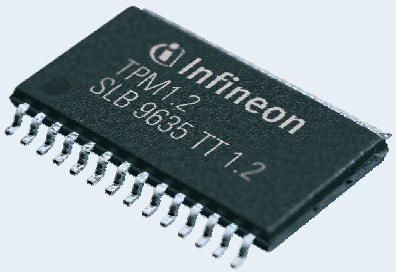
Trusted Platform Module
TPM Fundamental

APTISS, August 2008



Raymond Ng
Infineon Technologies Asia Pacific Pte Ltd
Raymond.ng@infineon.com

Never stop thinking

TPM Fundamental

- Introduction to TPM
- Functional Component of TPM
- Root of Trust
- TPM Keys
- Integration of a TPM into a platform
- Benefits of TPM

Fundamental Trusted Computing Functionality

- Security has become a major challenge for designers and developers of most systems and applications. An attack or unauthorized access can lead to critical loss of data

- A mechanism is required to record (measure) what software is/was running
  - Requires to monitor the boot process
  - Needs an anchor to start the measurement from a Root of Trust
  - Nobody should be able to modify or forge these measurements
  - Some shielded location for the measurements is required

- Now you know that your platform is in a defined state
  - Why should someone else believe this claim?
  - A mechanism to securely report the measurements to a 3rd party is required

- Secure storage
  - Allow access to data only if system is in a known state

- Cost efficient implementation and production

Trusted Computing Group (TCG)

- TCG is a non-profit organization formed to develop, define, and promote open standards for hardware-enabled trusted computing and security technologies, including hardware building blocks and software interfaces across multiple platforms

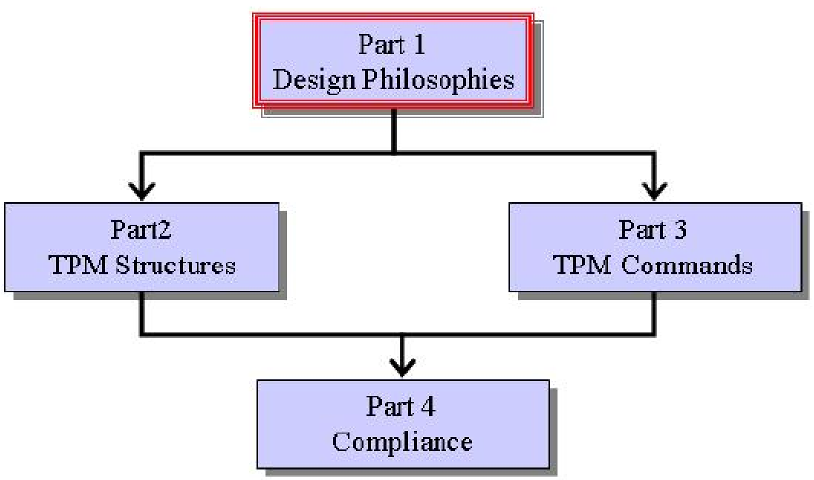
- TCG specifications enable more secure computing environment to protect and strengthen the computing platform against software-based attacks and physical attacks

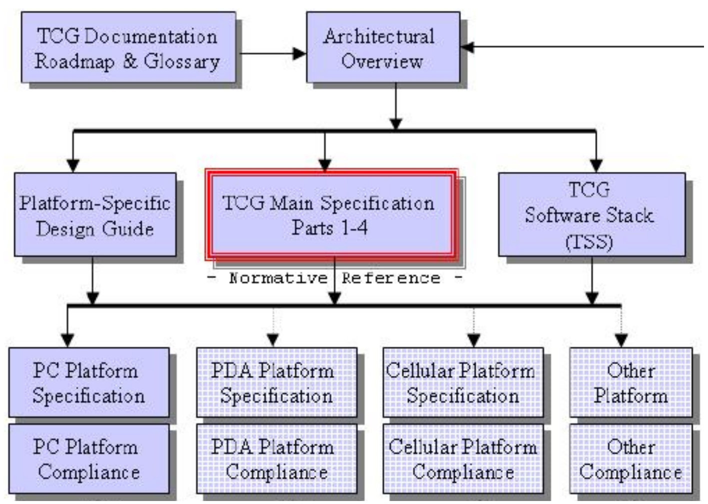
- TCG specifications are freely available from www.trustedcomputinggroup.org

- Trusted Platform Module (TPM) is a major building block to achieve the goals of a trusted computing system

TPM Specification

- TPM specification for 1.2 consists of 4 parts
  - Part 1: Design Principles
    - High-level architectural requirements
    - Defines TPM operational states and authentication protocols
  - Part 2: TPM Structures
    - External data definitions and structures
    - Defines TPM ordinals and general behaviour for each commands
  - Part 3: TPM Commands
    - Detail definition of commands
  - Part 4: Compliance



