Trusted Mobile Platforms:

Part 1: An introduction to trusted computing

Chris Mitchell
Royal Holloway, University of London
c.mitchell@rhul.ac.uk
http://www.isg.rhul.ac.uk/~cjm

Contents

- What is trusted computing?
- The TCG
- TCG – TPM and TSS
- Microsoft – NGSCB
- Microsoft – Vista
- Intel – LaGrande
- Open_TC – XEN/L4
- Software security – How can trusted computing help?

Contents

- What is trusted computing?
- The TCG
- TCG – TPM and TSS
- Microsoft – NGSCB
- Microsoft – Vista
- Intel – LaGrande
- Open_TC – XEN/L4
- Software security – How can trusted computing help?

Trusted Computing

- We start by introducing the notion of Trusted Computing.
- The notion originates from the Trusted Computing Group (TCG) – in fact from its predecessor body, the TCPA.
- The first fruits of what has been a large scale research and development effort are now visible in the form of a secure chip on the motherboards of many new PCs.
- Microsoft Vista incorporates support for these chips, and uses them as the basis for certain novel security functions.
- Open source software also exists that is capable of exploiting this hardware.
- However, the full potential of the hardware remains to be exploited.

A trusted system

- The word *trust* is used to mean many different things.
- For our purposes, a *trusted system* or component is one that behaves in the expected manner for a particular purpose.

  [*Trusted Computing Group*: www.trustedcomputinggroup.org]
- Trust by this definition is not the same as *secure*.
- Code known to be malicious, e.g. a virus, is trusted in the sense that it will do bad things, as expected!

Trust in a PC

- Trusted operation is difficult to achieve for a PC – where typically there is no way of telling whether the 'real' (uncorrupted) Windows is running.
- As a result there is no way of getting confidence in the correct running of applications. [Even if the operating system says that everything is OK, then this does not help because it cannot be believed].
- It is even more difficult to prove to a third party that the state of a PC is as claimed.

Fundamental requirements

- To achieve trust we first need to have a way of achieving assurance that the operating system has booted correctly.
- This requires assuming that the PC hardware has not been modified; this is made difficult, but not impossible, for the attacker by embedding key functions in a dedicated chip – the *Trusted Platform Module* (TPM).
- We need a way of monitoring the boot process.
- The component that measures the initial boot must be trusted – the '*Core Root of Trust*' – this is hardware-based.
- If the loaded software has been measured (and hence is reliable), it can measure the next software to be loaded, and again there is a solid basis for trust; this process is iterated.

Storing measurements

- As well as performing measurements during the boot process, there needs to be a reliable way of recording the results of each of these measurements.
- The trusted hardware incorporates hardware registers which store hash-codes of software that has been loaded – these registers provide a reliable record of all the software that has been executed on the trusted platform.
- Anyone wishing to check the state of the platform only needs to be given the contents of these registers (as long as they know what the values 'ought to be').

Building on the trusted base

- This base of trust can be used to support two fundamental trusted computing functions:
  - ***Attestation***, where a PC can reliably attest to its software state to a third party (by describing the contents of the registers which store hashes of software state);
  - ***Secure storage***, where a PC can store data in such a way that only if the PC is in a specific trusted state will the data be decrypted and available to an application (by linking the decryption keys to specific register contents).
- We now look in a little more detail at the set of technical functions provided by trusted computing (as needed to support the fundamentals we have outlined).

Components of a trusted computing framework I

- Shielded locations and protected capabilities:
  - *Protected capabilities* are those capabilities whose correct operation is necessary for the platform to be trusted.
  - *Shielded locations* are areas in which data is protected against interference or snooping.
  - Only protected capabilities have access to shielded locations.
- Attestation:
  - Attestation by the TPM;
  - Attestation to a trusted platform (incorporating a TPM);
  - Attestation of a trusted platform;
  - Authentication of a trusted platform.
- Integrity measurement, storage and reporting.

[TCG specification Architecture Overview]

Components of a trusted computing framework II

Microsoft's additional components:

- Process isolation, where an integrated *isolation kernel* supports the execution of several compartments/domains in parallel on the same machine, and controls the access of applications/OSs running in these compartments to system resources.
- A secure path from the peripherals to trusted applications.

[Microsoft Security Model for NGSCB]

Components of a trusted computing framework III

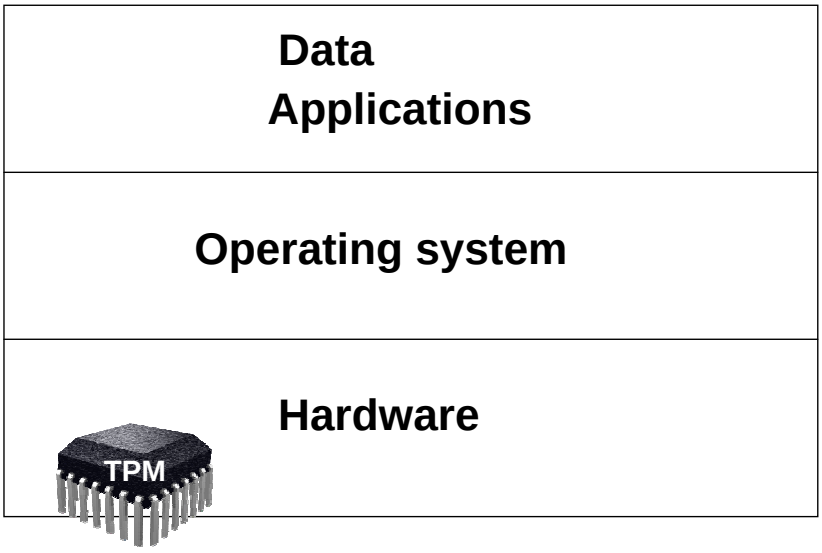
- Confidentiality and integrity protection of application code and data during execution.
- Confidentiality and integrity protection of application code and data during storage.
- Integrity protection of the operating system and underlying hardware so that the above properties can be satisfied.
- Platform attestation.
- A trusted path to the user so that the confidentiality of user input can be assured.
- Secure channels to devices and between applications to ensure the confidentiality, integrity, and authenticity of communicated data.
- Reliability assurance, necessitating size restrictions on trusted critical components.

[Sadeghi and Stüble: Bridging the Gap between TCPA/Palladium and Personal Security]

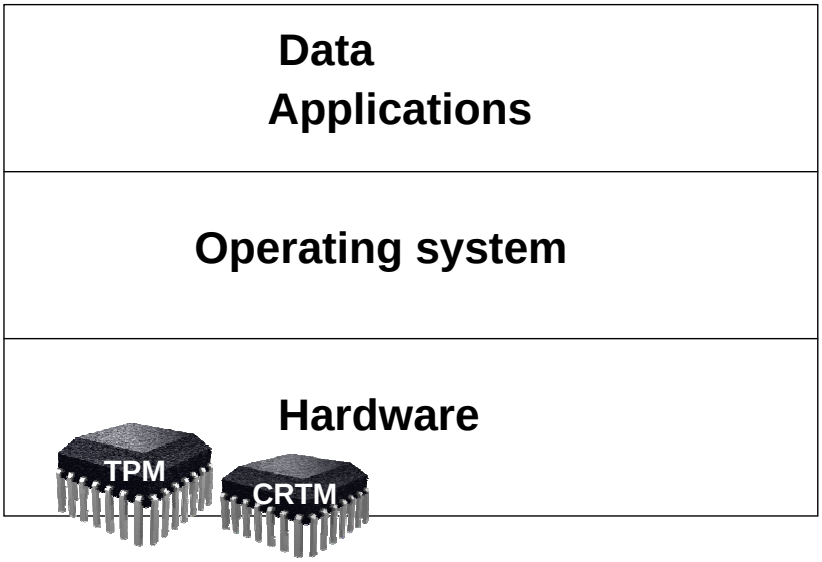
Components of a trusted computing framework IV (summary)

- ***Attestation*** – provides remote assurance of the state of the hardware and software stack running on a computer.
- ***Isolation*** – execution environments/domains/compartments.
- ***Secure storage***:
  - Encryption;
  - Sealing (binding of data to specific machine state).
- ***Secure I/O***.

