TCG Architecture Overview

TCG Specification. Architecture Overview Specification,
Revision 1.4, 2nd, August 2007

彭証鴻、賴冠廷

Outline

- Introduction
- Trusted Platform Features
- Trusted Platform Module (TPM)
- Root of Trust, Integrity Measurement, Storage, Reporting
- TPM as an Endpoint of Communication
- Interfacing with TPM and Software Services
- Conclusion

What's TCG (Trusted Computing Group)?

- Organization with the aim of enhancing the **security of the computing environment** in disparate **computer platforms**
- Open Standards & Specifications
- Enhance hardware and OS based trusted computing platform that enhances customers' trusted domains

Trusted Computing

- Trusted
  - the expectation that a device will behave in a particular manner for a specific purpose
- Trusted Computing
  - the computer will **consistently behave in expected ways**, and those behaviors will be enforced by computer hardware and software.
  - Enforcing this behavior is achieved by loading the hardware with a **unique encryption key inaccessible** to the rest of the system and **shielded locations**.

TCG Usage Scenarios

- Risk Management
- Asset Management
- Security Monitoring and Emergency Response

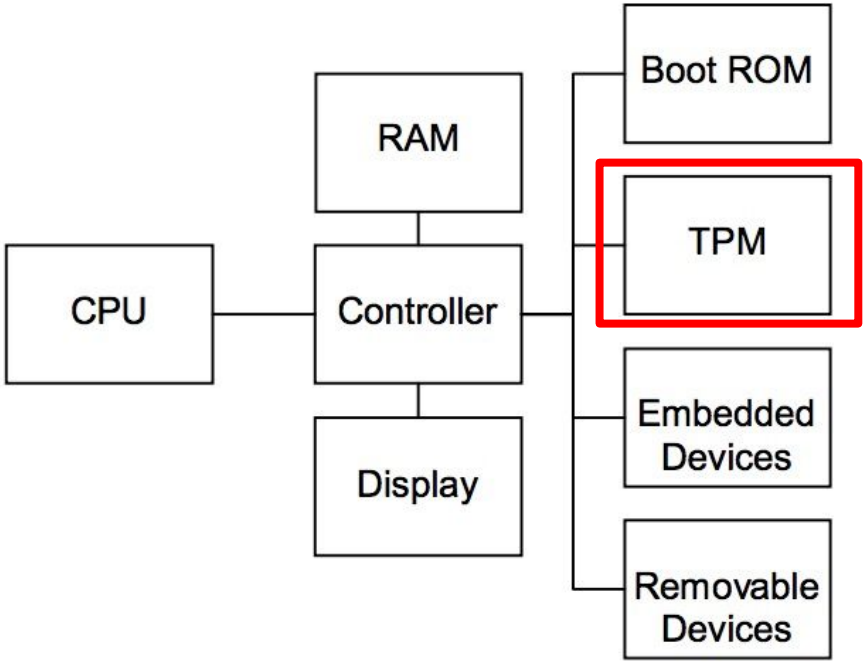


Figure 4:a - Reference PC Platform Containing a TCG Trusted Platform Module (TPM).

Trusted Platform Features(1/3)

1. Protected Capabilities
    - Shielded locations
        - Places (memory, register, etc.) where it is safe to operate on **sensitive data**
        - e.g. Cryptographic Keys, Integrity Measurements
    - A set of commands with **exclusive permission** to access shielded locations.

Trusted Platform Features(2/3)

2. Integrity Measurement, Logging and Reporting
    - Integrity Measurement
        i. Process of obtaining metrics of platform characteristics
        ii. Putting digests of those metrics in PCRs
    - Integrity Logging
        i. Store integrity metrics in a log for later use
    - Integrity Reporting
        i. Process of attesting to integrity measurements recorded in PCRs

Trusted Platform Features(3/3)

3. Attestation
    - Process of vouching for the accuracy of information
    - External entities can attest to shielded locations, protected capabilities, and Roots of Trust
    - Four types of attestation:
        i. Attestation by the TPM
        ii. Attestation to the platform
        iii. Attestation of the platform
        iv. Authentication of the platform