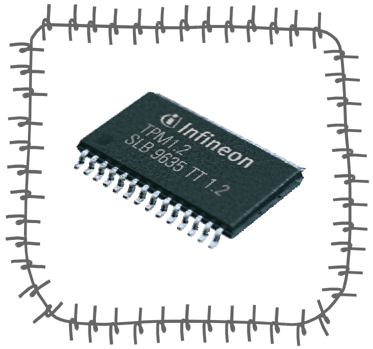


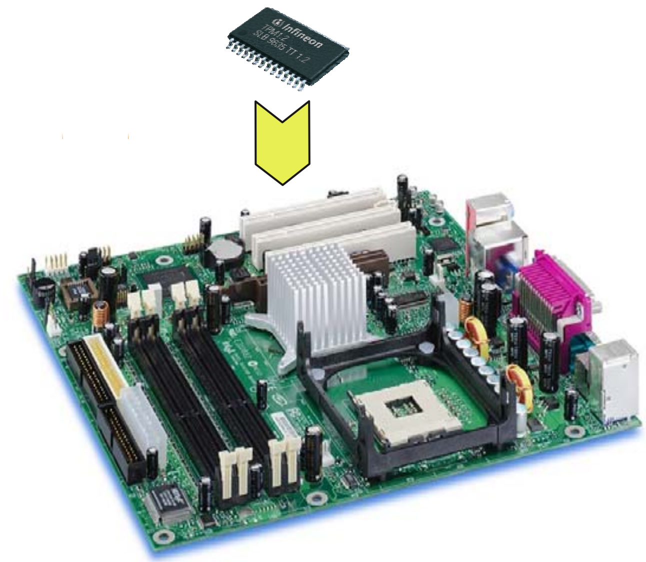
Trusted Platform Module (TPM)

- Specification defines two generic portions of the TPM
  - Shielded locations
    - An area where data is protected against interference from the outside exposure
    - The only functions that can access [read or write] a shielded location is a protected capability
  - Protected capabilities
    - A function whose correct operation is necessary in order for the operation of the TCG subsystem to be trusted
- Both shielded locations and protected capabilities are implemented in hardware and therefore resistant against software attacks
- The TPM is a platform component
  - NOT a platform all by itself
  - TPM becomes a permanent component of the platform

- The TPM is NOT an active component, always a responder to a request and never initiates an interrupt or other such operation
- TPM cannot alter execution flow of system (e.g. booting, execution of applications)

Integrating Trust and Security into Computing Platforms using a Security Chip





Standard Processor System

- Easy to program
- Easy to change
- Easy to attack

TPM-Security Module

- Shielded and encapsulated chip
- Controlled interface to external
- Trusted software in a protected hardware

Trusted platform

=> Security functions, protected against manipulations

TPM Functions and Features Overview

- TPM must be in Hardware
- Has a unique and signed Endorsement Certificate
- TPM MUST be bound (=soldered) to the platform
- TPM provides secure storage for
  - Platform metrics
    - SHA-1 for platform integrity measurements
  - Platform keys/certificates
    - physically and cryptographically bind secrets to a platform
  - User keys/certificates
- Supports an Owner- and User-separation role model
- Seals and binds data/keys/applications to the platform

Common Misconceptions

- The TPM does not measure, monitor or control anything
  - The TPM is a passive device in the system
  - The TPM has no way of knowing what was measured
  - Measurements are made by host software and sent to the TPM
- TPM does not perform bulk encryption
  (e.g. File and Folder encryption or Full Disk encryption)
- Digital Right Management (DRM) is not a goal of TCG specifications
  - All technical aspects of DRM are not inherent in the TPM
- TPM can work with any operating systems or application software
  - The specification is open and the API is defined, no TCG secrets

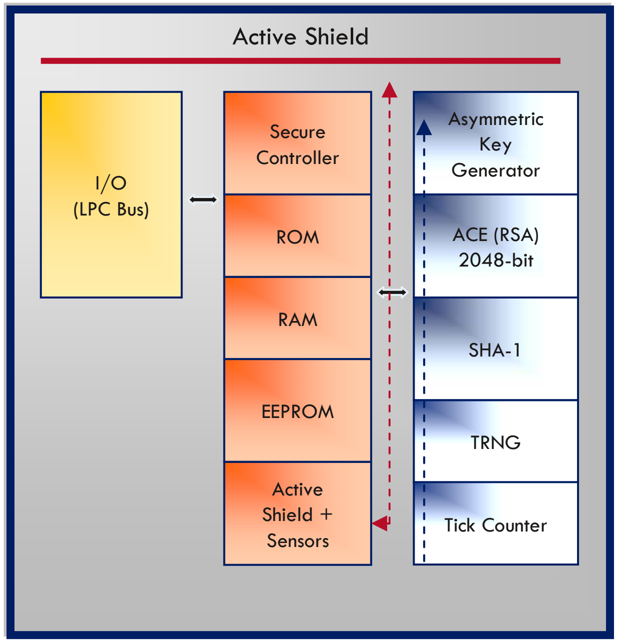
Functional Components of TPM

Raymond Ng
Infineon Technologies Asia Pacific Pte Ltd
Raymond.ng@infineon.com

Never stop thinking

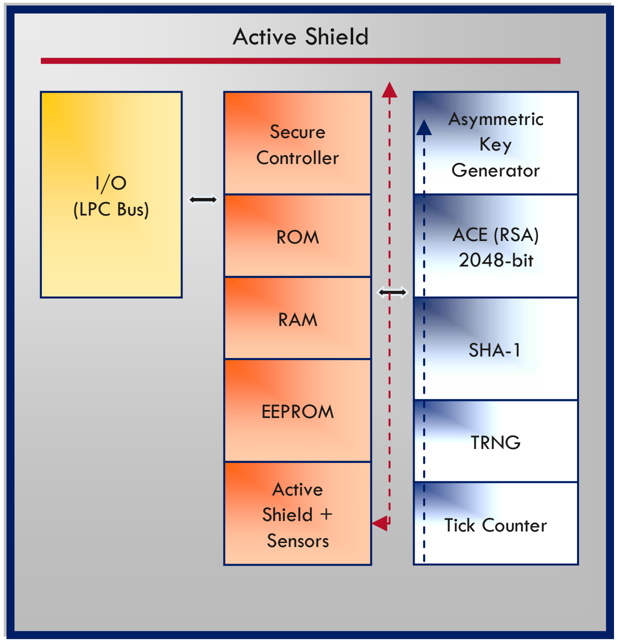
TPM Hardware

- I/O
  - Manages information flow over the communications bus
  - Typically LPC - Low Pin Count Bus
- Secure Controller
  - Command verification
  - Execution of the appropriate command code
  - Controls internal TPM execution flow
- ROM
  - TCG firmware
- EEPROM
  - User data
  - TPM keys [e.g., Endorsement Key (EK) and Storage Root Key (SRK) and owner secret]
  - Endorsement Key Certificate



