virtual Smart Card
- Provider
- Provisioning
- Microsoft TPM Stack (resource virtualization & power management)
- Attestation And BitLocker
<table>
<thead>
<tr>
<th>No Secure Boot</th>
<th>Secure Boot on by default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows update opt-in</td>
<td>Windows update opt-in</td>
</tr>
<tr>
<td>Runs Windows 7 apps</td>
<td>Runs Windows 7 apps</td>
</tr>
<tr>
<td>TPM 1.2 Optional</td>
<td>TPM Optional</td>
</tr>
<tr>
<td>Windows 7, maybe others</td>
<td>Windows 8, Windows 8 Pro, maybe others</td>
</tr>
<tr>
<td>Windows Defender and Firewall on by default</td>
<td>Windows Defender and Firewall on by default</td>
</tr>
</tbody>
</table>

**Windows 8 Certified**

- Secure Boot on by default
- Windows update opt-in
- Runs Windows 7 apps
- Battery saving design
- TPM 2.0 mandatory
- Windows 8 or Windows 8 Pro, maybe others
- Windows Defender and Firewall on by default
- Secure Boot always on
- Windows update always on
- Only runs additional apps from the Microsoft store
- Battery saving design
- TPM 2.0 mandatory
- Pre-installed Windows RT only
- Only works with hardware certified for Windows RT
- Windows Defender and Firewall always on
Getting the TPM Provisioned

- Windows (automatic) if things are configured correctly

- Administrators: Use tpm.msc and “Prepare the TPM” if the status is not “Ready for use”

- Launch tpminit.exe with administrator privileges from command line or an application
  - Interactively does the process of provisioning for applications using the Windows 8 UX
  - Might involve a reboot (or two) and actions during the boot process

- Use the Win32_TPM Class Provisioning Method
  - Provision(Boolean ForceClear_Allowed, Boolean PhysicalPresencePrompts_Allowed)
  - Your application controls the user experience and needs to do the interaction with users
## Windows 8 Automatic TPM 1.2 Provisioning

<table>
<thead>
<tr>
<th>Windows State</th>
<th>Out of Box</th>
<th>After Reboot</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning</td>
<td>Not Ready</td>
<td>Not Ready</td>
<td>Ready</td>
</tr>
<tr>
<td>Flag = True</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Firmware State  |                   |                    |              |
| Provisioning    | Provisioning Flag = True | Provisioning Flag = True | Provisioning Flag = True |
| Flag = True     |                   |                    |              |

<p>| TPM State       |                   |                    |              |
| Disabled        |                   | Enabled            | Enabled      |
| Deactivated     |                   | Activated          | Activated    |
| Ownership       | Ownership Not Taken | Ownership Not Taken | Ownership Taken |</p>
<table>
<thead>
<tr>
<th>Windows TPM State</th>
<th>Out of Box</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Ready</td>
<td></td>
<td>Ready</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Firmware State</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TPM Visible to OS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Owner Authorization</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EmptyAuth</td>
<td></td>
<td>Set</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Endorsement Authorization</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EmptyAuth</td>
<td></td>
<td>Set</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lockout Authorization</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EmptyAuth</td>
<td></td>
<td>Set</td>
</tr>
</tbody>
</table>
Platform Crypto Provider (PCP)

- Requires the TPM to be in a Ready state
- Works with TPM 1.2 or TPM 2.0 and runs in-process
- Provides the standard Crypto Next Generation interface using the TPM
- May specify the Platform Crypto Provider in a certificate template

Tracing
- Developers can trace PCP TPM commands by setting REG_SZ value “ProviderTraces” under the key HKLM\SYSTEM\CurrentControlSet\Services\TPM with a folder name
- The process needs write permissions for the directory
Exposes the TPM random number generator through the CNG RNG interface

Works with TPM 1.2 or TPM 2.0 and runs in-process

Only uses the TPM as a source (unlike the default Windows provider that uses many sources)

Depending how much entropy the TPM can generate in a single command it could take many TPM commands to generate entropy. The limit permitted is 4096 bytes per a call.

You can stir the TPMs entropy with up to 256 bytes too.
Supports a subset of the Microsoft software provider

Fully supports all actions used by certificate enrollment

Does not support exportable keys without any authorization

Does support exportable keys that require a migration authorization for controlled export

Supports a bunch of additional methods and properties allowing use of the TPM’s functionality
  - Keys bound to PCR values
  - Having the TPM constrain the key usage for a specific purpose like encryption or signing
  - Creating attestation information about a key to show it exists in a TPM
  - May be used by custom enrollment agents
Permits applications to send TPM 1.2 or TPM 2.0 commands directly

Resource virtualization

Available in kernel mode or user mode

Methods:
- Tbsi_Context_Create
- Tbsip_Context_Close
- Tbsi_Get_TCG_Log
- Tbsi_Revoke_Attestation
- Tbsi_GetDeviceInfo
- Tbsip_Submit_Command
Trusted boot architecture

1. UEFI Secure Boot prevents running an unknown OS loader
2. ELAM starts first and enforces its policy
3. Boot measurements were recorded during boot
4. Signed TPM boot measurements can be sent to an off-box service for analysis
Boot Process up to Windows Boot Manager

Unified Extensible Firmware Interface (UEFI) Boot

- Power On
- TPM Startup
- System Specific Firmware
- UEFI Environment
- UEFI Drivers
- UEFI Windows Boot Manager
- Compatibility Support Module (CSM)

Conventional BIOS Boot

- Conventional BIOS Environment
- Option ROMs
- Master Boot Record
- Boot Block
- Boot Sector
- Windows Boot Manager
Boot Process from Boot Manager

**Boot from Hibernation**

1. Measured in TPM by Win8
   - WinResume

2. Hibernation File

3. Windows 8 Operating System Running

**Hibernating**

1. Measured in TPM by Windows 8

2. Hibernation File

3. Windows Kernel
   - Core Windows Kernel
   - Early Launch Anti-Malware Drivers
   - Early Launch Driver Policy

4. Boot Drivers
   - Anti-malware Runtime Driver
   - Storage Driver
   - Other

5. Early Launch Anti-Malware Drivers Stopped

6. Normal Drivers
An Attestation identity key is provisioned for the platform.

Provisioning process verifies the attestation key only exists in the TPM.