TPM Component

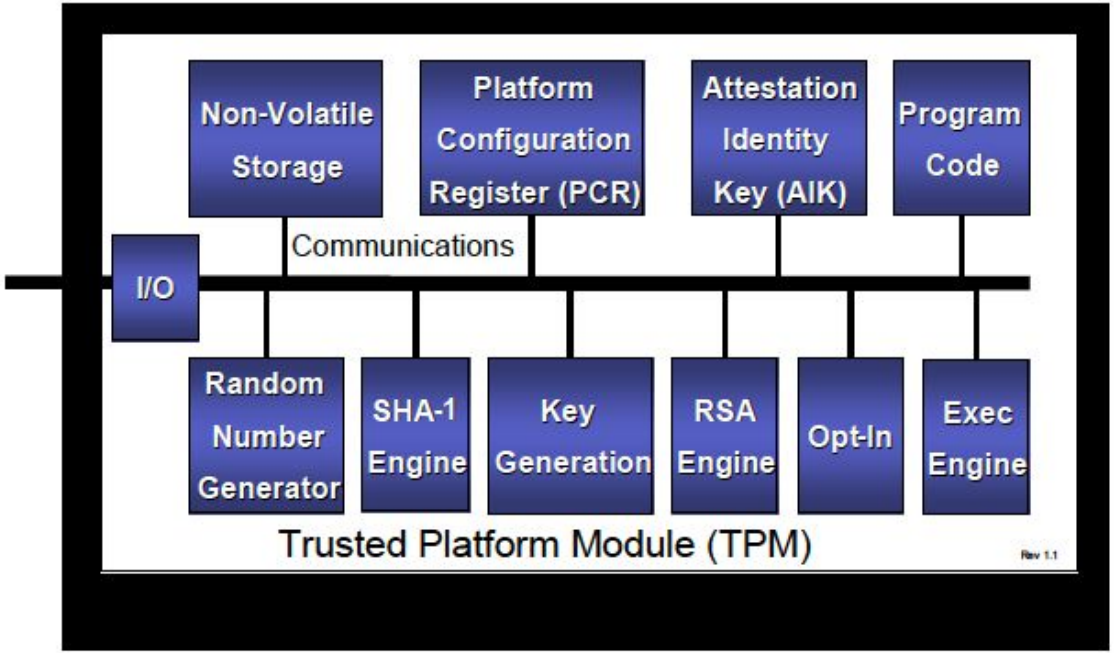


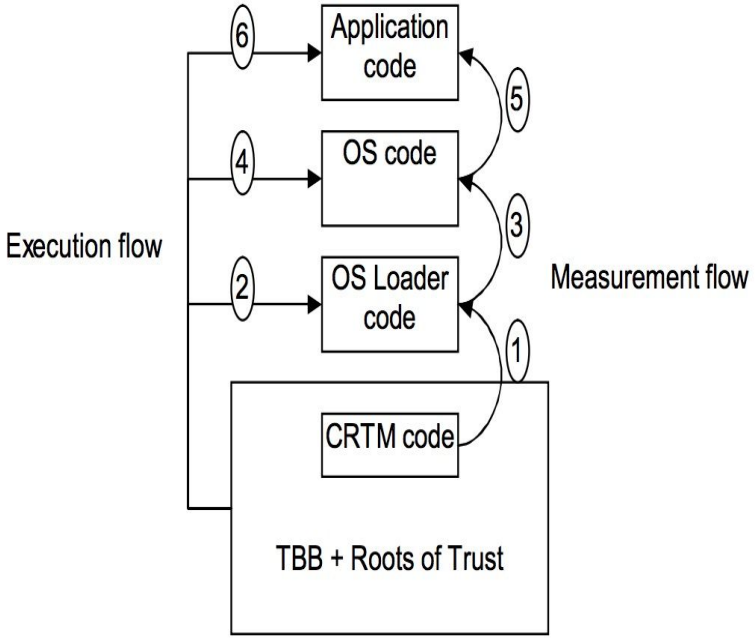
Figure 4:g – TPM Component Architecture

Root of Trust

- Components that **must be trusted** because misbehavior might not be detected.
- Three common *Roots of Trust*
  - RTM (Root of trust for measurement)
    - computing engine capable of making inherently reliable integrity measurements.
  - RTS (Root of trust for storage)
    - computing engine capable of maintaining an accurate summary of values of integrity digests
  - RTR (Root of trust for reporting)
    - computing engine capable of reliably reporting information held by the RTS
- Trusted Building Block (TBB)
  - Instructions for the RTM and TPM initialization functions

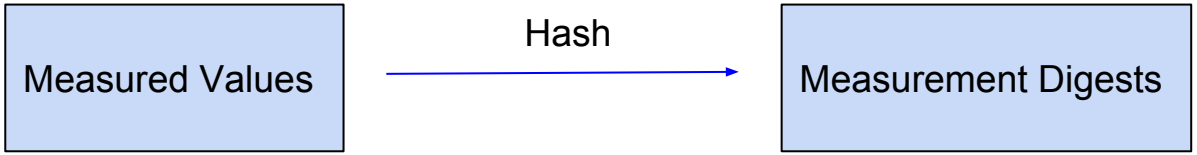
Trust Boundary, Transitive Trust

- Trust Boundary
  - Initial: TBB and Roots of Trust
  - Can be extended
- Transitive Trust
  - Process where the Root of Trust gives a **trustworthy description** of a second group of functions
  - Interested entity determines that the trust level of the second group of functions is acceptable



Integrity Measurement(1/2)

- Measurement Event
  - Kernel scans measured target
  - Consists of two classes of data:
    - Measured Values
      - A representation of **embedded data or program code**
      - Stored virtually **anywhere**
    - Measurement Digests
      - A **hash** of those values
      - Stored in the **TPM** using RTR and RTS functionality



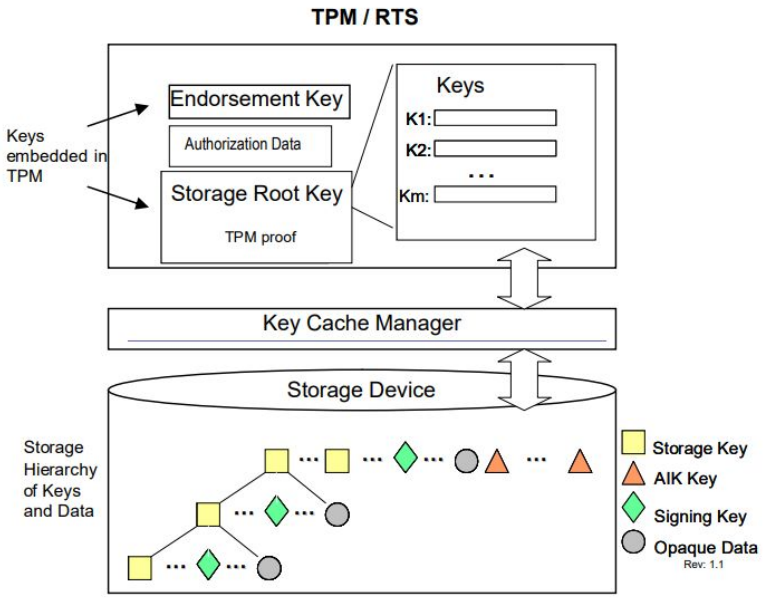
Integrity Measurement(2/2)

- Stored Measurement Log (SML)
  - Contain sequences of related measured values
  - Each sequence contain hash of measurement digest and measured values
- Platform Configuration Registers (PCR)
  - In TPM
  - Updates to a PCR follows as: *PCR[n] <- SHA-1(PCR[n] + measured data)*
  - PCR values are temporal and are reset at system reboot.