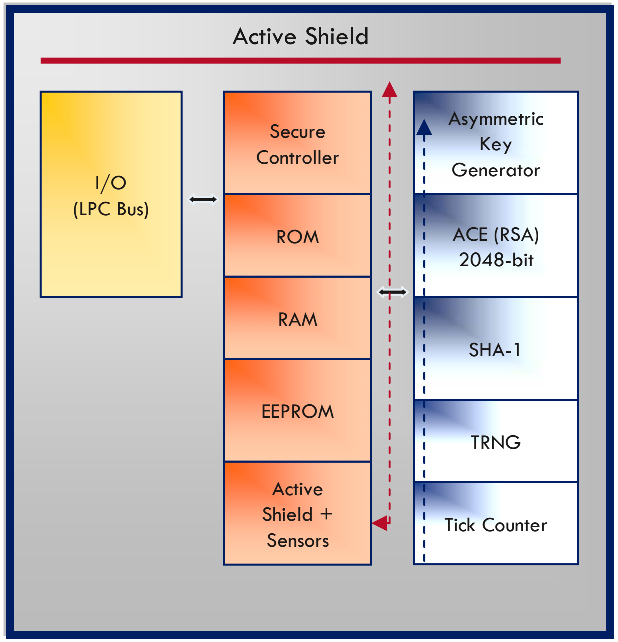
TPM Hardware

- Asymmetric key generation (RSA; storage and key size >= 2048)
  - Support 1024, 2048 bit keys
  - Use of 2048 recommended
  - To use an RSA key it has to be loaded into the TPM
  - The TPM can encrypt and decrypt using RSA keys
  - The use of keys is segregated into signing or encryption uses
- Advanced Crypto Engine (ACE)
  - Asymmetric key operations (up to 2048-bit key length)



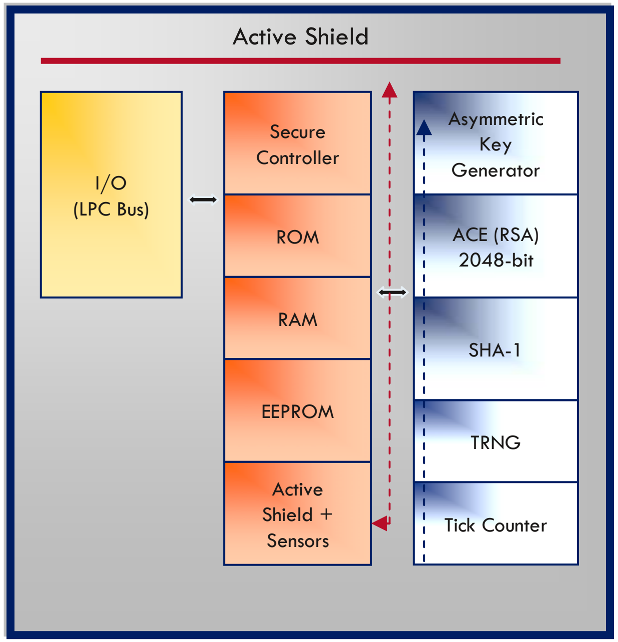
TPM Hardware

- SHA-1 engine (160 bits)
  - SHA-1 for Hashing (measuring of integrity)
  - Primarily used by the TPM as its trusted hash algorithm
  - Exposed to the outside to be used in the boot process
  - TPM is not a crypto accelerator
  - No regular structure
- Random Noise Generator (RNG)
  - Source of randomness in the TPM
  - Used for nonce (Number Used Once) and key generation
  - The RNG output is used both internally by the TPM and is offered to outside consumers of randomness
- Tick counter
  - Provide an audit trail of TPM commands



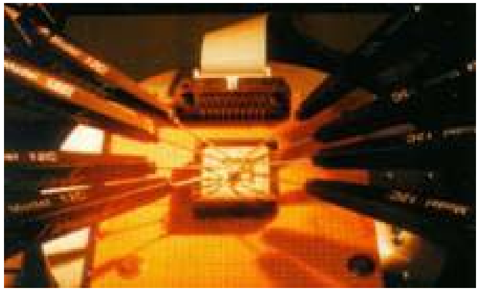
TPM Hardware

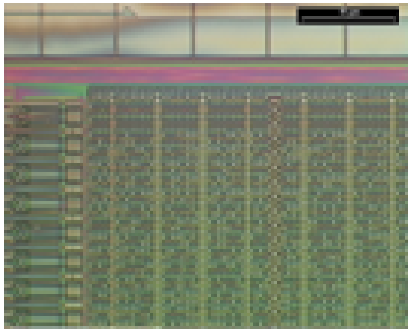
- Security Features
  - Active shield
  - Over/Under voltage detection
  - Low/High frequency sensor
  - Reset filter
  - Memory encryption

