presented by

The TPM 2.0 specs are here, now what?

UEFI Spring Plugfest – March 29-31, 2016
Presented by Dick Wilkins, Ph.D.
Phoenix Technologies, Ltd.

Agenda

- What I said in 2013
- What is happening now
- What is ready to go
- What still needs work
- Call to action
- Questions?

Summerfest July 2013

- I did a presentation on TPM 2.0 at the plugfest at Microsoft in July 2013
- I described the minimal necessary interface changes to talk to the new hardware
- TCG is finally publishing the TPM 2.0 firmware specs, now what?

Background

- A TPM is:
  - Tamper-resistant functionality, state and operations (hardware and/or software)
  - Protected storage for keys and certificates
  - Platform Configuration Registers (PCRs)
  - Cryptographic engine
  - And more

PCRs

- Cannot be written directly
  - *extend(i, v) := pcr[i] ← hash(pcr[i], v)*
- Extending PCRs with hashes of code and data during boot can be compared to previous boots
- Firmware created log entries allow detection of "where things went wrong"
- This approach allows "*measured*" and "*authenticated*" boot
- This is not the same as "secure boot" as enabled by the UEFI spec

Sealed Storage

- Sealing uses TPM cryptographic support with PCRs to provide secure storage
  - "*sealing*" provides a key, a set of PCR values and some data
  - The result of sealing is a "*blob*" of data
    - That can only be unsealed by the TPM that sealed it
    - Can only be unsealed if the current PCRs match those used to seal the data

TPM 2.0 algorithm flexibility

TPM 1.2

- Support for three algorithms
  - SHA1 – hash
  - RSA – asymmetric
  - XOR – symmetric
  - AES is allowed in limited cases

TPM 2.0

- Support for:
  - Any hash algorithm with a fixed digest size
  - Any asymmetric algorithm that has a public and private portion
  - Any symmetric algorithm

Why is this important?

- TPM 2.0 is important in the marketplace
- The TCG PC Client Working Group has finally released specs describing how firmware should implement the TPM 2.0 functionality
- Most of the UEFI community has not been paying much attention
- Code changes are coming

Applicable TCG Specs

- PC Client Platform Physical Presence Interface Specification V1.30 Rev .52
  Published, July 2015
- PC Client Platform Firmware Profile Specification, Rev .21
  Draft, Under Public Review
- EFI Protocol Specification, Rev .13
  Publishing today! (30-Mar-2016)

http://www.trustedcomputinggroup.org/resources/specifications_in_public_review

What’s new?

- Back in 2013, I described the low-level code differences to talk to the new TPM
- Now that specifications for how to use it are being published…
- Significant code changes will be required in firmware and OSes to conform to the specs
- These are not all in EDK II yet

What is in EDK II

(But not everyone has picked up the code changes)