Protected Storage(1/3)

- RTS protects keys and data entrusted to the TPM
- Two type of keys: non-migratable/migratable
  - whether a key may be transferred from one TPM to another

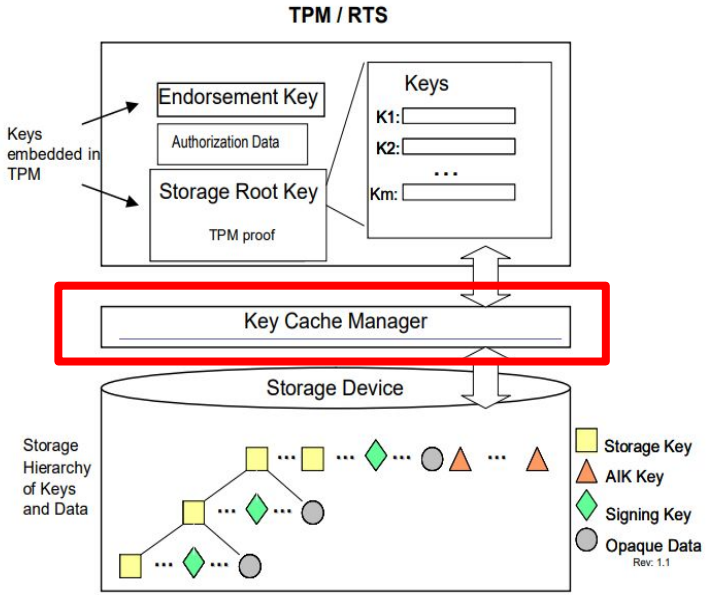
Protected Storage(2/3)



- Endorsement Key (EK)
  - A non-migratable decryption key for the platform
- Storage Root Key (SRK)
  - Used to wrap other TPM keys
- Signing keys
  - Used to sign application data and messages
- Storage keys
  - Used to encrypt data or other keys
- Identity Keys (a.k.a. AIK keys)
  - Used to sign data originated by the TPM

Protected Storage(3/3)

- Key Cache Management (KCM)
  - Interfaces that allow external programs the ability to manage the limited storage resources of the TPM
  - Brokering movement of keys between TPM and external storage
  - Separate the ability to cache keys from the ability to use a key
    - Exception: storage keys



Integrity Reporting

The Root of Trust for Reporting (RTR) has two functions :

1. To expose shielded-locations for storage of integrity measurements.
2. The second objective is to attest to the authenticity of stored value based on trusted platform identities.

Integrity Reporting

- Integrity reports are digitally signed to authenticate PCR values using Attestation Identity Keys (AIK), then report to the challenger.
- A nonce is included with the signed PCRs to prevent replay attack.

Integrity Reporting Protocol

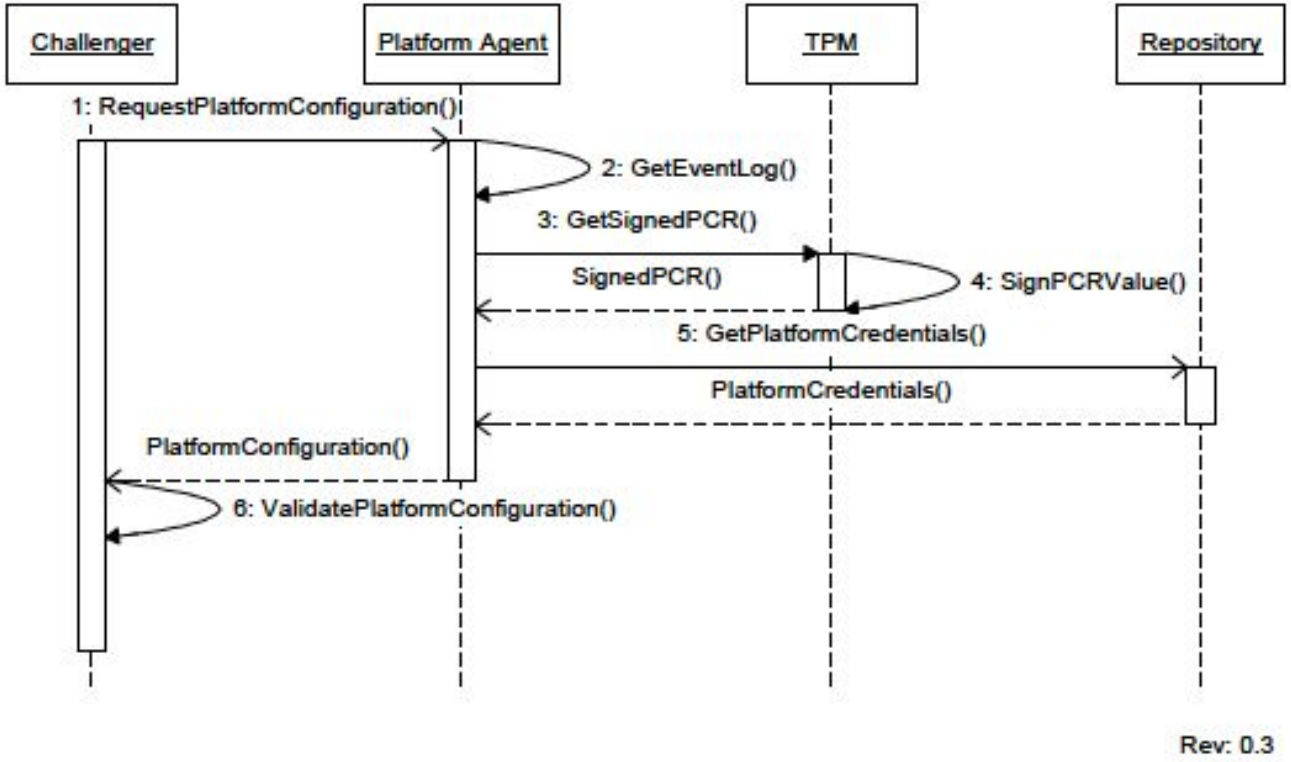


Figure 4:d - Attestation Protocol and Message Exchange

Integrity Reporting Protocol

1. A Challengers requests one or more PCR values from a platform.
2. An agent on the platform containing a TPM, collects SML(Stored Measurement Log) entries.
3. The Platform Agent receives PCR values from the TPM.
4. The TPM signs PCR values using an AIK.

