TPM and Intel ® PTT Overview

TCE 4th summer school on computer security and big data

Ahuva Kroizer, Intel
September 8, 2015
Agenda

- TPM overview
  - What is TPM?
  - Chain of trust measurements

- TPM usage examples
  - Sealing data
  - Attestation
  - Virtual Smart Cards
  - Storage keys and hierarchies

- Intel® PTT
  - How TPM is implemented in Intel platforms
What is a TPM?

- **TPM = Trusted Platform Module**
  - TPM is a HW device which provides trust capabilities to the platform
- **Standard developed by TCG = Trusted Computing Group**
- **Why a TPM, why not software?**
  - Asset protected from “Host” software. E.g., OS, VMM
  - Host has no access to assets (secrets) except thru TPM 2 defined interfaces
    - No direct memory access

- **TPM Commands**
  - Binding to RTM (root of trust for measurement)
    - More than ever 1:1 relationship people:device
Usages – Just a few!

• Digitally signed transactions
• Local encryption (e.g., disk encryption, BitLocker)
  • Lower value w/o TPM
• Enables low entropy passwords (easy to remember)
  • Dictionary attacks ineffective
• Protect assets against unauthorized software (e.g., Rooted OS)
  • Measure the boot chain – can use TXT / BootGuard
• Network admission
  • Local network
  • VPN
Trusted Platform Components and TCG Specifications

- TSS
- Platform Firmware (BIOS)
- OS Driver
- Software Interface
- Applications

- PC Client Firmware
- PTP
- PC-Client Platform TPM Profile Spec
- TPM 2.0 Family Library Specification

Other platform profiles: Server, Mobile, Automotive...

- Interface (FED4_XXXX) FIFO / CRB
- Required Resources, Algorithms, PCRs, etc.
- TPM 2.0 Firmware
PCR: Collecting platform measurements
Roots of trust
• To trust a platform, we need to know
  1. The HW identity is what we expect
  2. SW stack can be trusted
• TCG defines three “roots of trust” in a trusted platform
• Root of Trust for Measurement (RTM)
  • The first set of instructions executed when a new chain of trust is established.
  
• Root of Trust for Storage (RTS)
  • A shielded location that cannot be accessed by CPU by any mean other than TPM command

• Root of Trust for Reporting (RTR)
  • Attests the HW identity of the platform
• **CRTM** – Core Root of Trust for Measurement
  - The first component executed after TPM_INIT or D-RTM Event
  - Immutable
  - Inherently trusted

• **Chain of Trust** – a series of recordings resulting in a log, allowing audit of execution sequence.
PCR= Platform Configuration Register

• A PCR can only be extended, not written:

\[
\text{TPM\_PCRExtend}(n, \text{digest}) := \\
\text{pcr}[n] \leftarrow \text{hash}(\text{pcr}[n] \mid \text{digest})
\]

“Extending” allows to store more than one measurement in a limited space

• nominate particular PCRs to hold sequence of measurements at a particular phase in the boot
  • E.g. PCR0 used for ACM and BIOS measurements

• PCRs may be read by anyone – not a secret

• Access to objects may be tied to particular PCR values
TPM usage: Keys and Hierarchies
TPM objects – keys and data

• TPM can create, use and protect objects (which can be either data objects or keys for signing or encryption)

• Signing keys usages:
  • Remote Attestation
  • General purpose signing

• Encryption keys usages:
  • Data “sealing” (or local attestation)
  • General purpose encryption
  • Protecting other keys
    • A hierarchy of keys can be generated this way
Local Attestation: Unseal Data

- **Send to TPM**
  - Sealed Data + auth value
  - Define policy for unsealing - tie to specific PCR values

- **TPM decrypts data**
  - Using an Encryption key
  - If and only if Hash[PCR] from Sealed data == current PCR values
    - i.e. Platform is in the Trusted State
  - Auth value must match

Unsealed data returned to Trusted Environment
Remote Attestation: Quote

Basic feature of trusted computing is an ability of platform to report its current execution snapshot encoded in PCR values to remote site.

Example: TPM Quote

- TPM Quote is signed data blob containing PCR values

Requester sends qualifying data, list of PCRs

TPM creates response using qualifying data (added to hash) and list of PCRs

TPM signs response using Attestation Key

Sends signed response to Requester
Virtual smart cards

- Virtual smart cards are SW devices that act as a physical smart card, using the TPM’s cryptographic abilities.
- The key associated with the VSC is created by the TPM as part of the storage hierarchy, and can be stored off-chip, e.g. the computer’s hard drive.
- The VSC meets the main criteria a traditional smart card meets:
  - Non-Exportability: Using the key is only possible on the TPM associated with the computer itself
  - Isolated cryptography: the cryptographic operations occur on the TPM itself which is isolated from the main processor
  - Anti-hammering: The TPM has a built-in dictionary attack mechanism to prevent hammering on the user’s PIN.
Storage Keys

• A key used to protect other TPM keys is a **Storage Key**.
• The purpose of a storage key is to have a place to attach keys generated somewhere else
  • IT creates a key and wants to put on your system but only wants the TPM to be able to use the key
  • Give the IT department the public key of a Storage Key on your TPM
  • IT encrypts their key to the TPM key on your computer
  • Sends the encrypted bundle to you
  • Your TPM imports the key
• Keys can also be created internally and attached to a storage key
  • E.g. Attestation keys
A storage key is a restricted, asymmetric key. When a storage key is created, the TPM adds a seed value that is used to create protection values for other objects.
Unwrap (inside the TPM). Then encrypt with a symmetric key derived from the seed in the storage key.