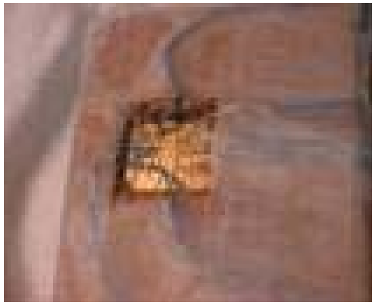


Typical Attacks

- Software attacks
  - Exploit implementation flaws!
- Fault attacks
  - Physical perturbation of Vcc, clock, temperature, UV light, X-Rays
- Side channel attacks
  - monitoring of analogue signals e.g. time, power, electro-magnetic
- Invasive attacks
  - Reverse the content of the ROM
  - Probing data
  - Circuit modification







Root of Trust

Raymond Ng
Infineon Technologies Asia Pacific Pte Ltd
Raymond.ng@infineon.com

Never stop thinking

Roots of Trust

- Root of Trust is a hardware or software mechanism that one implicitly trusts

- Root of Trust for Measurement (RTM)
  - Uses Platform Configuration Registers (PCR) to record the state of a system
  - Static entity like the PC BIOS

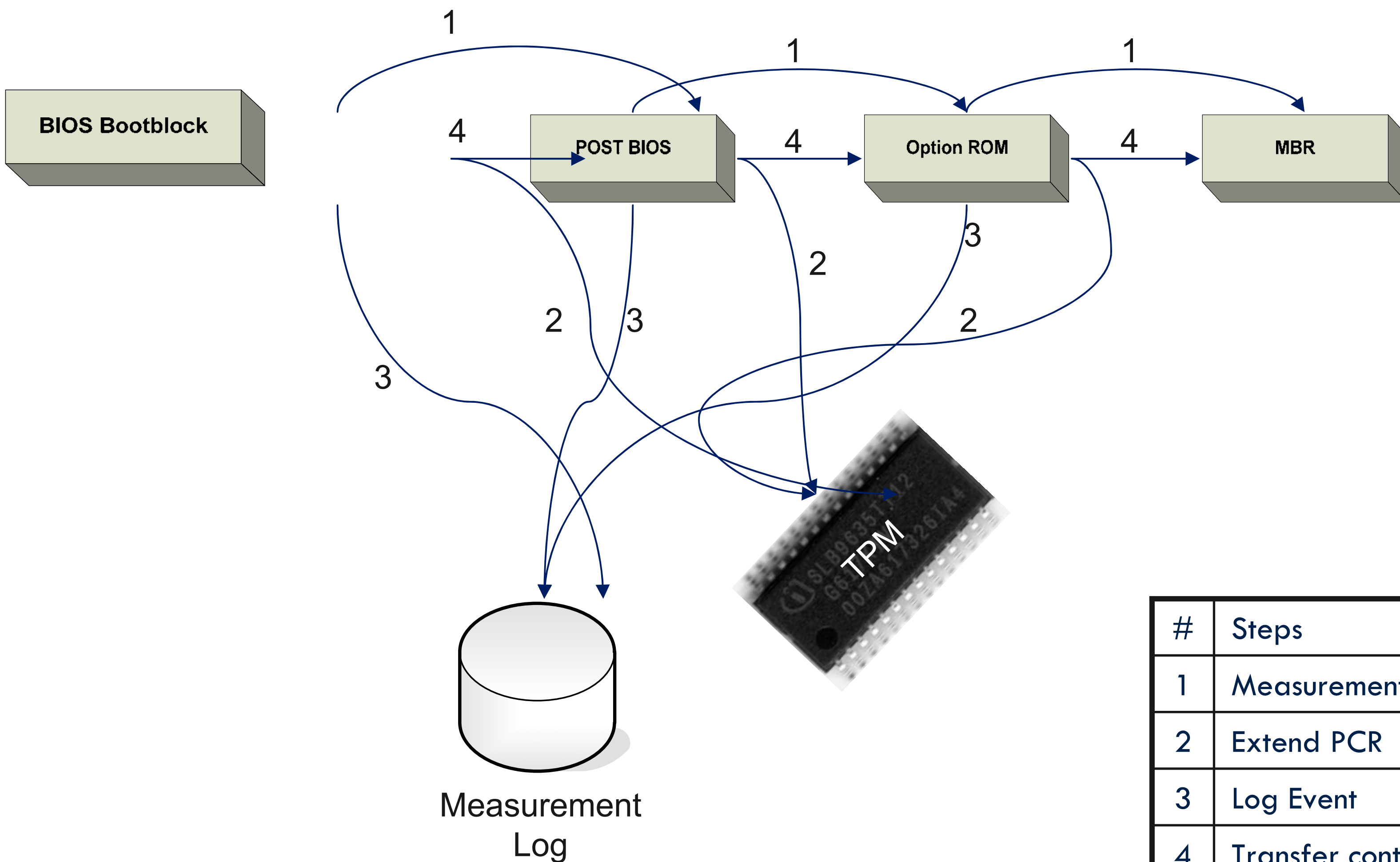
- Root of Trust for Reporting (RTR)
  - Entity trusted to report information accurately and correctly
  - Uses PCR and RSA signatures to report the platform state to external parties in an unforgettable way

- Root of Trust for Storage (RTS)
  - Entity trusted to store information without interference leakage
  - Uses PCR and RSA encryption to protect data and ensure that data can only be accessed if platform is in a known state