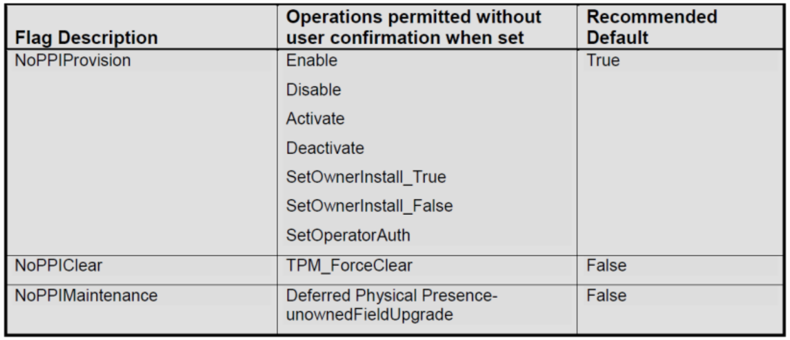
A new EFI Protocol for TCG

- EFI_TREE_PROTOCOL has been deprecated and replaced by the EFI_TCG2_PROTOCOL. Some of the interfaces include:
  - GetCapability function provides more information about the TPM2
    - Identify supported algorithms, such as SHA1/256/384/512/SM3_256
    - Query supported event log formats, such as new TCG2 event log structure. (The previous EFI_TREE_PROTOCOL only supported EFI_TCG2_EVENT_LOG_FORMAT_TCG_1_2, the new EFI_TCG2_PROTOCOL adds EFI_TCG2_EVENT_LOG_FORMAT_TCG_2 type)
    - NumberofPcrBanks – Maximum number of PCR banks (hash algorithms) supported
    - ActivePcrBanks – a bitmap of currently active PCR banks (hash algorithms)
  - GetEventLog function provides the user the ability to retrieve the event log base on TCG1.2 or TCG2.0 structure. TCG1.2 structure only provides SHA1 digests, but TCG2 structure provides different hash digest formats.
  - GetActivePcrBanks function return currently active PCR banks. (New function to support the TPM2)
  - SetActivePcrBanks function sets the currently active PCR banks. (New function to support the TPM2)
    - FYI, All the events MUST be extended into all allocated banks for a PCR.
  - GetResultOfSetActivePcrBanks function retrieves the result of a previous invocation of SetActivePcrBanks. (New function to support the TPM2)

Physical Presence Indicators

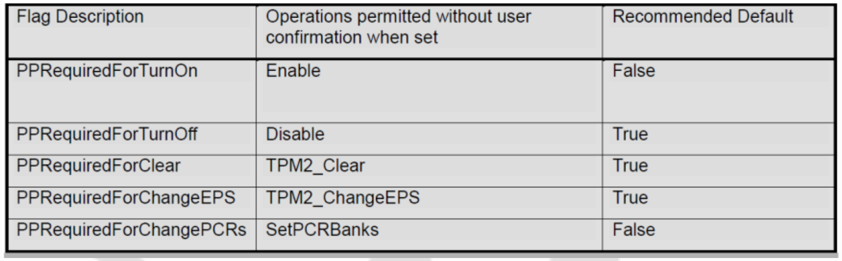
- EDKII has extended the PPI operations (23 thru 34) to support TPM2 only operations (i.e. Operation 23 – SetPCRBanks and 33 - LogAllDigests).
- Persistent Firmware Management Flags and their default settings have changed

TPM 1.2



Flag Description	Operations permitted without user confirmation when set	Recommended Default
NoPPIProvision	Enable Disable Activate Deactivate SetOwnerInstall_True SetOwnerInstall_False SetOperatorAuth	True
NoPPIClear	TPM_ForceClear	False
NoPPIMaintenance	Deferred Physical Presence-unownedFieldUpgrade	False

TPM 2.0



Flag Description	Operations permitted without user confirmation when set	Recommended Default
PPRequiredForTurnOn	Enable	False
PPRequiredForTurnOff	Disable	True
PPRequiredForClear	TPM2_Clear	True
PPRequiredForChangeEPS	TPM2_ChangeEPS	True
PPRequiredForChangePCRs	SetPCRBanks	False

TCG Event Log Changes

- TCG2 event logs are now located in the UEFI configuration table
- Event log consumers may now retrieve the event log via the TCG2 EFI protocol GetEventLog API (No longer retrievable directly from the ACPI tables)
- The PC Client Platform Firmware Profile Spec, Section 2.4.2.2 – *Errors Recording Measurements* states:

  If the measurement of the SRTM, POST BIOS or Embedded UEFI Drivers cannot be made, the SRTM MUST be capped by extending the digest of 00000001h to each PCR[0-7], and an EV_SEPARATOR event per Section 9.3.1 (Event Types) SHOULD be recorded in the event

What hasn’t made it into EDK II yet?

Thanks to David Liu, SW Designer with Phoenix Technologies - Taipei

What is missing from EDK II

- There are a series of specific technical changes called out in the TCG specifications for TPM 2.0 that have not yet made it into the EDKII and Tianocore codebase.
- The following is probably not and exhaustive list but it catches many of the high points
- Clearly there is a lot of work that needs to happen in order for everyone to comply with these specifications

Errors at TPM Initialization

- Section 2.4.2.1, If the startup command fails during TPM Initialization…

  The platform must make the TPM interface inaccessible via hardware for the remainder of the power cycle or Reboot/Disable the Host Platform

- To make this happen (a proposal):

Making the TPM Inaccessible

- Based on Section 7.3.3 - Off to S0 (Working), firmware should always call the TPM2_STARTUP(CLEAR) command
- If firmware wants to make TPM invisible, issue a TPM2_HierarchyControl (EH Disable and SH Disable)
- If firmware wants to make the TPM visible to the OS, the platform manufacturer MUST set platformAuth and MAY set platformPolicy during execution of the SRTM such that later software is unable change objects in the Platform Hierarchy or operations that require platform authorization
- The former option should make the TPM inaccessible

PCR Usage Changes

- PCR[0-7] –
  - Per Section 2.4.1, the digest of 00000000h or FFFFFFFFh MUST be extended in PCR[0-7] and an EV_SEPARATOR event MUST be recorded in the event log for PCR[0-7] prior to the first invocation of the first Ready to Boot call if the TPM is disabled or hidden.
  - Per Section 2.4.2 (Error conditions), if an error occurs, the digest of 00000001h MUST be extended in PCR[0-7] and an EV_SEPARATOR event SHOULD be recorded in the event log for each PCR. See Section 9.3.1 (Event Types).