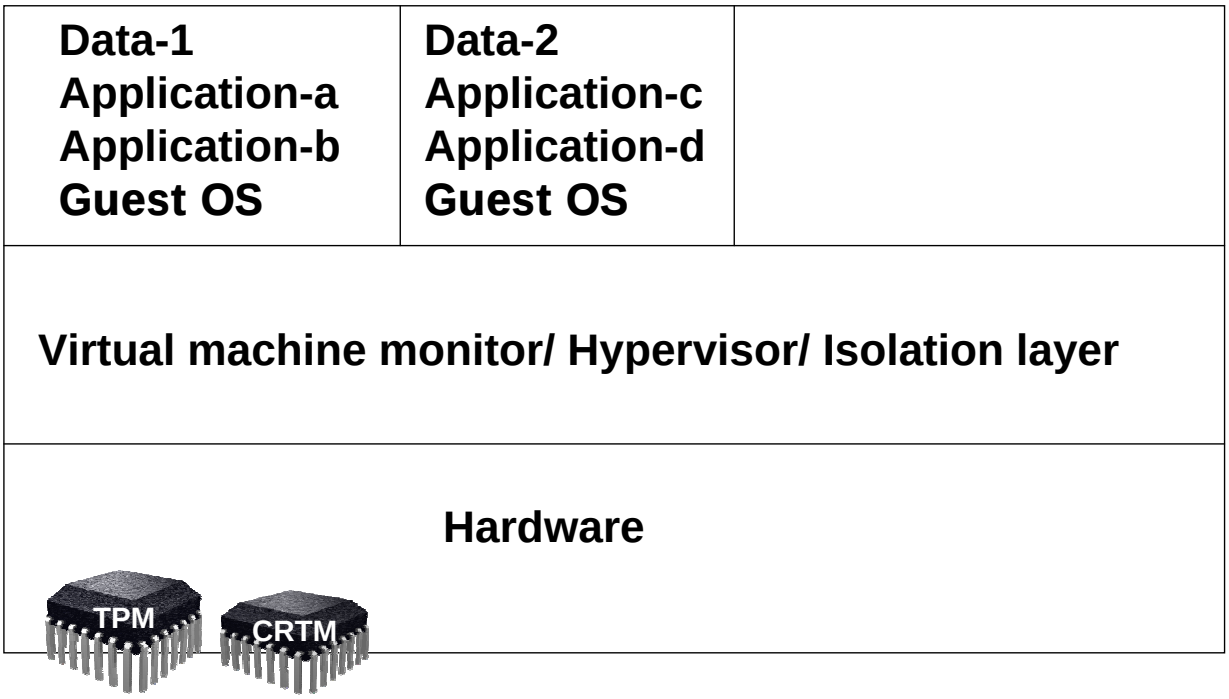
Current platforms with integrated TPMs



Envisaged trusted platforms (stage 1)



Envisaged trusted platforms (stage 2)



Envisaged trusted platforms (stage 3)

Data-1 **Application-a** **Application-b** **Guest OS**	**Data-2** **Application-c** **Application-d** **Guest OS**	
Virtual machine monitor/ Hypervisor/ Isolation layer		
Hardware (including hardware support for isolation – CPU, chipset, keyboard, mouse, video graphics card extensions) 		

Contents

- What is trusted computing?
- The TCG
- TCG – TPM and TSS
- Microsoft – NGSCB
- Microsoft – Vista
- Intel – LaGrande
- Open_TC – XEN/L4
- Software security – How can trusted computing help?

Trusted Computing Group (TCG)

- We next consider the body responsible for the trusted computing specifications.
- Whilst some of the key ideas of TCPA/TCG had existed before the TCPA was formed, the impetus for the development of a comprehensive set of specifications has been provided by these industry consortia.

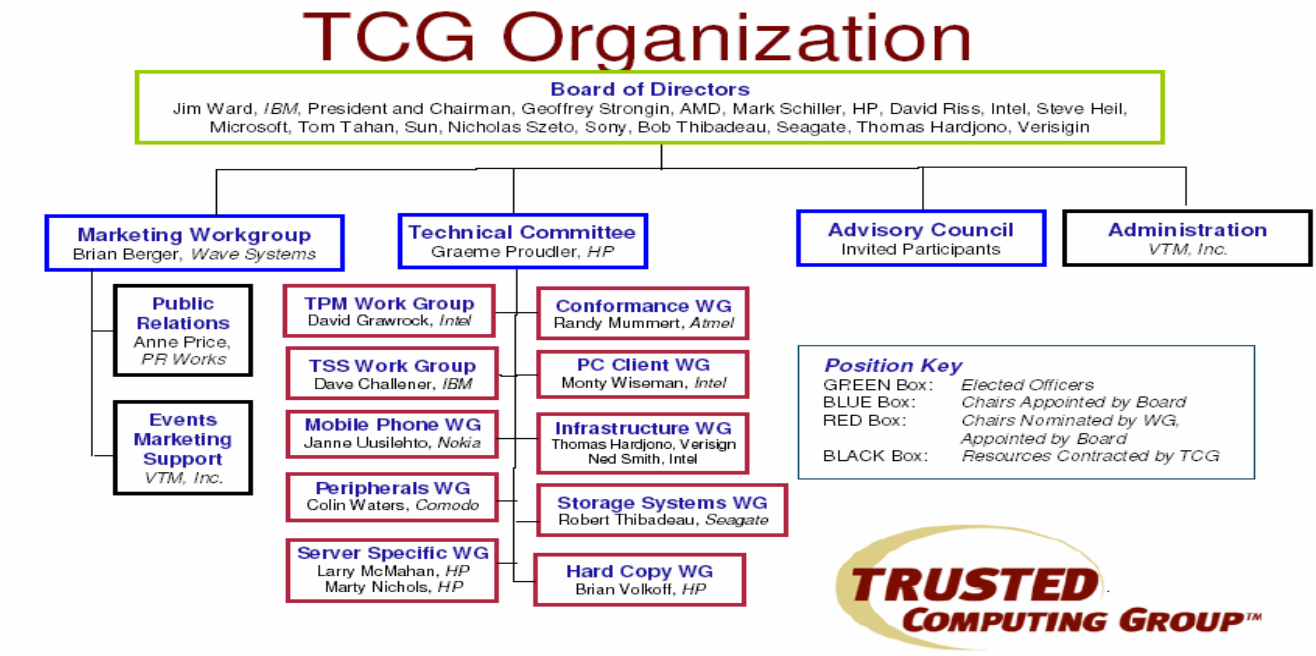
The TCPA

- The TCPA (Trusted Computing Platform Alliance) was an industry working group with the goal of enhancing trust and security in computing platforms.
- Originally an alliance of promoter companies (HP, IBM, Intel and Microsoft). Founded in 1999.
- Initial draft standard unveiled in late 1999.
- Invitation then extended to other companies to join the alliance.
- Specification eventually became an open industry standard.
- By 2002 the TCPA had over 150 member companies.