Platform Configuration Register (PCR)

- Platform Configuration Registers (PCR) is a 160 bit storage location for integrity measurements
- Shielded location inside TPM
- The integrity measurement of executables is cumulatively stored in a PCR
  - PCR[i] = SHA-1(PCR[i] || newMeasurement)
- PCR extends are not commutative (i.e. measuring A then B does not result in the same PCR value as measuring B then A)
- PCR can keep track of unlimited number of measurements
- What can be measured and cumulatively stored (cannot be overwritten until reboot)
  - BIOS, ROM, Memory Block Register [PCR index 0-4]
  - OS loaders [PCR index 5-7]
  - Operating System (OS) [PCR index 8-15]
  - Debug [PCR index 16]
  - Localities, Trusted OS [PCR index 17-22]
  - Applications specific [PCR index 23]

Chain of Transitive Trust



#	Steps
1	Measurement
2	Extend PCR
3	Log Event
4	Transfer control

PCR Event Log

- Together with PCR extensions also PCR event log entries can be made
- A log entry contains the PCR number, the value that was extended into the PCR and a log message (giving details what was measured)
- The event log does not need to be protected by the TPM and therefore is managed on external mass storage (managed by Trusted Software Stack - TSS)
- The event log can be used to validate the individual steps that lead to the current PCR value
  - Calculate the extends in software starting at the beginning of the log
  - Compare the result to the PCR value in the TPM
  - If the values match the verifier has assurance that the log was not tampered with
- PCR content is digitally signed inside the TPM

Root of Trust for Measurement

- Goal is to measure system state into PCR
- Using PCR a communication party can be convinced that the system is in some known state
- System users are NOT prevented from running any software they want, but the execution is logged and cannot be denied
- From the RTM the trust is extended to other system components. This concept is called transitive trust
- Involved steps:
  - Measure (compute the hash value of) the next entity: e.g. the BIOS measures the OS loader
  - The measurement is extended into one of the TPM PCR
  - Control is passes to the measured entity
- This process is continued for all components of a system up to user level applications
- PC client specifications defines which PCR are used for what
- Measurements change with system updates and patches

Root of Trust for Reporting

- Root of Trust for Reporting (RTR) is a mechanism to securely report that state of a platform to a third party. The idea is to digitally sign the PCR values inside the TPM and send the signature to the requester

- Endorsement Key (EK) forms the RTR
  - 2048 bit RSA key contained inside the TPM
  - Private part never leaves the TPM (only exists in shielded location)
  - EK is unique for every TPM and therefore uniquely identifies a TPM
  - Typically generated by TPM manufacturer in the fab inside the TPM
  - The EK is backed by an EK certificate typically issued by the TPM manufacturer
  - The EK certificate guarantees that the key actually is an EK and is protected by a genuine TPM
  - EK cannot be changed or removed

Root of Trust for Storage

- Root of Trust for Storage (SRK) is the root of the TPM key hierarchy and never leaves the TPM
- Use of TPM keys for encrypting data and keys
- Two approaches
  - Without using PCR: bind/unbind
  - With using PCR: seal/unseal
- Binding
  - Happens outside of the TPM
  - Encrypt data with the public part of a TPM key
  - Only the TPM that the key pair belongs to can decrypt the data and private key can only be used inside the TPM
  - Binding to a specific TPM, use a non-migratable binding key
- Unbinding
  - Decryption of bound data inside the TPM using the private key

Root of Trust for Storage

- Sealing
  - A way to combine measurements (PCR content) and external data
  - Encrypt externally provided data with reference to a specific PCR state
  - Only the TPM that sealed the data can do the unseal (ensured by including a nonce that only is known to this specific TPM)
  - PCR values specified do not have to be the platforms current PCR values but can be some other (future) PCR values
  - Using a storage key
- Unsealing
  - Load key that was used for sealing into TPM
  - Decrypt sealed blob inside TPM
  - TPM checks the tpmProof included in the internal data, if the nonce does not match the one of the TPM it returns an error
  - If the specified PCR values do not match the platforms current PCR values an error is returned

PCR Revisited

- Summary of PCR usage scenarios
  - Protecting data (TPM_Seal/TPM_Unseal)
  - Specify set of PCR upon key creation where key is only usable if these PCR are present
- Collection of measurements is done outside of the TPM by the platform (chain of trust starting at the RTM)
- Chain must not be broken

TPM Keys

Raymond Ng
Infineon Technologies Asia Pacific Pte Ltd
Raymond.ng@infineon.com

Never stop thinking

TPM Keys

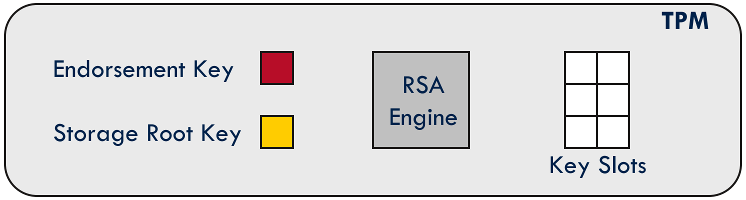
- Endorsement Key (EK)
  - Unique platform identity
  - Created by manufacture in a secure environment
  - Non-migratable, store inside the chip, cannot be remove
- Storage Root Key (SRK)
  - 2048 bit RSA key
  - Is top level element of TPM key hierarchy
  - Created during take ownership
  - Non-migratable, store inside the chip, can be remove
- Storage Keys
  - RSA keys used to wrap (encrypt) other elements in the TPM key hierarchy
  - Created during user initialization
- Signature Keys
  - RSA keys used for signing operations
  - Must be a leaf in the TPM key hierarchy