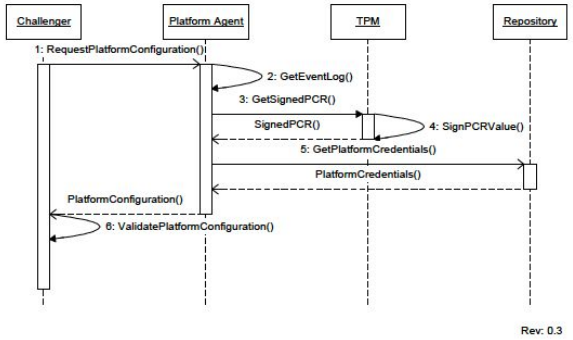


Figure 4:d - Attestation Protocol and Message Exchange

Integrity Reporting Protocol

5. The Platform Agent collects credentials that vouch for the TPM. The signed PCR value, SML entries and Credentials are returned to the Challenger.
6. The Challenger verifies the request. The measurement digest is computed and compared with PCR value. The platform credentials are evaluated and signatures checked.

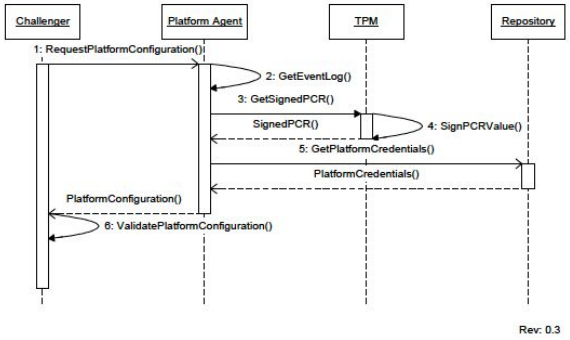


Figure 4:d - Attestation Protocol and Message Exchange

TCG Credentials

- Endorsement or EK credential
  - TPM Manufacturer Name
  - TPM Part Model Number
  - TPM Version of Stepping
  - EK Public Key
- Conformance credential
  - Description of Conformance Entity
  - Pointer to TPM Conformance
  - Pointer to Platform Conformance

TCG Credentials

- Platform credential
  - Platform Manufacturer Name
  - Platform Model Number
  - Platform Version
  - Endorsement Credential
  - Conformance Credential
- Validation credential
  - Validation Entity Name
  - Component Manufacturer Name, Model and Version
  - Measurement Value(s)

TCG Credentials

- Identity or AIK credential
  - It contains the AIK public key, a reference to the part of the Endorsement Credential and Platform Credential.
  - A challenger could use this information, along with other information in the credential to trust the platform via Attestation protocol.

