Using the Windows 8 Platform Crypto Provider and Associated TPM Functionality

Stefan Thom
Senior Software Development Engineer and Security Architect for IEB, Microsoft
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
TPM Scenarios with Windows 8

- 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 Attestation – Measurements are recorded during the boot process and proven to a 3rd party.

- 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
  - Virtual Smart Card

- 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

- Attestation And BitLocker

Trustworthy Computing
## Windows Hardware Landscape

### Windows 7 Logo
- No Secure Boot
- Windows update opt-in
- Runs Windows 7 apps
- TPM 1.2 Optional
- Windows 7, maybe others

<table>
<thead>
<tr>
<th>x86/x64 CPU</th>
<th>x86/x64 CPU Connected Standby Scenario</th>
<th>ARM Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Boot on by default</td>
<td>Secure Boot on by default</td>
<td>Secure Boot always on</td>
</tr>
<tr>
<td>Windows update opt-in</td>
<td>Windows update opt-in</td>
<td>Windows update always on</td>
</tr>
<tr>
<td>Runs Windows 7 apps</td>
<td>Runs Windows 7 apps</td>
<td>Only runs additional apps from the Microsoft store</td>
</tr>
<tr>
<td>TPM Optional</td>
<td>Battery saving design</td>
<td>Battery saving design</td>
</tr>
<tr>
<td>Windows 8, Windows 8 Pro, maybe others</td>
<td>TPM 2.0 mandatory</td>
<td>TPM 2.0 mandatory</td>
</tr>
<tr>
<td>Windows Defender and Firewall on by default</td>
<td>Windows 8 or Windows 8 Pro, maybe others</td>
<td>Pre-installed Windows RT only</td>
</tr>
<tr>
<td></td>
<td>Windows Defender and Firewall on by default</td>
<td>Only works with hardware certified for Windows RT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Windows Defender and Firewall always on</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Windows State</th>
<th>Out of Box</th>
<th>After Reboot</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Ready</td>
<td></td>
<td>Not Ready</td>
<td>Ready</td>
</tr>
<tr>
<td>Firmware State</td>
<td>Provisioning Flag = True</td>
<td>Provisioning Flag = True</td>
<td>Provisioning Flag = True</td>
</tr>
<tr>
<td>Disabled</td>
<td></td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Deactivated</td>
<td></td>
<td>Activated</td>
<td>Activated</td>
</tr>
<tr>
<td>Ownership Not Taken</td>
<td></td>
<td>Ownership Not Taken</td>
<td>Ownership Taken</td>
</tr>
</tbody>
</table>
## Windows 8 Automatic TPM 2.0 Provisioning

<table>
<thead>
<tr>
<th>Windows TPM State</th>
<th>Out of Box</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Not Ready</td>
<td>Ready</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Firmware State</th>
<th>Out of Box</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TPM Visible</td>
<td>TPM Visible</td>
</tr>
<tr>
<td></td>
<td>to OS</td>
<td>to OS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Owner Authorization</th>
<th>Out of Box</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EmptyAuth</td>
<td>Set</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Endorsement Authorization</th>
<th>Out of Box</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EmptyAuth</td>
<td>Set</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lockout Authorization</th>
<th>Out of Box</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EmptyAuth</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.
NCrypt RSA Platform Key Storage Provider

- 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
Behaves from the software point of view exactly like a smart card
- Software Card with TPM based keys
- Software simulated reader

Remote desktop services: Client cards are mapped to server

Keys are strongly bound to the machine: PC becomes the card

Several cards may exist in parallel with different PINs for separation

Full support for PUK unblock and 3DES challenge response admin authorization

Hardware enforced dictionary attack protection using TPM provided solution
- Does not behave like regular Smart Cards
- Shared across all cards and other TPM features
TPM Base Services API (TBS)

- 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**

1. **Power On**
2. **TPM Startup**
3. **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**
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