Take Ownership of a TPM

- TPM is shipped in "unowned" state
- To make proper use of TPM, platform owner has to execute "TakeOwnership" operation
- Setting owner password - inserting a shared secret into the TPM (stored in shielded location)
- Certain TPM operations require owner authorization
- Physical presence allows access to certain (otherwise owner protected) TPM functionality; does not reveal any TPM secrets (e.g., ownership password cannot be revealed using physical presence)
  - ForceClear allows to "clear" the TPM using physical presence
- SRK is created as part of TakeOwnership
- (Private) SRK is stored inside the TPM and never leaves it
- Password required for SRK usage can be set

Creating TPM Keys

- EK and SRK are the only keys permanently stored inside the TPM
- TPM keys are generated inside the TPM
- To use a TPM key, it has to be loaded into the TPM
- Management of key slots is done in software – Trusted Software Stack (TSS)



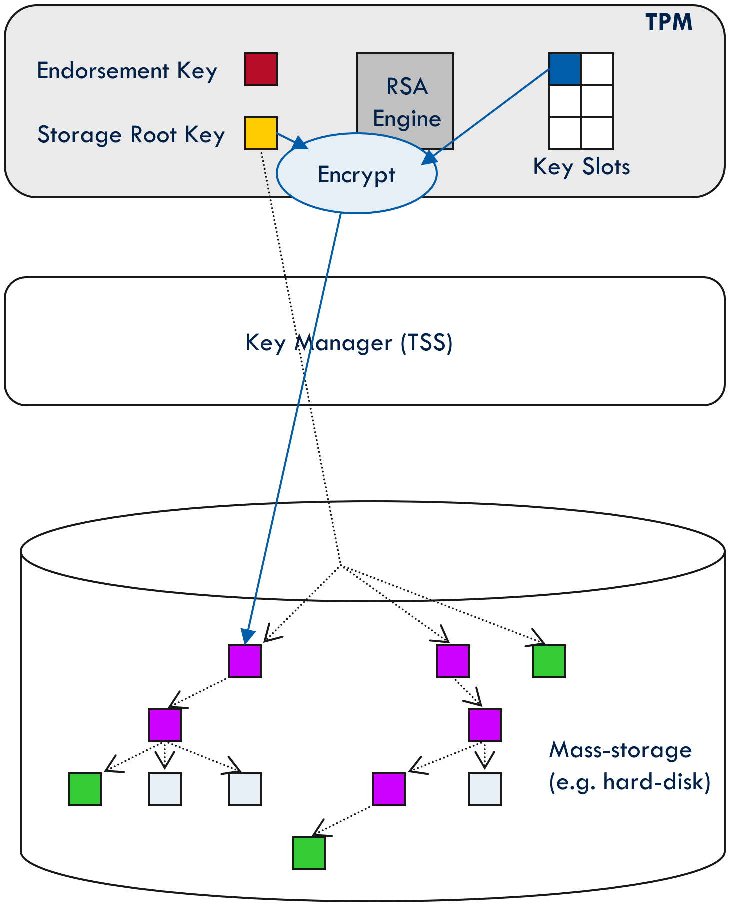
- RSA Engine creates RSA key
- To create a key pair, a parent key has to be specified

TPM Key Hierarchy

- When moving out keys from a TPM a key hierarchy is established
- Whenever a key is exported from the TPM, its private part is encrypted using the public key of the parent
- In TCG terminology the child key is wrapped using the parent key
- Since the parents private key (required to load/decrypt the child key) never leaves the TPM in plain, the private key of a TPM can never be decrypted/used outside of the TPM
- The private SRK, sitting at the top level of the key hierarchy, is never exported from the TPM
- Storage keys form the nodes of the key hierarchy while signing keys always are leaves

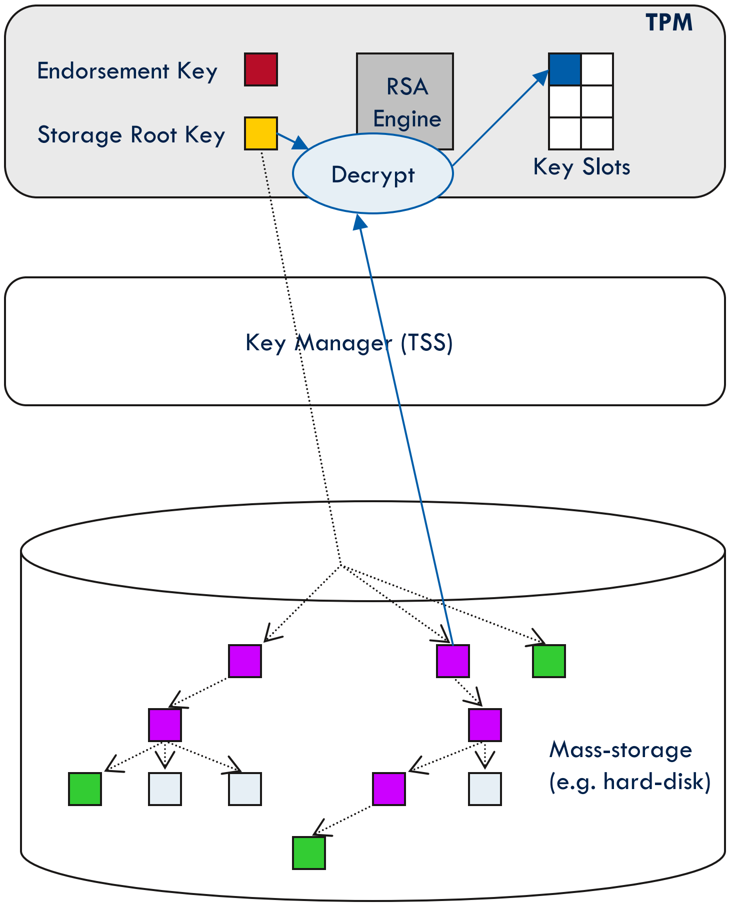
Unloading TPM Keys

- Key hierarchy with SRK as root
- Private SRK never leaves the TPM
- Exporting key blob from TPM
- Private part is encrypted with public parent key before key blob leaves TPM