The TCG

- TCG: announced April 8, 2003.
- TCPA recognised TCG as successor organisation for the development of trusted computing specifications.
- TCG adopted the specifications of the TCPA.
- Aim:
  - To extend the specifications for multiple platform types;
  - To complete software interface specifications to facilitate application development and interoperability;
  - To ensure backward compatibility.

The TCG



The TCG main specifications

- TCG TPM main specification (general platform specification) version 1.2:
  - Design principles;
  - Structures of the TPM;
  - TPM commands.
- TCG software stack (TSS) specification version 1.2.
- TCG software stack (TSS) specification header file.
- All the specifications are available at:

  www.trustedcomputinggroup.org

Contents

TCG – main components

- We next briefly review the main components specified by the TCG.
- These make up what is known as the Trusted Platform Subsystem (TPS).
- The TPS is a combination of hardware enhancements to a PC, and software that makes it possible to use the functionality of the hardware.

Trusted Platform Subsystem (TPS)

The TPS is composed of three fundamental elements:

- The root of trust for measurement (RTM);
- The trusted platform module (TPM), which incorporates the root of trust for storage (RTS), and the root of trust for reporting (RTR); and
- The TCG software stack (TSS), which encompasses the software on the platform that supports the platform's TPM.

Roots of trust: RTM I

- The RTM is a computing engine which generates integrity measurements of software components running on the platform.
- The measurement (a hash digest) is then recorded to a platform configuration register (PCR) in the TPM.
- Details of the software component that has been measured are then recorded to the stored measurement log (SML), held outside the TPM.

Roots of trust: RTM II

- For the foreseeable future, it is envisaged that the RTM will be provided by the normal computing engine of the platform, where special BIOS boot block or BIOS instructions (the CRTM) cause the main platform processor to function as the RTM.
- Ideally, however, for the highest level of security, the CRTM would be part of the TPM.

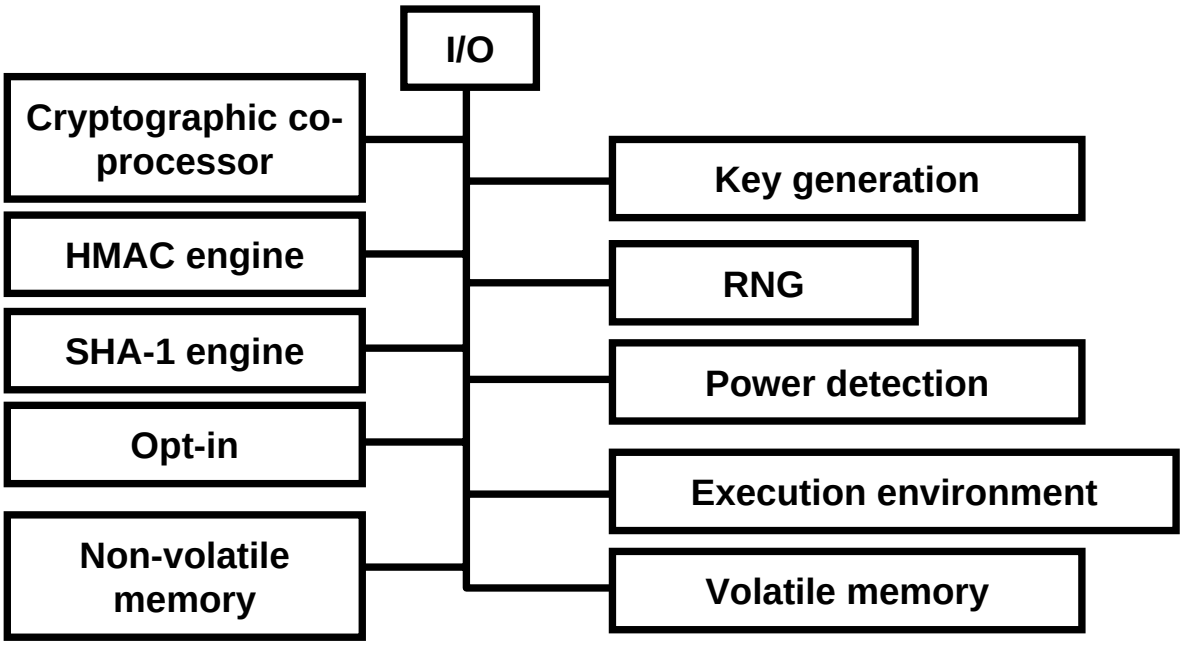
Roots of trust: RTS/RTR

- The RTS is a collection of capabilities which must be trusted if storage of data inside a platform is to be trusted.
  - The RTS provides integrity and confidentiality protection to the data used by the TPM, but that is stored externally (in the SML);
  - It also provides a mechanism to ensure that, if required, the release of specific data only occurs in a named environment.
- The RTR is a collection of capabilities that must be trusted if reports of integrity measurements (which represent the platform state) are to be trusted.

The TSS

- The TCG software stack (TSS) is software (running on the host platform) which supports use of the TPM.
- The TSS architecture consists of a number of software modules, which provide fundamental resources to support access to the TPM:
  - The TPM Device Driver;
  - TPM Core Services;
  - TPM Service Provider.

The TPM components



TPM functional components

- A TPM incorporates the following functionality:
  - Key generation, including the generation of RSA key pairs, secret keys, and random nonces.
  - Cryptographic co-processor, providing:
    - RSA encryption and signing;
    - Symmetric encryption.
  - Program execution.
  - HMAC computation.
  - SHA-1 computation.
  - Power detection.
  - Random number generation.
  - Non-volatile and volatile memory.
  - Platform Configuration Registers (PCRs).

Cryptographic aside

- The cryptographic functions are fixed ('hard coded') in the v1.2 TPM specifications.
- This has recently caused major problems, with the discovery of weaknesses in the design of SHA-1, since SHA-1 is one of the functions built into the v1.2 TPM specifications.
- SHA-1 now looks set to be phased out by NIST over the next few years.
- There will thus be a need for a v1.3 TPM specification in the next couple of years, which looks likely to use crypto in a more flexible way (e.g. with algorithm identifiers, as in X.509, instead of fixed algorithms).

Entities in the TCG model

- The **TPM owner** is in complete control of a trusted platform's (TP's) TPM:
  - Some commands are *Owner authorised* (they can only be executed by owner).
- **TPM user** (who may be different to the TPM owner).
- **Challenger** (who wishes to verify the platform state).
- **Protected object owner** (i.e. the owner of data and/or software on a platform, which may be distinct from the TPM owner and TPM user).
- **Intermediaries** – used to support migration.

Trusted Third Parties

- The TCG system relies on a number of Trusted Third Parties (TTPs), typically to issue signed certificates asserting certain properties of hardware or software.
- We refer to these as **Certification Entities**.
- A Trusted Platform should be shipped with a number of certificates created by these entities.

Certification entities I

- A **Trusted Platform Module Entity** (TPME) asserts that the TPM is genuine by signing an **endorsement credential** containing the public endorsement key for that TPM. The TPME is likely to be the TPM manufacturer.
- A **Conformance Entity** (CE) signs a **conformance credential** to assert that the design and implementation of the TPM and the trusted building blocks (TBB) within a trusted platform meet established evaluation guidelines.
- A **Platform Entity** (PE) signs a **platform credential** to assert that a particular platform conforms to a TP design, as described in conformance credentials, and that the platform's TPM is genuine.
- In the future, it is planned that every trusted platform will be shipped with an endorsement credential, one or more conformance credentials, and a platform credential.