Import to Storage Key

Use the public part of the storage key to wrap the key before it is imported.

Now we have a blob that can be stored off-chip and loaded to the TPM when needed.
TPM hierarchies

• TPM2.0 can support up to 3 storage hierarchies
  - Storage hierarchy: for usage by OS user – allows protecting keys and data
  - Endorsement hierarchy: for keeping “Attestation Keys”. Root of the hierarchy is the Endorsement Key (EK)
    - EK is unique per platform
    - An EK certificate can be provided by the TPM vendor
    - A certified EK can be used to generate additional certified attestation keys
  - Platform hierarchy: for usage by BIOS – allows orthogonal use of TPM by BIOS and OS

• Each hierarchy also represents a control surface to the TPM – for example, commands can be authorized by the platform auth value to make sure they are executed in a controlled environment.
## TPM1.2 vs TPM2.0

<table>
<thead>
<tr>
<th>TPM1.2</th>
<th>TPM2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed crypto algorithm set: RSA, SHA-1, AES</td>
<td>Flexible algorithm set</td>
</tr>
<tr>
<td>Command authorization using auth value</td>
<td>Add enhanced authorization: operations can be tied to</td>
</tr>
<tr>
<td></td>
<td>varied and multiple factors</td>
</tr>
<tr>
<td>Single storage hierarchy and ownership</td>
<td>3 hierarchies, allowing orthogonal use by BIOS and OS</td>
</tr>
<tr>
<td>Cumbersome provisioning required to start using the</td>
<td>Simplified provisioning – TPM can be used out of the</td>
</tr>
<tr>
<td>TPM</td>
<td>box</td>
</tr>
<tr>
<td>Assets protected internally using asymmetric</td>
<td>Assets protected internally using symmetric algorithm</td>
</tr>
<tr>
<td>algorithm</td>
<td></td>
</tr>
</tbody>
</table>
Intel® PTT
Intel® PTT (Platform Trust Technology) implementation

- PTT is a TPM2.0 implementation implemented as a FW application in ME
  - An OEM can use either a discrete TPM part, or use the Intel® PTT embedded in FW
- HW interface implemented in ME HW
- RAM is isolated from host access and other applications access by ME task isolation mechanisms
- NVRAM data is protected using blob mechanism
  - Integrity
  - Confidentiality
  - Anti-Replay
- A subset of TPM commands is available during the FW bring-up to allow early access to PTT by the host
References

• TCG website:
• http://www.trustedcomputinggroup.org
• “A Practical Guide to TPM2.0: Using the Trusted Platform Module in the New Age of Security” by Will Arthur and David Challener
Backup
PCR operations

• A PCR cannot be written, only “extended”
  • $TPM_{\text{PCRExtend}}(n, \text{digest}) :=$
    $$pcr[n] \leftarrow \text{hash}(pcr[n] \mid \text{digest})$$
  • $TPM_{\text{PCREvent}}(n, \text{data}) :=$
    $$pcr[n] \leftarrow \text{hash}(pcr[n] \mid \text{hash}([\text{data}])$$

• PCRs can be read by anyone – not a secret
  • $TPM_{\text{PCRRead}}(n) := \textbf{output} PRC[n]$

• Access to TPM objects may be tied to a specific PCR value

• PCR ‘quote’ operation, for nonce $i$:
  • $TPM_{\text{Quote}}({n_1, \ldots n_m}, i, \text{auth}, k) :=$
    • $\text{output} ({PCR[n_1], \ldots PCR[n_m], i}_{\text{key}(k)})$
History of TCG

• TCPA (Trusted Computing Platform Alliance)
  • “Letter-based” organization
  • Compaq, IBM, Intel, Hewlett-Packard, Microsoft
  • No formal governance
  • Defined TPM 1.1b, 1.2

• TCG (Trusted computing Group)
  • Incorporated “non-profit” organization
  • Multiple membership Levels
    • Promoter; Contributor; Adopter
      • + Liaison & other new Levels
  • TCG took over TPM 1.2 Spec
Chains of Trust: Intel Secure boot technologies

Static Trust Chain

TPM_INIT -> BtG ACM

PCRs:
- S-CRTM
- PCR

PCR

Dynamic Trust Chain

SENTER

PCR

Unmeasured Code

D-CRTM

PCR

PCR

PCR
Locality

• Access Control to TPM Resources

Locality provides identity (source authentication) of component accessing TPM