Loading TPM Keys

- Load signing key into TPM to use it for signing operation
- Establish entire key chain up to SRK
- Decrypt private key of storage key using the private SRK
- Requires SRK usage secret



Clearing a TPM

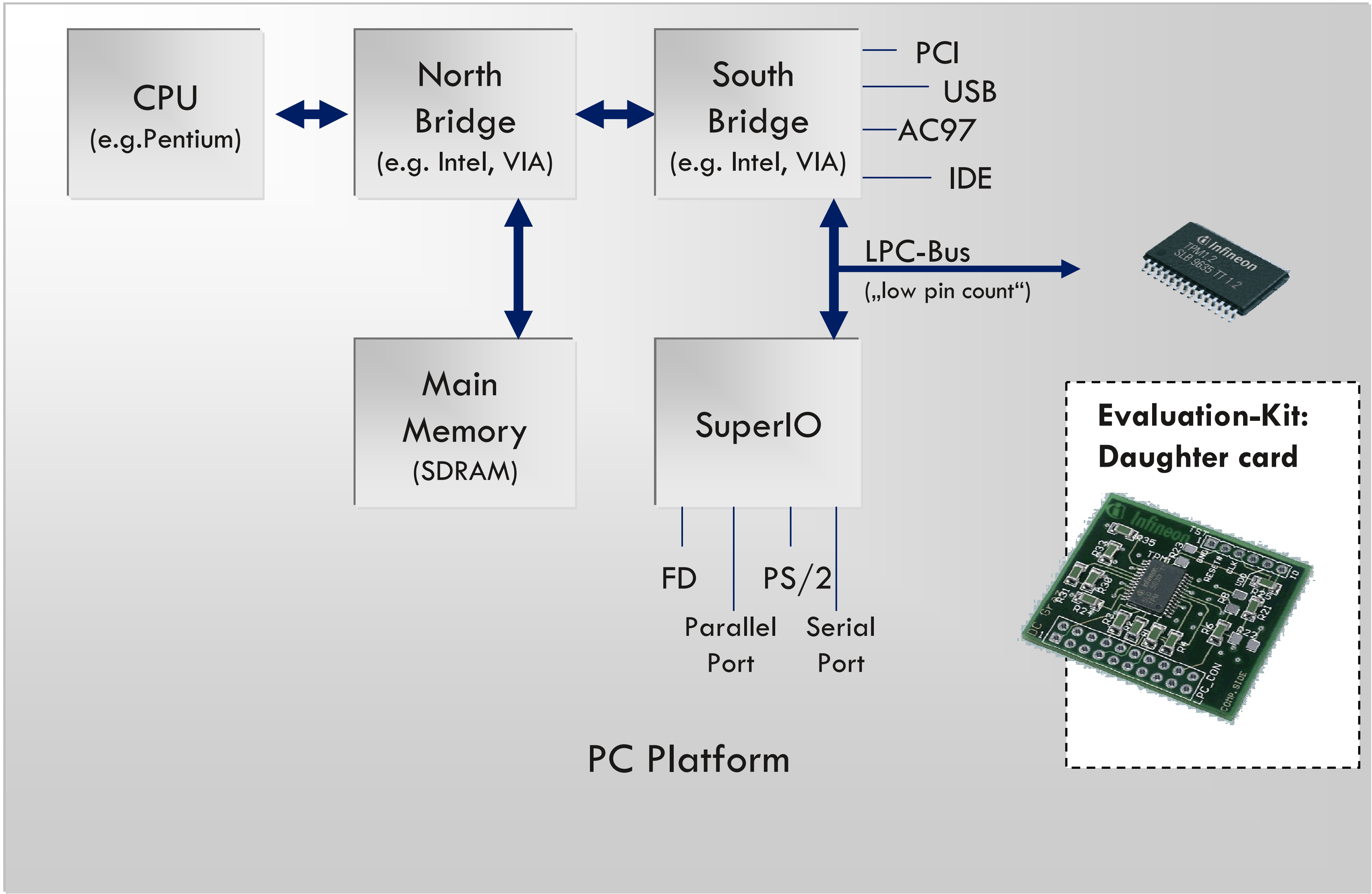
- Resetting the TPM to the factory defaults
- Clearing requires owner secret or physical presence (ForceClear)
- There are no mechanisms to recover a lost TPM owner password
- Tasks executed when clearing the TPM
  - Invalidation of the SRK and thereby all data protected by the SRK will not be able to decrypt
  - Invalidation of the TPM owner authorization value
  - Reset of TPM memory to factory defaults
  - EK is NOT affected
  - PCR values are undefined after clear (reboot required)
- ForceClear is only available during boot (and disabled thereafter)
- OwnerClear can also be disabled (permanent is ForceClear required)