Certification entities II

- Two other types of certification entity are defined:
  - A ***validation entity*** (VE) signs **validation certificates**; these contain integrity measurements, i.e. measured values and measurement digests corresponding to correctly functioning or trustworthy platform components, for example embedded data or program code.
  - A ***Privacy-CA*** creates a certificate to assert that an identity (and an attestation identity public key) belong to a trusted platform.

TCG keys

- To perform the tasks expected of it, a TPM uses a range of different types of key, including secret keys and key pairs for asymmetric algorithms.
- These key types include:
  - **Endorsement Key** (an asymmetric encryption key pair, unique per TPM, and typically generated at time of manufacture);
  - **Attestation Identity Keys** (signature key pairs, generated by the TPM during use – a TPM may have many);
  - **Storage Root Key** (a single asymmetric encryption key pair used to support secure storage of data external to the TPM).

Endorsement Key Pair (EK)

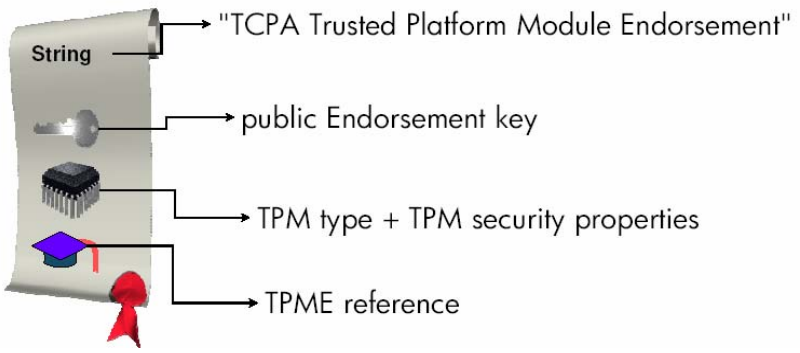
- It is a fundamental requirement that:
  - Each TPM has an endorsement key pair stored in it;
  - The public part of the endorsement key pair is certified by the TPME (e.g. the manufacturer) in the form of the endorsement credential.
- The private part of the EK is used by a TPM to prove that it is a genuine TPM. It is never used for signing.
- It is only ever used for decryption in two scenarios:
  - To take ownership of a TPM;
  - To get a public key certificate for a platform attestation identity public key (a 'platform identity').

Platform Credentials

- Prior to use, a trusted platform (and the TPM within the platform) are equipped with a set of signed certificates – generated by some of the TTPs referred to earlier.
- These certificates bind the public part of the EK to the platform, and also assert to properties of the platform.
- We refer to these certificates as the **Platform Credentials**.

Credentials I

- An **Endorsement credential**:
  - Certifies that a public encryption key (the public endorsement key) belongs to a genuine TPM;
  - Constructed by a Trusted Platform Management Entity.

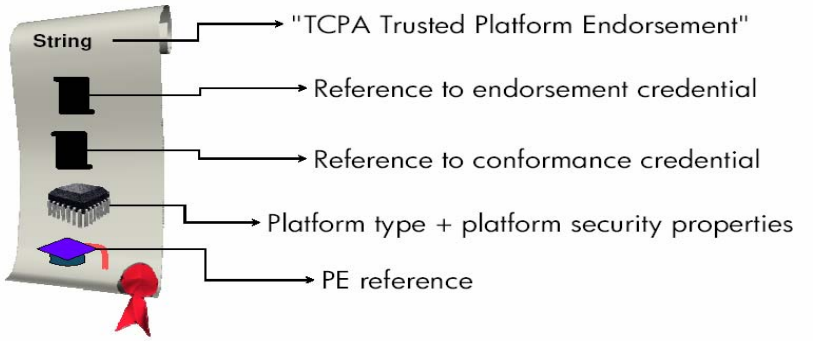


Credentials II

- A **Conformance credential** is:
  - a document that vouches that the design and implementation of the TPM and the trusted building blocks (TBB) within a trusted platform meet established evaluation guidelines;
  - signed by a Conformance Entity.

Credentials III

- A **Platform credential**:
  - is a document that proves that a TPM has been correctly incorporated into a design which conforms to the specifications;
  - proves the trusted platform is genuine
  - Is signed by a Platform Entity



Attestation Identity Key Pairs (AIKs)

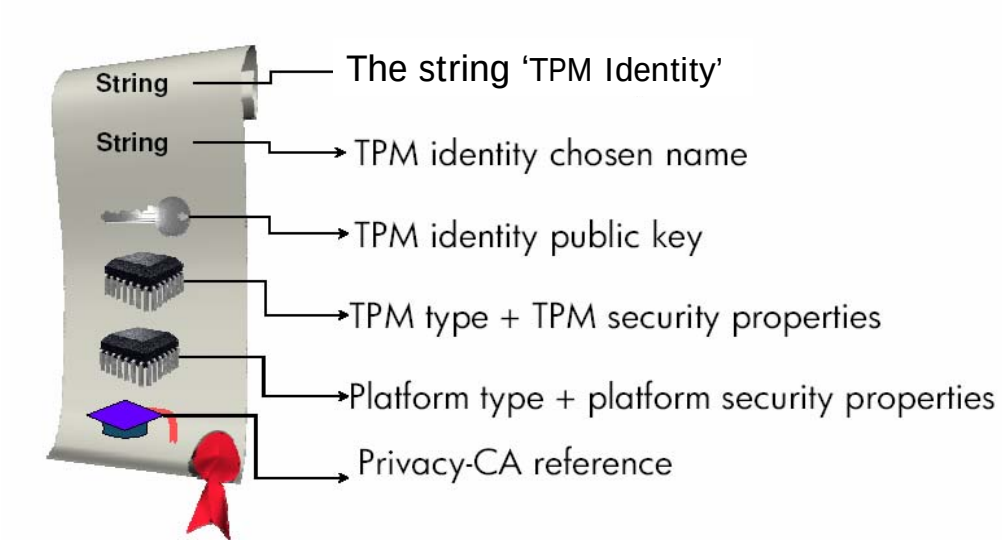
- These signature key pairs are used by a TPM to attest to platform properties to external entities.
- Used by a ‘challenger’ of the platform to verify that a TPM is indeed genuine, without identifying a specific TPM.
- A special trusted third party called a Privacy-Certification Authority (P-CA) supports the use of AIKs.

Generation of AIKs

- TPM generates a new AIK pair, chooses an 'identity', and selects a P-CA which will be asked to attest to this new identity.
- The TPM signs the AIK public key, the chosen identity, and the identifier of the chosen P-CA, using the newly generated AIK private key.
- The AIK public key, identity, signature and TPM credentials are all encrypted using the P-CA public key and sent to the P-CA.
- The P-CA decrypts the data, and then verifies the credentials and the signature.
- If all the checks succeed, the P-CA generates the **Platform Identity Certificate**, a statement that the AIK public key and the identity belong to a genuine trusted platform with the specified properties.