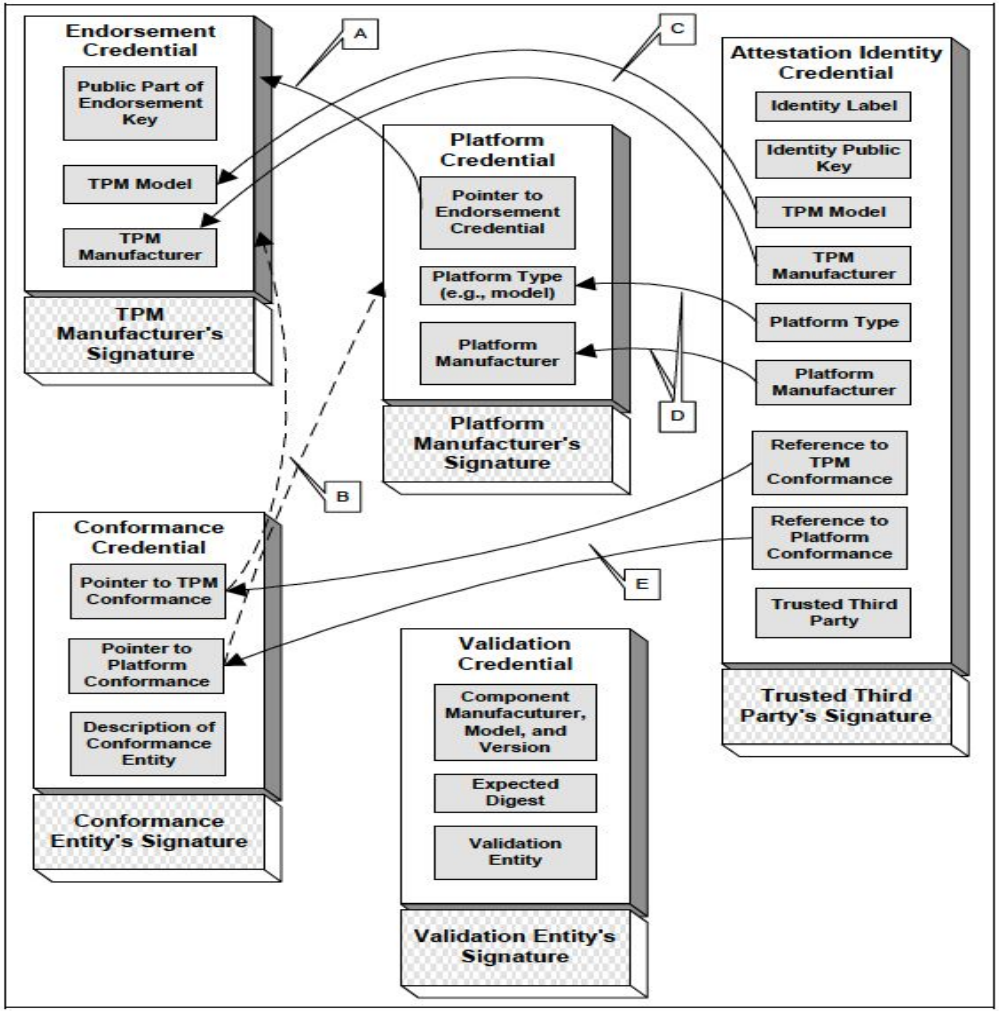


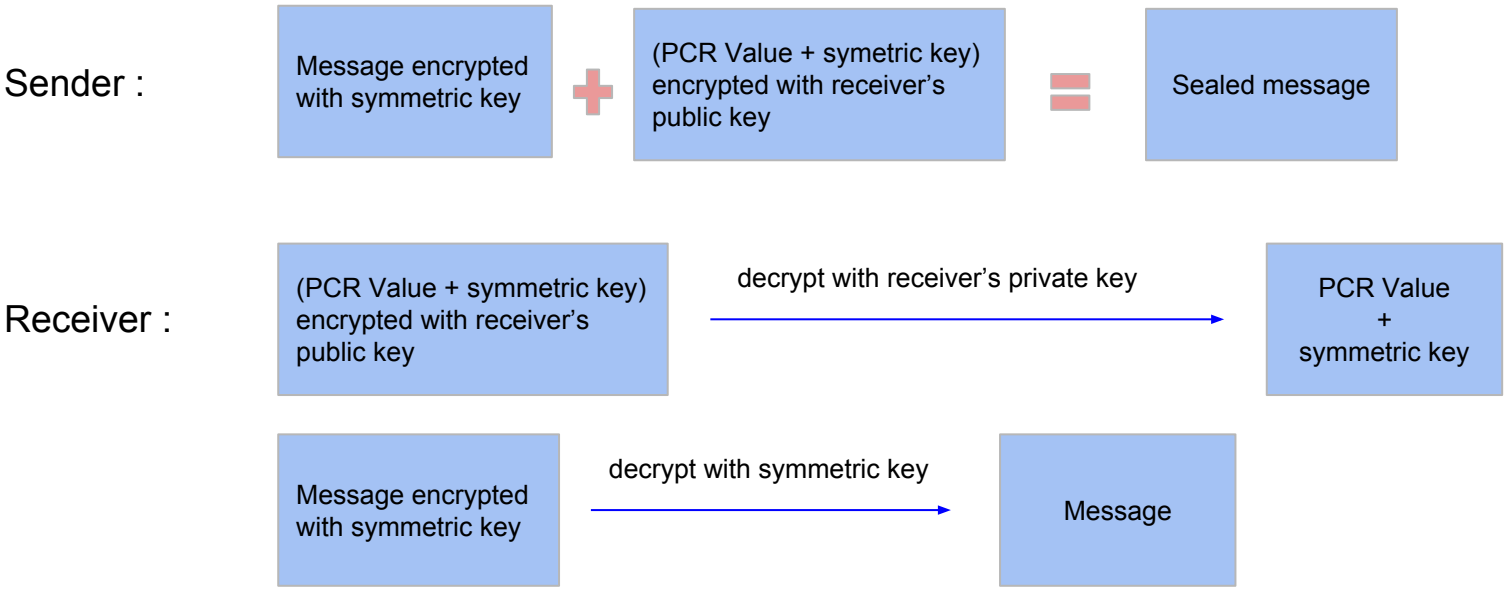
Figure 4:e - Diagram of Credentials and their Relationships.

TPM as an Endpoint of Communication

- Binding
  - Binding is the traditional operation of encrypting a message using a public key. That is, the sender uses the public key of the intended recipient to encrypt the message. The message is only recoverable by decryption using the recipient's private key.
  - Hence, a message encrypted with the public key, "bound" to a particular instance of a TPM.
- Signing
  - Signing also in the traditional sense, associates the integrity of a message with the key used to generate the signature.
  - The TPM tags some managed keys as signing only keys, meaning these keys are only used to compute a hash of the signed data and encrypt the hash.

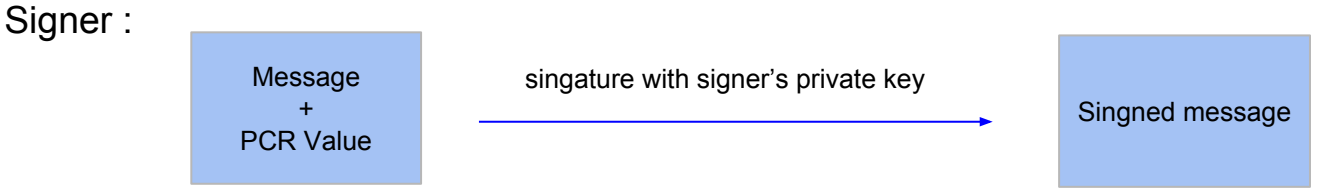
TPM as an Endpoint of Communication

- Sealing
  - Sealed messages are bound to a set of platform metrics specified by the message sender.



TPM as an Endpoint of Communication

- Sealed-Signing
  - Signing operations can also be linked to PCR registers as a way of increasing the assurance that the platform that signed the message meets a specific configuration requirement.



Interfacing with TPM and Software Services

There are three interfaces envisaged for TCG software. They correspond to services layering common to most general purpose computing platforms.