Integration of a TPM into a platform

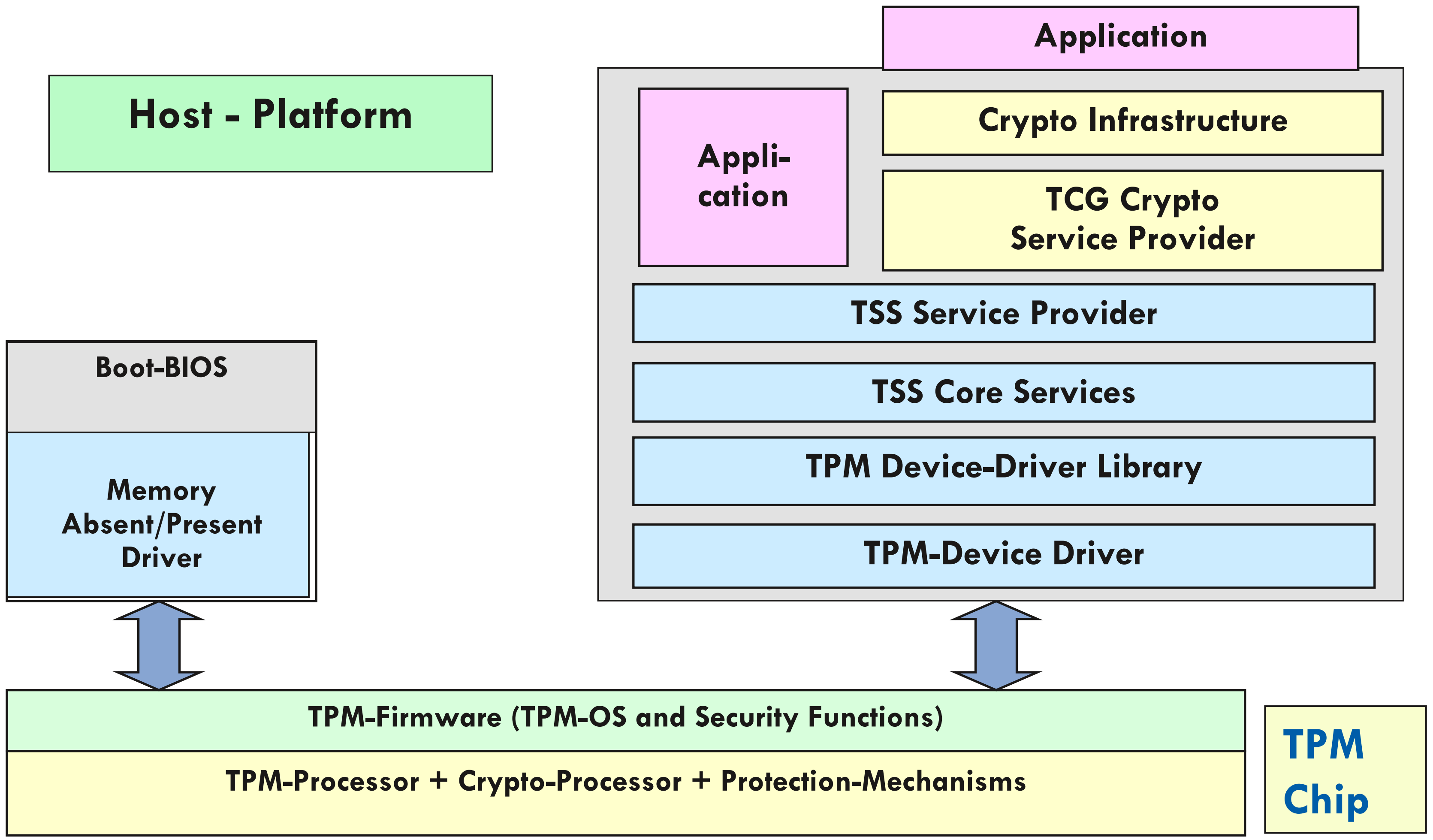
Raymond Ng
Infineon Technologies Asia Pacific Pte Ltd
Raymond.ng@infineon.com

Never stop thinking

PC Motherboard Architecture: TPM is connected to the LPC-Bus



TPM-Driver and API are as important as a TPM-chip: Customer expect availability of a complete solution package



Trusted Software Stack

- TPM Device Driver (TDD)
  - A kernel-mode component that receives byte-streams from TDDL, sends to TPM and then return responses from TPM back to TDDL
  - Handles system power states transitions (S0 – S5) for the TPM chip
- TPM Device Driver Library (TDDL)
  - Provides a user-mode interface
  - A single-instance, single threaded module
  - All TPM commands sent to TDDL must be serialized
- TCG Core Service (TCS)
  - Synchronizes access to the TPM from multiple applications
  - Provides key and authorization context caching
  - Controls the TPM during power mode transitions
- TCG Service Provider (TSP)
  - Persistent storage of keys
  - Handling of Authorization Secrets
  - Handling of Authorization Sessions
  - Encryption of Data
  - Hashing of Data

Benefits of TPM

Raymond Ng
Infineon Technologies Asia Pacific Pte Ltd
Raymond.ng@infineon.com

Never stop thinking

Benefits of TPM

- Enhance confidence in platform
- Proof that a platform is a Trusted Platform
- Binding of data to a particular platform
- Sealing data to a trusted system state/configuration
- Owner privacy and control
- Secure boot
- Low cost exportable technology

Never stop thinking