Platform identity certificate

- A Platform identity certificate (as generated by a P-CA) has the following content):



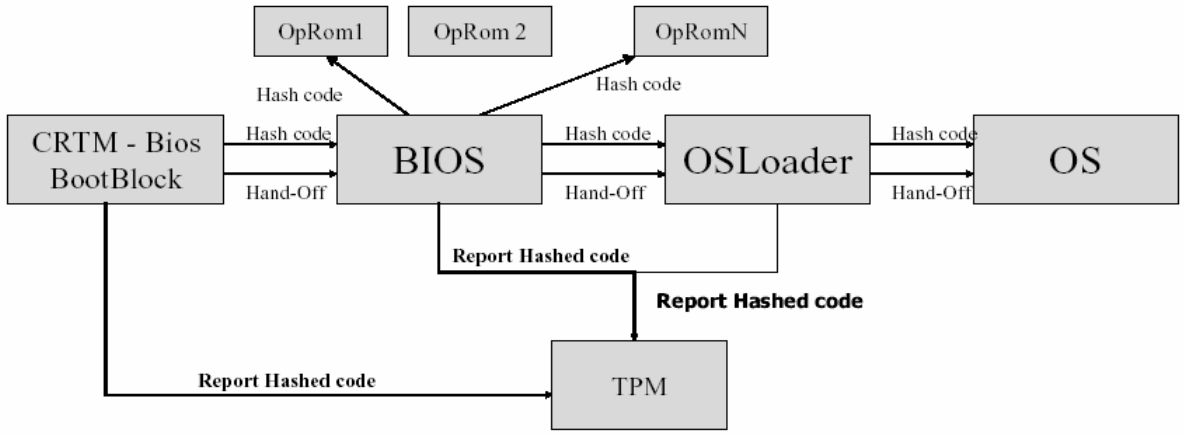
Sending the platform identity certificate to the TPM

- The P-CA generates a random secret encryption key.
- The platform identity certificate is encrypted using this secret key.
- The secret key is encrypted using the TPM's EK.
- The encrypted certificate and encrypted secret key are then sent back to the requester, thus ensuring that only the appropriate TPM can access the certificate.

Issues with use of a P-CA

- The P-CA gets to see all the platform credentials, including the endorsement credential (and the public part of the EK).
- A TPM has only one EK, and hence the P-CA can link the AIK (and associated identity) with a unique trusted platform.
- Hence, although a TPM can have many AIKs/identities, and hence a degree of anonymity/pseudonymity, this depends on the honesty of the P-CA, i.e. the P-CA can compromise this anonymity.

Authenticated boot I



Authenticated boot II

- A TPM incorporates a set of Platform Configuration Registers (PCRs).
  - They are used to store platform software integrity metrics.
  - A TPM has several PCRs (a minimum of sixteen) and uses them to record different aspects of the state of the trusted platform.
  - Each PCR has a length equal to a SHA-1 digest, i.e. 20 bytes.

Authenticated boot III

- Each PCR holds a value representing a summary of all the measurements presented to it since system boot:
  - This is less expensive than holding all the individual measurements in the TPM;
  - This means that an unlimited number of results can be stored.
- A PCR value is defined as:
  - SHA-1(existing PCR value || latest measurement result).
- A PCR must be a TPM shielded location, protected from interference and prying.
  - The fewer sequences/PCRs there are, the more difficult it is to determine the meaning of the sequence;
  - The more sequences/PCRs there are, the more costly it is to store sequences in the TPM.

Reporting on integrity

- Measurements reported to the TPM during or after the boot process cannot be removed or deleted until reboot.
- The attestation identity keys are used to sign integrity reports.
- The recipient can then evaluate the trustworthiness of the:
  - signed integrity measurements, by examining the platform identity certificate;
  - software configuration of the platform, using the reported measurements.

Authenticated versus secure boot

- The above measures provide **authenticated boot**, i.e. a means by which a third party can verify that a certain set of software has booted.
- They do not guarantee **secure boot**, i.e. guarantee that only a particular set of software is able to boot.

Secure boot – possible solutions I

- The DIR (Data Integrity Register) is a TCG v1.1 function.
- It provides a place to store information using the TPM's NV (non-volatile) storage.
- Use of the DIR is deprecated in the v1.2 specifications.
- The TPM must still support the functionality of the DIR register in the NV storage area.

Secure boot – possible solutions II

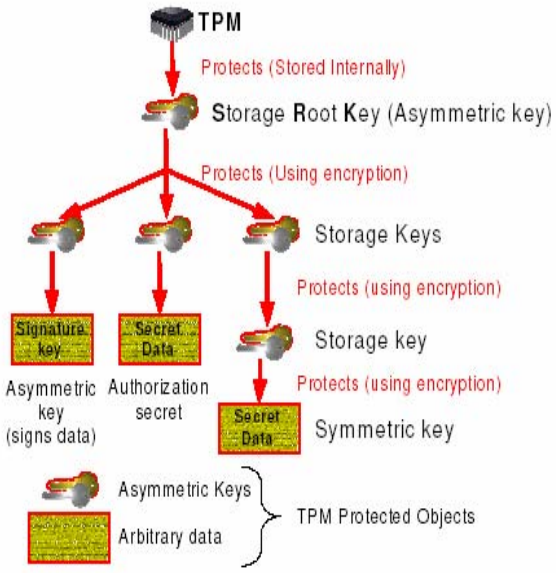
- The TPM has the same number of DIRs as PCRs.
- The expected PCR values can be written by the TPM owner to the corresponding DIRs.
- During boot, the CRTM and the measurement agents measure the software components on the platform.
- Every time a final PCR value is computed, the PCR value is compared to the corresponding DIR value.
- If the two values match, control is passed to the next software component, and the boot process continues; otherwise an exception is called and the boot process is halted.

Secure boot – possible solutions III

- Alternatively, if the TPM has access to non-volatile memory, all expected PCR values can be held in unprotected non-volatile memory and their summary (cumulative digest) held in a single DIR.
- When a PCR value has been calculated, the RTM or measurement agent checks that:
  - the cumulative digest of the expected table of PCR values matches that held in the DIR; and
  - the calculated PCR value then matches its expected value in the table.

Secure storage I

- Each trusted platform contains a key hierarchy.
- At the root is the storage root key, SRK, stored securely in the TPM.
- Data or keys can be encrypted in such a way that they can only be decrypted by the TPM.
- Asymmetric encryption is used.



Secure storage II

- Binding (data):
  - This TPM capability allows external data to be encrypted using a public TPM parent key such that it can only be decrypted by the TPM.
- Wrapping (keys):
  - **TSS Wrap Key**: This TPM capability allows an externally generated key to be encrypted using a parent key.
- Wrapping variants:
  - **TSS Wrap key to PCR**: Similar to **TSS Wrap Key** but the externally generated key is 'wrapped to' PCR values;
  - **TPM Create wrap key**: Creates a TPM key, which may or may not be locked to PCRs.

Secure storage III (sealing)

- **Sealing** (data / secret keys):
  - This is an important aspect of protected storage.
  - The seal operation can bind a secret to an individual TPM.
  - External data is concatenated with the value of an integrity metric sequence at the time the seal operation is performed, and encrypted using the public key of a parent key pair.
  - It provides the capability to store a secret such that it can only be revealed by the TPM when the platform is in an specified software state.
  - The caller of the seal operation may choose not to wrap the secret to any PCR values.

Demonstrating privilege (access control)

- TPM access control functions support:
  - Owner authorised commands;
  - Protected objects;
  - Before a TPM is owned, the TPM is unavailable
- Owner control is based on 'Cryptographic authorisation':
  - 20 bytes, for example a hashed password, or 20 bytes from a smartcard submitted to a hash algorithm, may be used;
  - Separate authorisation data must exist for the TPM owner as well as protected objects;
  - There are a number of authorisation protocols which protect against:
    - Man in the middle attacks;
    - Replay;
    - The exposure of the authorisation data.
- Physical presence:
  - Certain commands require the physical presence of a human, e.g. to push a switch.