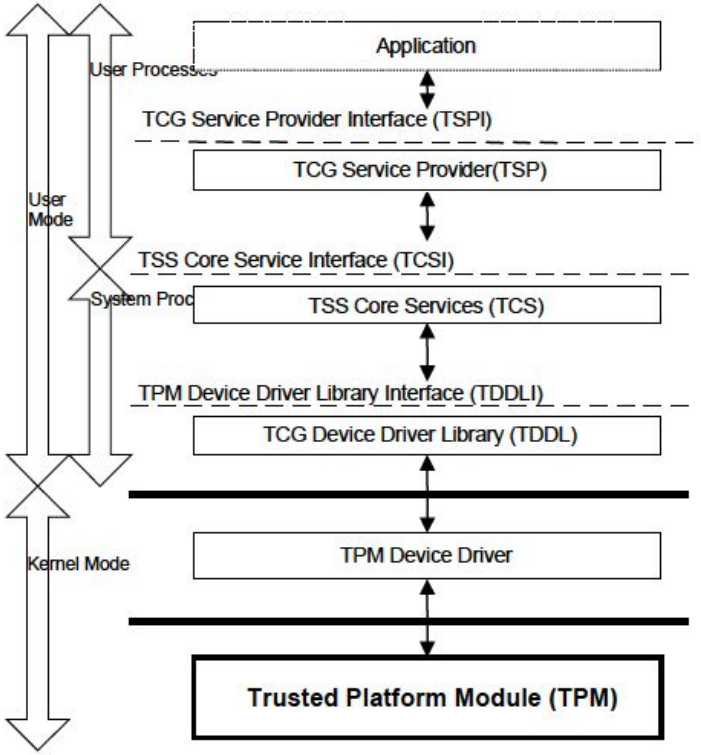


Figure 4:i – TCG Software Layering.

Interfacing with TPM and Software Services

TSP(TCG Service Provider) Interface

- The TCG Service Provider (TSP) exposes a C interface to the TPM, based on an object oriented underlying architecture.

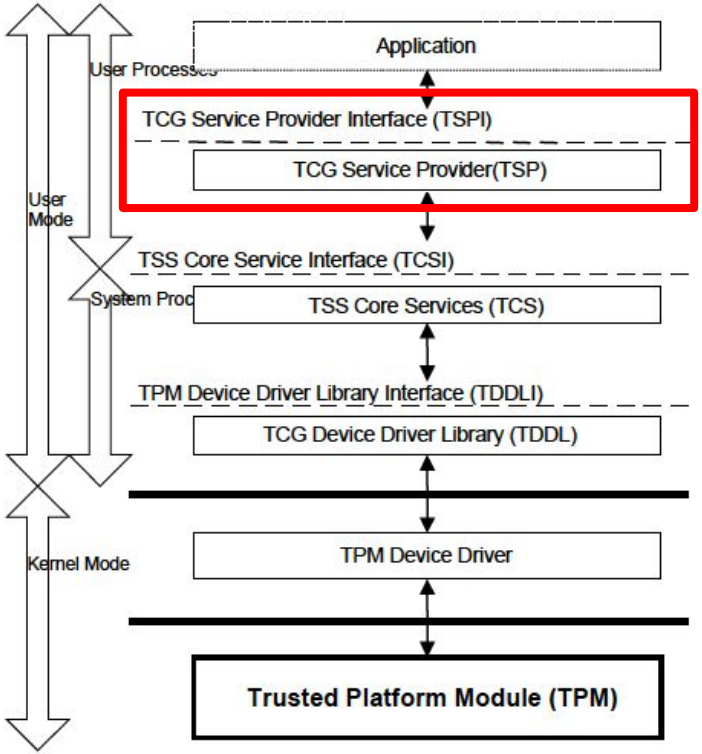


Figure 4:i – TCG Software Layering.

Interfacing with TPM and Software Services

TCS(TSS Core Service) Interface

- The TCS operates as a system process in user mode. It is trusted to manage authorization information supplied to the TPM.
- The TCG Core Services (TCS) provides an interface to a common set of platform services.

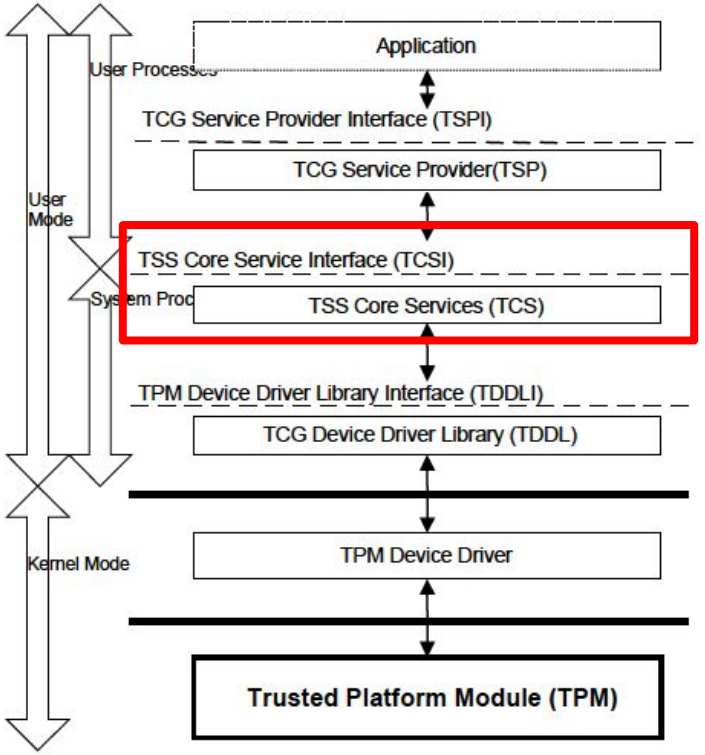


Figure 4:i – TCG Software Layering.

Interfacing with TPM and Software Services

TDDL(TCG Device Driver Library) Interface

- TDDLI is a user mode interface.
- Such an interface has several advantages over a kernel mode driver interface:
  - It ensures different implementations of the TCG software stack properly communicate with any TPM.
  - It provides an OS-independent interface for TPM applications.