Attestation keys are used to sign the current PCR values, the result is a Quote.

Proof is a Boot Log and a Quote of PCR values that can be reviewed by a remote verifier to determine if the client system is trustworthy.
What executes in Hibernation/Resume Cycles

**Full Boot Process**

1. **TPM Startup**
2. **System Specific Firmware**
3. **UEFI BIOS**
4. **Windows Boot Manager**
   - Record TPM Restart Counter
5. **WinLoad**
6. **Kernel Startup**
   - Early Launch Anti-Malware Drivers
   - Early Launch Driver Policy
   - Anti-malware Runtime Driver Handoff
7. **Windows 8 Operating System Running**
   - Anti-malware Runtime Driver Policy Enforcement

**Possible Extra Boot with “Offline” Event**

1. **TPM Startup**
2. **System Specific Firmware**
3. **UEFI BIOS**
4. **Something else booted?**
5. **Hibernation File Tampering?**
   - **Hibernation File**
   - **Archive Boot Log with Quote of the PCRs**
    - **Windows 8 Operating System Running**
     - Anti-malware Runtime Driver Policy Enforcement

**Another Boot Resuming from Hibernation**

1. **TPM Startup**
2. **System Specific Firmware**
3. **UEFI BIOS**
4. **Windows Boot Manager**
   - Record TPM Restart Counter
5. **WinResume**
6. **Restored Hibernation File**
7. **Windows 8 Operating System Running**
   - Anti-malware Runtime Driver Policy Enforcement
What if anti-malware stops enforcing policy?

- Anti-malware might not be able to continuously enforce policy
- Might need to do something it doesn’t trust

**Full Boot Process**

- Can call Windows RevokeAttestation API
  - Changes the value of PCR[12] so the boot log & quote don’t match
  - Increments the Event Counter the archive log doesn’t match the value in the TPM on hibernation/resume
Attestation for Hibernation Resume Cycle

Another Boot Resuming from Hibernation

- Can be reviewed by a remote verifier to determine if the client system is trustworthy
- Continuous boot logs can be confirmed by successive TPM restart counter measurements in the archived logs
- If the Event Counter skips a value, the RevokeAttestation was called in previous boot
## Boot Process Measurements

<table>
<thead>
<tr>
<th></th>
<th>TPM Startup</th>
<th>Firmware + BootManager</th>
<th>Boot Manager Execution</th>
<th>WinLoad or WinResume Execution</th>
<th>Windows Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Firmware PCR[0-7]</strong></td>
<td></td>
<td>Power-on components through Boot Manager</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PCR[8-10]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Details PCR[12]</strong></td>
<td></td>
<td>Boot Manager, WinLoad &amp; Resume Configuration Restart Counter Event Counter</td>
<td></td>
<td>WinLoad / Resume Config Kernel Options Early Launch Antimalware Data</td>
<td></td>
</tr>
<tr>
<td><strong>OS Modules PCR[13]</strong></td>
<td></td>
<td>BootMgr.mui WinLoad WinResume</td>
<td></td>
<td>WinLoad.mui WinResume.mui Windows Kernel Boot Drivers</td>
<td></td>
</tr>
<tr>
<td><strong>OS Authorities PCR[13]</strong></td>
<td></td>
<td>Boot Mgr.mui WinLoad WinResume</td>
<td></td>
<td>Boot Mgr.mui WinLoad WinResume</td>
<td></td>
</tr>
<tr>
<td><strong>Monotonic Counter</strong></td>
<td>Restart Count++</td>
<td>Event Counter Incremented</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optional Revoke Attestation Action

Invalidated

Separator Measurement

Event Counter Incremented

Trust Boundary
Creating AIKs

TPM
1.2: CreateIdentity
1.2: AIK, IDBinding
2.0: CertifyCreation
2.0: IDBinding
1.2: ActivateIdentity
2.0: ActivateCredential
Secret

Windows
NcryptSetProperty
NcryptFinalizeKey
NcryptedProperty
IDBinding
NcryptedProperty
EKPub/EKCert
NcryptSetProperty
NcryptedProperty
Secret

Client
Nonce
NcryptedProperty
IDBinding
NcryptedProperty
EKPub/EKCert and IDBinding
NcryptedProperty
Secret

Server
TpmAttPubKeyFromIDBinding
KeyHandle
TpmAttGenerateActivation
ActivationBlob

TpmAtt
ActivationBlob

TPM 2.0 Library Specification: [https://www.trustedcomputinggroup.org/resources/tpm_library_specification](https://www.trustedcomputinggroup.org/resources/tpm_library_specification)


Q & A

- Stefan Thom (stefanth@microsoft.com)
- Rob Spiger (rspiger@microsoft.com)