Direct Anonymous Attestation (DAA)

- As discussed previously, the P-CA is a threat to privacy since it is capable of:
  - user/TPM activity tracking; or
  - making unwanted disclosures of platform information.
- The DAA protocol removes the necessity to disclose the public value of the endorsement key to a P-CA.
- DAA is based on a family of cryptographic techniques known as *zero knowledge proofs*.
- DAA allows a TPM to convince a remote `verifier' that it is indeed valid without disclosing the TPM public endorsement key, thereby removing the threat of a TTP collating data which may jeopardise the privacy of the TPM user.

Locality

- The functionality allows a TPM owner to assign privileges to external processes based on their locality.
- It allows the characteristics (integrity metrics) of the external software processes to be recorded in locality-specific PCRs.
- When a trusted process sends commands to the TPM:
  - A non-spoofable modifier is sent with it which indicates the locality of the process and thereby its trust value;
  - This can be used as a qualifier for more granular access to specific TPM resources.

Delegation

- This allows an owner to have fine-grained control over the use of specific owner-authorised TPM commands.
- In the v1.1b TPM specifications (which do not support this function), an owner that wishes to authorise a software module to perform an owner-authorised TPM function is required to provide the software with the TPM owner's password.
- With the delegation function provided in the v1.2 specifications, the TPM owner may delegate to a software object or other entity the ability to use any individual owner-authorised TPM command or subset of TPM commands, without granting it the ability or permission to use any other TPM commands.

Transport protection

- This is implemented to improve the security of the communication channel between the TPM and trusted processes.
- A transport session provides integrity and confidentiality protection to commands sent to the TPM:
  - integrity is provided by the use of a MAC; and
  - confidentiality is provided by the encryption of the command using a stream cipher, with keystream generated inside the TPM.
- The logging of commands sent to the TPM within a transport session is also supported.

Monotonic counters

- A monotonic counter provides an incremental value.
- The TPM is required to provide four such counters which may be implemented as:
  - Four unique counters; or
  - One counter with pointers which keep track of the other counter values.
- The internal ‘base’ – i.e. the main counter – is not directly accessible by external processes; it is used internally by the TPM.
- External counters – used by external processes – may be unique or linked to the main counter (implemented using pointers and difference values).
- To create an external counter, owner authorisation data is required.
- In order to increment an external counter, authorisation to use the counter must be passed to the TPM.

Migration (of keys)

- Non-migratable keys:
  - are locked to a particular TPM and never duplicated;
  - must be created by the TPM.
- Migratable keys:
  - can be replicated ad infinitum by its owner (who knows the migration authorisation data);
  - the extent of duplication is only known to the owner of the key;
  - can be created either outside the TPM or by the TPM;
  - no control over where the keys can be migrated to (owner’s choice).
- Certifiable migratable keys:
  - are keys created in the TPM which may be migrated but only under strict controls;
  - the destination of the key must be authorised by the TPM owner and a migration selection authority.

Time-stamping

- This functionality provides proof of a time interval not a time instance.
- It is the responsibility of the caller of the TPM capability to associate the TPM ticks (a number) to the actual UTC time.
- A sample protocol is given in the TPM specifications, demonstrating how this may be achieved.
- Use of the specified protocol is not required.

Other TPM features

- Other TPM features include:
  - TPM audit;
  - Maintenance; and
  - Context management.

Limits to TC hardware capabilities

- The notion underlying trusted computing is to reliably measure and report on the software running on a machine.
- This is fine for a simple machine, for which software will not often change (e.g. dedicated systems such as mobile phones).
- However, for a PC this is infeasible.
- The operating system alone (e.g. Windows) is incredibly large and complex, and has a very large number of versions.
- If applications are added to this, then the problem of deciding whether or not a given state is trustworthy becomes impossible.

Isolation layer

- Instead, the idea is to measure all software only up to a certain point, and then to rely on the software to ‘look after itself’.
- If the measured software provides the basis for virtualisation and secure compartments for individual processes, then we should be in good shape.
- This is the idea behind the isolation layer.
- An isolation layer is a small, secure, mini-operating system, which is measured by the trusted computing hardware, and which takes care of the security of subsequently run applications.
- Microsoft has described what its isolation layer would be like (NGSCB), and there are a variety of open source initiatives (including OpenTC).

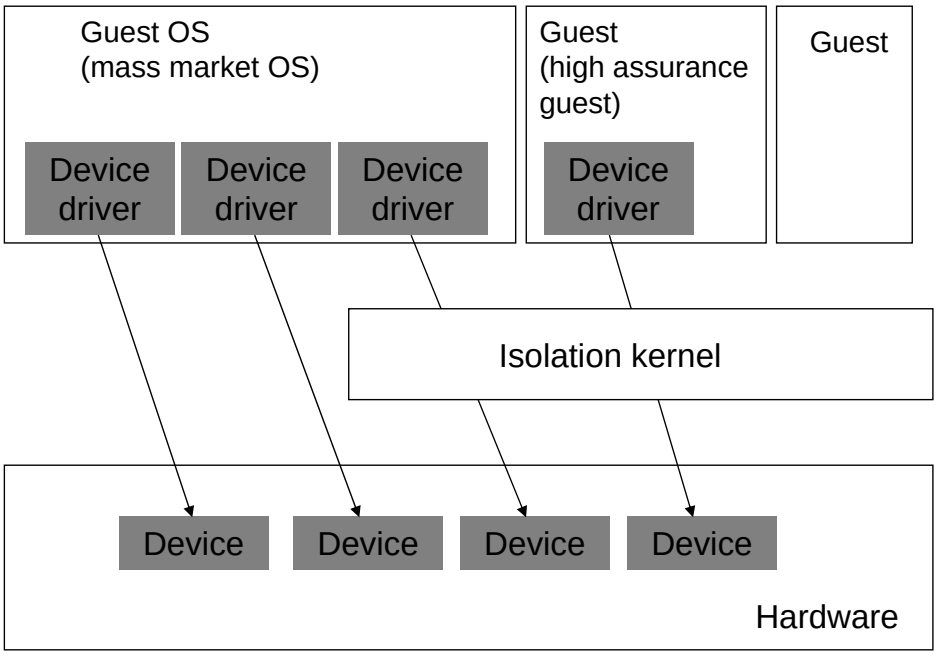
Contents

- What is trusted computing?
- The TCG
- TCG – TPM and TSS
- Microsoft – NGSCB
- Microsoft – Vista
- Intel – LaGrande
- Open_TC – XEN/L4
- Software security – How can trusted computing help?

Microsoft's Trusted Computing Initiative

- The Microsoft trusted computing initiative was originally introduced under the name Palladium.
- In January 2003 the name Palladium was dropped:
- The work continued under the name NGSCB, *for Next Generation Secure Computing Base*.

NGSCB architecture



NGSCB architecture

- The NGSCB architecture has the following components:
  - a TPM v1.2 (in NGSCB this is called a Security Support Component (SSC));
  - the isolation kernel;
  - a mass market operating system and untrusted applications (running on this OS);
  - high assurance software components.

Crypto chip/SSC

- The SSC is required to provide the following services:
  - Authenticated boot;
  - Persistent protected storage:
    - Seal/unseal;
    - Monotonic counter;
  - Attestation:
    - Quote;
    - PkSeal/PkUnseal.
- A TCG v1.2 compliant TPM provides a concrete implementation of the SSC.