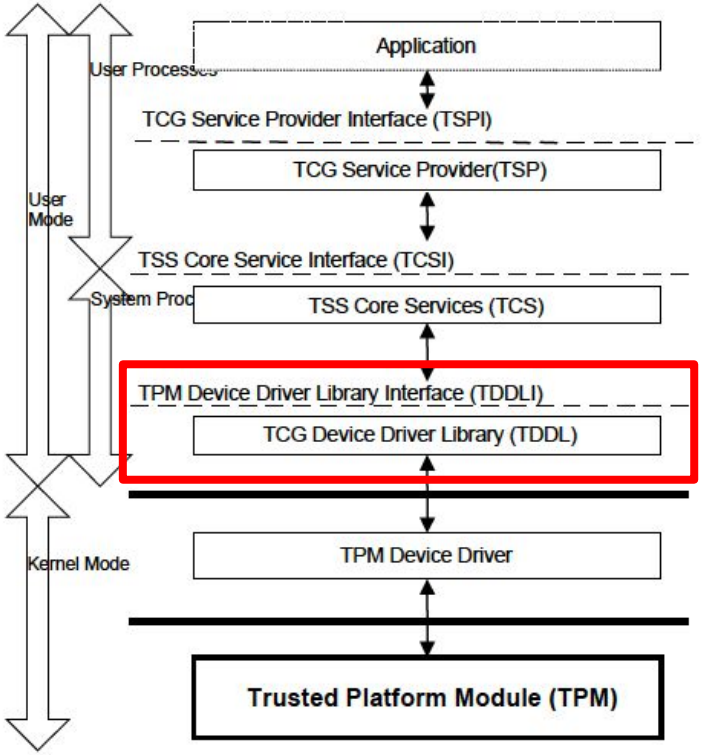
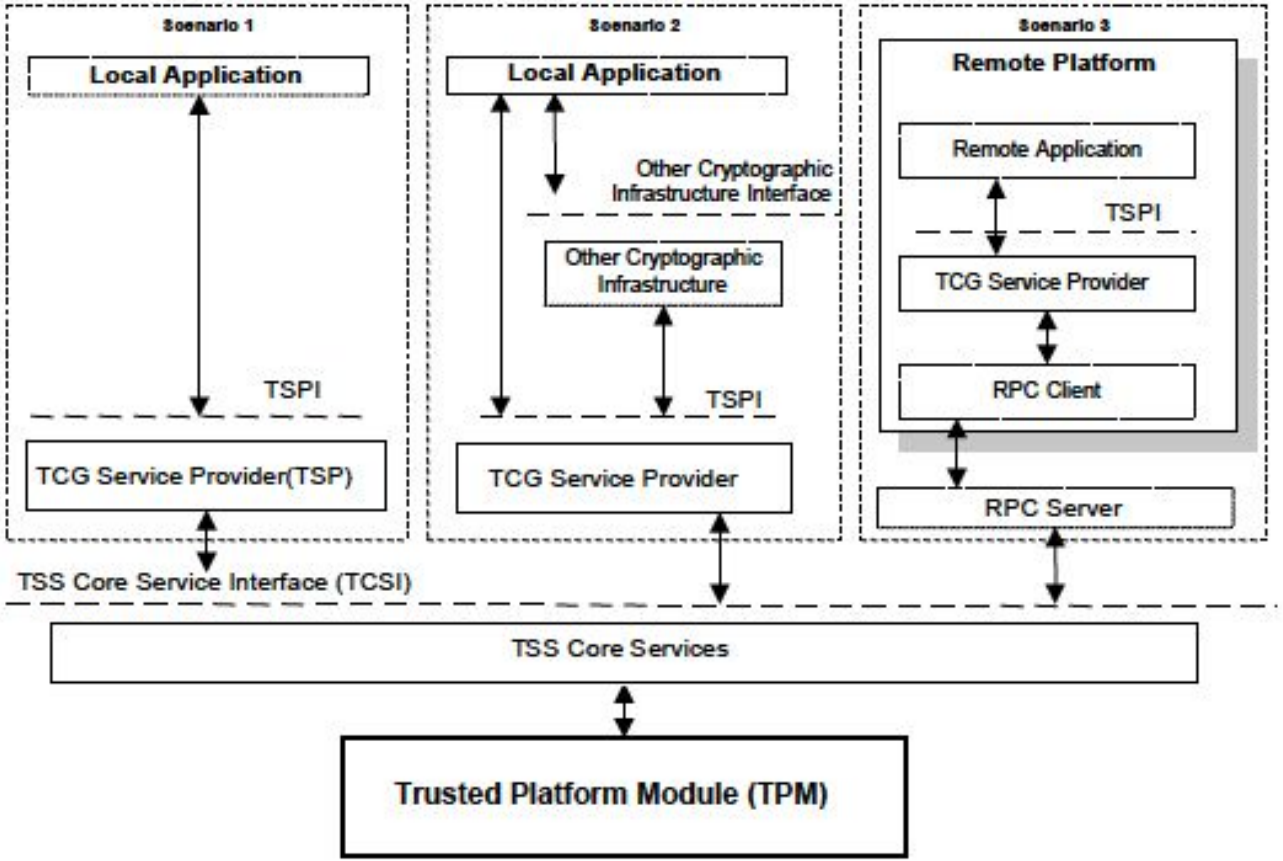


Figure 4:i – TCG Software Layering.

Interfacing with TPM and Software Services



Interfacing with TPM and Software Services

Scenario 1 - Comparing Measurement Events

- This scenario traces the calling sequence of an application verifying platform configuration contained in TPM managed PCR registers with expected values contained in Validation Credentials.
- The following steps are involved:
  1) Initialize application objects and prepare to read PCR registers.
  2) Read PCR5 value.
  3) Compare PCR value(s) with validation values.

Interfacing with TPM and Software Services

Scenario 1 - Comparing Measurement Events

- Application

```
{
Tspi_Contest_Create( &hContext);
Tspi_Context_Connect(hContext, NULL);
Tspi_Context_GetTpmObject(hContext,&hTPM);
Tspi_TPM_PcrRead(hTPM, 5, &ulPcrValueLength, &rgbPCRValue);
Compare(correctValueOfPCR5, rgbPCRValue);
}
```

Interfacing with TPM and Software Services

Scenario 1 - Comparing Measurement Events

- TSP

```
{ …
hKey = Tcsip_LoadKeyByBlob(ikey); //Load the key identified by its hash
Tcsip_Quote(hKey,…);              //Retrieve signed PCR
… }
```

Interfacing with TPM and Software Services

Scenario 1 - Comparing Measurement Events

- TCS

```
{ …
loadKeyMsg = PBG_LoadKey(hKey); //Marshall TPM_LoadKey command
quoteMsg = PBG_Quote();         //Marshall TPM_Quote command
Tddli_Open();                   //Open TPM communications channel
Tddli_TransmitData(loadKeyMsg); //Send/Recv response
Tddli_TransmitData(quoteMsg);   //Send/Recv response
Tddli_Close();                  //Optionally close channel w/ TPM
… }
```

Interfacing with TPM and Software Services

Scenario 1 - Comparing Measurement Events

- TPM

Upon receiving the messages to load a key (loadKeyMsg) and retrieve PCR values (quoteMsg), the TPM parses the command blocks sequentially and performs the appropriate operation.