“MUST Implement” PCR Changes

- PCR [0]
  - The VendorID and ReferenceManifestGUID of the platform firmware which either contains or measures the UEFI Boot Services and UEFI Run Time Services using EV_NO_ACTION event ReferenceManifestEvent. This event is only required if the platform supports NIST SP800-155. This event is not extended. (optional, only if need to support SP800-155)
  - The event type EV_NO_ACTION Startup Locality Event to record the locality from which the TPM2_Startup command was sent. See Section 9.3.4.3 (Startup Locality Event).
  - EV_POST_CODE event should cap with different strings such as "POST_CODE", "SMM CODE" or "Embedded UEFI Driver". Now it only cap with "ACPI DATA" with TPM ACPI table.
  - the digest of 00000000h or FFFFFFFFh MUST be extended in PCR[0-7] and an EV_SEPARATOR event MUST be recorded in the event log for PCR[0-7] prior to the first invocation of the first Ready to Boot call even if the TPM is disabled or hidden.
  - If Platform Firmware loads a CPU microcode update, CPU microcode has to be measured with EV_POST_CODE.

“MUST Implement” PCR Changes

- PCR [1] –
  - If Platform Firmware loads a CPU microcode update, it MUST be measured, using event type EV_CPU_MICROCODE. Alternatively, CPU microcode updates MAY be measured in PCR[0] as part of a EV_POST_CODE event.
  - EFI Boot#### and UEFI BootOrder variables MUST be measured using event type EV_EFI_VARIABLE_BOOT.
    (**Current EDK code measures the boot related variables into PCR [5] instead PCR [1], but according to the TCG2 spec, it should measure them into PCR [1]. The old specification “TCG EFI Platform Specification, Version 1.20 Final, Revision 1.0, 7 June 2006”, specifies boot related variables should be measured to PCR [5], but it has already changed since “TCG EFI Platform Specification For TPM Family 1.1 or 1.2, Specification Version 1.22, Revision 15, 27 January 2014”. EDKII code needs to be updated to implement this according to the new specification**)

“MUST Implement” PCR Changes

- PCR [4] –
  - If the platform firmware is configured or designed to not record each attempt to boot a device, an EV_OMIT_BOOT_DEVICE_EVENTS event MUST be measured once.
- PCR [6] –
  - The Host Platform Manufacturer MAY define the purpose of this PCR, but it MUST record a corresponding log entry. (This PCR is available for OEM use)
- PCR [7] –
  - Should also measure additional SecureBoot variable such as “dbt”, as for the new SecureBoot mode after UEFI spec 2.5, it should also measure the contents of “AuditMode”, ”DeployedMode” and “dbr” variables. If these variables change during boot (mode change), they need to be remeasured.
- PCR [16] –
  - Any component on the Host Platform MAY use and reset PCR[16] at any time for debug purposes. (This PCR is available for debug purposes by OEMs and platform manufacturers)

TPM2 Platform Hierarchy Protection

- Section 10 of the PC Client Platform Profile specification says:
  TPM 2.0 augments the concept of Physical Presence with the Platform Hierarchy authorization. … Because the platform hierarchy is the point of control for the state of the TPM, it is important that the platform hierarchy be properly protected.
  - Firmware SHALL disable the platform hierarchy or
  - MUST set a random value to platformAuth if the hierarchy remains enabled

Call to Action

- All IBVs, OEMs and OSVs need to make themselves aware of these required changes
- They may matter to IHVs as well, particularly those with option ROMs
- Code changes are needed for Tianocore/EDKII, can you help?
- The TPM 2.0 is mandatory on most platforms. So these code changes are now mandatory as well

Closing Remarks

Questions?

Thanks for attending the UEFI Spring Plugfest 2016

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

phoenix® technologies