Isolation layer approaches: device support

- There are two ways an isolation layer can allow guest OSs to access devices:
  - A **Virtual machine monitor** (VMM) exposes devices to guest OSs by virtualising them:
    - VMM intercepts a guest OS's attempt to access a physical device, and performs the actual device access on its behalf, with possible modifications of the request and /or access control checks;
    - VMM co-ordinates access requests from guests to share devices;
    - requires a driver for each virtualised device to be part of the isolation layer.
  - The device can be exported to a guest OS:
    - isolation layer controls which guest can access a device;
    - device accesses by guests are made directly;
    - DMA devices have unrestricted access to the full physical address space of the machine, and so a guest in control of a DMA device can circumvent isolation layer protections.

Isolation layer approaches: OS compatibility

- A VMM:
  - exposes the original hardware interface, and so supports ‘off the shelf’ OSs;
  - increases the complexity of the isolation layer, particularly on PC hardware where the x86 CPU is not virtualisable.
- Exokernels / microkernels:
  - expose different interfaces, and hence require new OSs to be written or existing OSs modified.

The NGSCB isolation kernel

- The isolation layer exposes the original hardware interface to one guest.
- The CPU has the following properties:
  - the x86 CPU has four protection rings (rings 0-3);
  - upcoming versions of x86 processors will have a new CPU mode;
  - This new mode is more privileged than the existing ring 0 (effectively ring -1);
  - the Microsoft isolation kernel will execute in this ring, and virtualisability problems will be solved.

NGSCB isolation kernel: Memory

- Virtualisation is used to partition memory among guests.
- Instructions executing on the CPU will address memory via virtual addresses.
- Each virtual address is translated to a physical address, which is then used to access physical resources.
- The page table edit control (PTEC) algorithm partitions physical memory among the guest OSs using page maps.
- Any attempt by a guest to edit its page map traps to the isolation kernel which consults its security policy, providing isolation between guests.

NGSCB isolation kernel: Devices

- In a PC, many devices are memory mapped.
- Control registers of a given device can be accessed by writing to, or reading from, certain physical addresses.
- The isolation kernel makes a device accessible to a guest by allowing a guest to map the control registers of the device into its virtual address space.
- The isolation kernel controls which guest can access the device, but contains no device drivers.

NGSCB isolation kernel: DMA devices

- In existing PC hardware, DMA devices have access to the full physical address space.
- Therefore a guest in control of a DMA device could circumvent the virtual memory protections.
- Solution: Chipset extensions:
  - store a DMA policy in main memory;
  - the policy is set by software, e.g. the isolation kernel;
  - the policy is read and enforced by hardware.

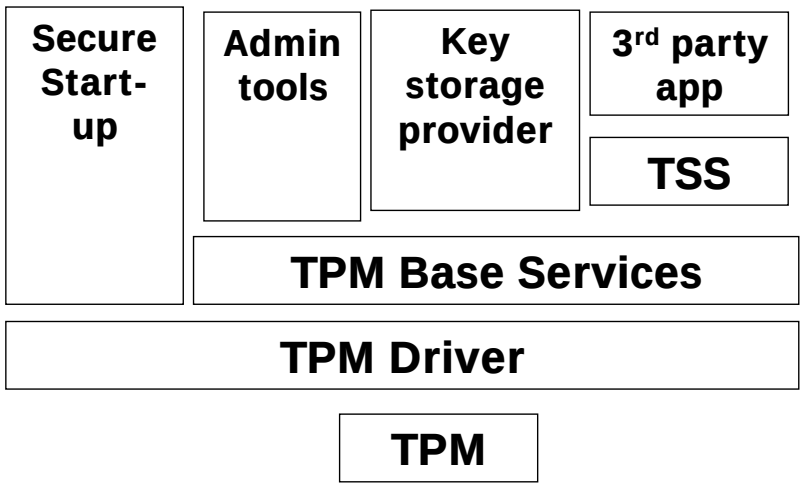
NGSCB isolation kernel: other issues

- Enhancements to input devices such as keyboards and mice may be deployed to facilitate the MACing and encryption of data as it is communicated to a trusted application on the platform.
- Secure graphics hardware may also be deployed in parallel to the complex mass-market graphics system, and used only by the isolation kernel and high assurance guests.

Contents

- What is trusted computing?
- The TCG
- TCG – TPM and TSS
- Microsoft – NGSCB
- Microsoft – Vista
- Intel – LaGrande
- Open_TC – XEN/L4
- Software security – How can trusted computing help?

Microsoft Windows Vista



Vista – TC features

- A TPM device driver designed for the TPM v1.2 chip.
- TPM base services provide sharing of limited resources on the TPM.
- The **Secure Startup**, **Admin Tools** (used, for example, to curtail use of TPM commands that may reveal privacy-sensitive information about the user or workstation), and **Key Storage Provider** components are Microsoft applications and services that rely on TPM Services.
- The “3rd-party Application” and TSS components are third-party components that rely on TPM Services:
  - No plans for v1.2 compliant TSS for Vista;
  - Microsoft say they will work with TSS vendors to create TSS products that interface with TBS infrastructure.

BitLocker Drive Encryption

- BitLocker Drive Encryption provides full volume encryption of the Windows volume, which helps protect data on a lost or stolen machine against compromise.
- In order to provide a solution that is easy to deploy and manage, a Trusted Platform Module (TPM) 1.2 chip may be used to store the keys that encrypt and decrypt the Windows volume.

BitLocker

- BitLocker also enables a key to be bound to measurements of the system volume (using ‘sealing’).
- When the computer is started, Vista verifies the system volume has not been modified in an offline attack, e.g. where an attacker boots an alternative operating system to gain control of the system.
- If the system volume has been modified, Vista alerts the user and refuses to release the key required to access protected Windows document, file, directory, and machine level data.
- The system then goes into a recovery mode, prompting the user to provide a recovery key to allow access to the Windows volume.

Contents

- What is trusted computing?
- The TCG
- TCG – TPM and TSS
- Microsoft – NGSCB
- Microsoft – Vista
- Intel – LaGrande
- Open_TC – XEN/L4
- Software security – How can trusted computing help?

LaGrande

- LaGrande is defined as “a set of enhanced hardware components designed to help protect sensitive information from software-based attacks, where LT [= LaGrande Technology] features include capabilities in the microprocessor, chipset, I/O subsystems, and other platform components”.

  [Intel LaGrande]

The architecture

- The standard partition provides an environment identical to today's Intel Architecture – 32 (IA-32) environment.
- In the standard partition, users may freely run software of their choice.
- The existence of this standard partition implies that, despite the addition of supplementary security mechanisms to the platform, code already in existence will retain its value, and software unconcerned with security will have somewhere to execute unaffected.

The architecture

- The protected partition provides a parallel environment, in which hardened software can be executed with the assurance that it cannot be tampered with by software executing in either the standard or protected partition.
- This protected partition is hardened against software attacks by the implementation of a number of components, which provide domain separation; memory protection; protected graphics; and a trusted channel to peripherals.

The architecture

- The existence of a domain manager, which facilitates this domain separation, is also assumed.
- This domain manager may be constructed in various ways, depending on the architecture implemented. A concrete example of this domain manager is the isolation kernel as described in NGSCB.
- The domain manager is physically protected via processor and chipset extensions and, in turn, protects standard and protected partitions from each other.

Hardware enhancements and extensions

- In order to facilitate the implementation of the protected partition, in conjunction with protected input and output and TPM functionality to a platform, Intel are in the process of extending and enhancing the following hardware components:
  - The CPU;
  - The memory controller or chipset;
  - The keyboard and mouse;
  - The video graphics card; and
  - The graphics adaptor.
- A v1.2 TPM must also be added.