Interfacing with TPM and Software Services

Scenario 2 - TPM as a Fixed Token Storage Device

- In the scenario, existing interfaces provide fixed-token / smartcard storage capabilities (e.g. PKCS1111) that may be leveraged to access the TPM device for storage / retrieve of symmetric / asymmetric keys.

Interfacing with TPM and Software Services

Scenario 3 - Reading a PCR from a Remote Platform

- The application interacts with a TPM that is remotely connected.
- The TCS implementation is built using an remote procedure call (RPC) or other messaging service.
- The remote application in this case will talk directly to the TCS.

Interfacing with TPM and Software Services

Scenario 3 - Reading a PCR from a Remote Platform

- Remote TCS

```
{ …
loadKeyMsg = PBG_LoadKey(hKey); //Marshall TPM_LoadKey command
quoteMsg = PBG_Quote();         //Marshall TPM_Quote command
RPC_Open();                     //Open TPM communications channel
RPC_Send(loadKeyMsg);           //Send/Recv response
RPC_Send(quoteMsg);             //Send/Recv response
RPC_Recv(loadKeyMsg);           //Send/Recv response
RPC_Recv(quoteMsg);             //Send/Recv response
RPC_Close();                    //Optionally close channel w/ TPM
…}
```

Interfacing with TPM and Software Services

Scenario 3 - Reading a PCR from a Remote Platform

- Local TCS

```
{ …
RPC_Recv(loadKeyMsg);                  //Recv command
RPC_Recv(quoteMsg);                    //Recv command
Tddli_Open();                          //Open TPM communications channel
Tddli_TransmitData(loadKeyMsg);         //Send/Recv response
Tddli_TransmitData(quoteMsg);          //Send/Recv response
Tddli_Close();                         //Optionally close channel w/ TPM
RPC_Send(loadKeyMsg);                  //Send reply
RPC_Send(quoteMsg);                    //Send reply
…}
```

Example - ICP(Internet Content Provider)

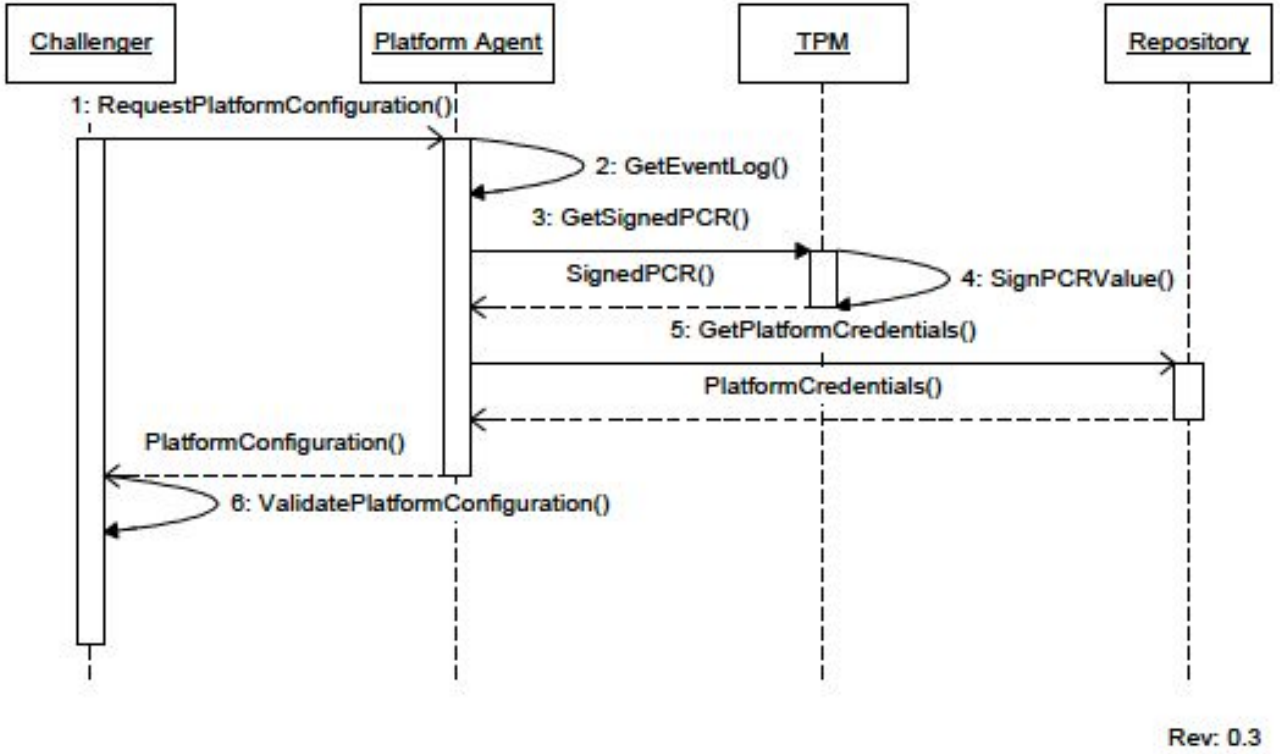


Figure 4:d - Attestation Protocol and Message Exchange

Conclusion

With the rise of IoT devices, trusting computing once again received attention...