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
5. WinLoad
6. WinResume
7. TPM Startup
8. System Specific Firmware
9. UEFI BIOS
10. Windows Boot Manager
11. WinLoad
12. WinResume
13. Increment TPM Restart Counter
14. Record TPM Restart Counter
15. Kernel Startup
16. Early Launch Anti-Malware Drivers
17. Early Launch Driver Policy
18. Anti-malware Runtime Driver Handoff
19. Windows 8 Operating System Running
20. Anti-malware Runtime Driver Policy Enforcement
21. Stored Hibernation File
22. Archive Boot Log with Quote of the PCRs
23. Trustworthy Computing

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

1. TPM Startup
2. System Specific Firmware
3. UEFI BIOS
4. Something else booted?
5. Hibernation File Tampering?
6. Increment TPM Restart Counter
7. Record TPM Restart Counter
8. Windows 8 Operating System Running
9. Anti-malware Runtime Driver Policy Enforcement
10. Stored Hibernation File

**Another Boot Resuming from Hibernation**

1. TPM Startup
2. System Specific Firmware
3. UEFI BIOS
4. Windows Boot Manager
5. WinResume
6. Restored Hibernation File
7. Increment TPM Restart Counter
8. Record TPM Restart Counter
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

- TPM Startup
- System Specific Firmware
- UEFI BIOS
- Windows Boot Manager
- WinResume
- Restored Hibernation File
- Stored Hibernation File
- Windows 8 Operating System Running
  - Anti-malware Runtime Driver Policy Enforcement
  - Proof is a Boot Log and a Quote of PCR values
  - Archived Boot Logs and Quotes from previous boots

- Increment TPM Restart Counter
- Record TPM Restart Counter
- Event Counter Increment & Measurement

- 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

## TPM Startup
- **Firmware PCR[0-7]**
  - Power-on components through Boot Manager

## Firmware + BootManager
- **PCR[8-10]**
  - Details
    - **PCR[12]**
      - OS Modules
        - **PCR[13]**
          - OS Authorities
            - **PCR[14]**
              - Monotonic Counter
                - Restart Count++

## Boot Manager Execution
- **Boot Manager, WinLoad & Resume Configuration**
  - Event Counter
    - Incremented

## WinLoad or WinResume Execution
- **WinLoad / Resume Config Kernel Options**
  - Early Launch Antimalware Data
  - **WinLoad.mui**
  - **WinResume.mui**
  - Windows Kernel
    - Boot Drivers
  - Boot Mgr.mui
    - WinLoad
    - WinResume

## Windows Kernel
- **Optional Revoke Attestation Action**
  - Invalidated
  - Event Counter
    - Incremented

---

**Trust Boundary**
Creating AIKs

TPM

1.2: CreateIdentity/2.0: Create
1.2: AIK, IDBinding / 2.0 AIK
2.0: CertifyCreation
2.0: IDBinding

1.2: ActivateIdentity/2.0: ActivateCredential
Secret

Windows

NcryptSetProperty
NcryptFinalizeKey

NcryptGetProperty
IDBinding
NcryptGetProperty
EKPub/EKCert

Secret

Client

Nonce

EKPub/EKCert and IDBinding

ActivationBlob

Server

TpmAtt

TpmAttPubKeyFromIdBinding

KeyHandle
TpmAttGenerateActivation

ActivationBlob
PCP Toolkit

TPM Platform Crypto Provider Toolkit:
http://research.microsoft.com/en-us/downloads/74c45746-24ad-4cb7-ba4b-0c6df2f92d5d/

- TPM integration documentation
- Supports 2.0 and 1.2 TPMs
- Sample code
- Software abstraction layer
- Handy tool to explore TPM features
TPM Platform Crypto Provider Toolkit: http://research.microsoft.com/en-us/downloads/74c45746-24ad-4cb7-ba4b-0c6df2f92d5d/

TPM 2.0 Library Specification: https://www.trustedcomputinggroup.org/resources/tpm_library_specification


Windows 8 Hardware Certification Requirements: http://msdn.microsoft.com/library/windows/hardware/hh748188
Q & A

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