The protected partition

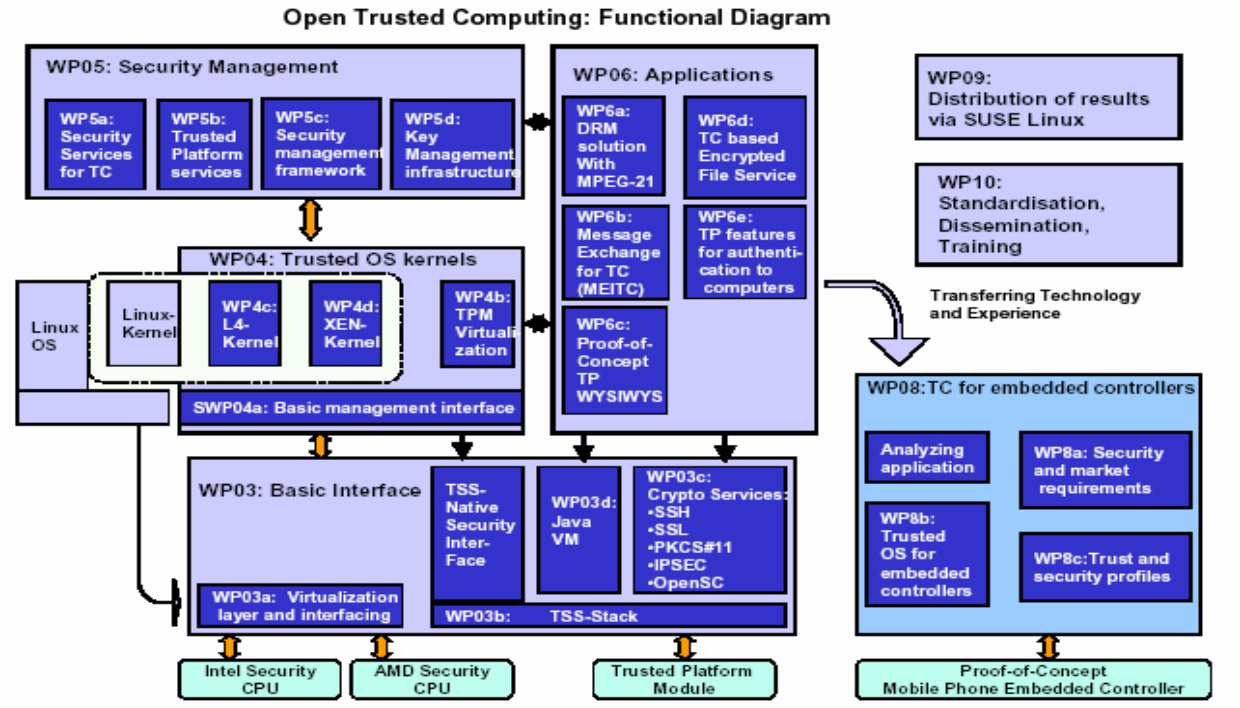
The protected partition is hardened against software attacks because:

- LT's domain separation allows hardened software to run in memory pages that are protected from viewing or modification by unauthorized applications;
- LT's memory protection prevents DMA engines from reading or modifying protected memory pages;
- LT's protected graphics processes application data from the protected partition such that it is not visible either to software in the standard partition or other software running in the unprotected partition;
- LT provides a trusted channel to keyboard and mouse that prevents keyboard snooping and/or modification of user's keystrokes or mouse movements.

Contents

- What is trusted computing?
- The TCG
- TCG – TPM and TSS
- Microsoft – NGSCB
- Microsoft – Vista
- Intel – LaGrande
- Open_TC – XEN/L4
- Software security – How can trusted computing help?

Open_TC



L4 and XEN (Open source isolation layers)

L4

- Fine grained isolation between applications;
- Minimal TCB for trusted applications / services:
  - Reuse of untrusted components via trusted wrappers:
    - Sandboxing;
    - Perimeter wrapping.
- Support for TC hardware.
- Open source alternative to Microsoft NGSCB.

  http://tudos.org/papers_ps/nizza.pdf

XEN

- Xen is a virtual machine monitor (VMM) for x86-compatible computers.

  http://www.cl.cam.ac.uk/Research/SRG/netos/xen/

Contents

Software security – What TC will not do

- Software vulnerabilities result from both:
  - design errors;
  - coding errors;
- TC technology will not prevent such errors, or aid in the development of secure software without vulnerabilities.
- Vulnerabilities can, at any time, be attacked by viruses and worms. TC will not prevent the exploitation of such vulnerabilities.
- Viruses/malicious code – TC will not stop them being written or circulated.

The isolation layer – What TC will do

- Security of the isolation layer code itself:
  - Architecture allows a relatively small isolation layer – few lines of code;
  - If it is sufficiently small then it may be possible to deploy a provably secure isolation layer;
  - Separation of the isolation layer into ring -1.
- These features help prevent a potential attack by malicious software against the isolation kernel.

The isolation layer – What TC will do

- Security of software running in protected domains supported by the isolation layer.
- Confidentiality and integrity of application code and data:
  - During execution: memory protections (which prevent software attack);
  - During storage: sealing;
  - DMA (which prevent physical attack which may allow software controls to be bypassed);
  - Inter Process Communication (IPC): a program should be able to exchange data with another program such that the integrity and confidentiality of the data is assured;
- Helps prevent a potential attack by malicious software.

Peripheral security – What TC can do

- Trusted path to the user in order to ensure the confidentiality and integrity of user input:
  - prevents malicious applications from displaying a faked dialogue, for example to enter a password;
  - prevents user input from being read/copied or altered by a malicious application:
- Secure channel to output devices to ensure integrity of output can be assured:
- Helps prevent a potential attack by malicious software.

The TPM – What TC can do

- Persistent storage:
  - Encrypted data protected from malicious code;
  - Assurance that data can only be accessed within a certain environment:
- Secure boot:
  - While not described within the TCG v1.2 specifications, all the necessary elements are in place to implement such a service;
- Attestation:
  - Enables a platform challenger to verify what versions of software are running on a platform;
  - For example, can check whether or not the latest anti-virus definitions have been downloaded:
- These all help to prevent some of the damaging effects of an attack by malicious software.

The TPM – What TC can do

- Software can be built to build on the TPM security mechanisms.
- Many software security problems arise from misuse of cryptography:
  - Misuse of randomness:
    - Many programs require sources of randomness;
    - Most common method of generating “randomness” is to use a deterministic pseudo-random generator;
    - Must be designed and implemented well – simply counting the milliseconds since midnight on the system clock is not normally good enough!
  - Poor key management:
    - Cryptographic key management is a complex issue;
    - Cannot protect long cryptographic keys with potentially weak short passwords.
  - Customised cryptography.

A crypto chip in every PC

- Putting a TPM on every PC motherboard means that every PC will have a crypto chip, with secure key storage, a random number generator, …
- Possible security applications for such a chip are almost endless.
- For example, currently there are PC crypto boards available.
- These can be used to make a PC into a secure system, e.g. to:
  - run a Certification Authority as part of a PKI;
  - to perform key management functions for a company network;
  - …
- In some cases, the TPM may be sufficiently secure to avoid the need for a separate crypto board.

Managing distributed systems

- In the long term, one of the key roles envisaged for trusted computing is to enable the secure management of distributed systems (especially in a corporate setting).
- One node in the distributed system can test the level of security offered by another node before deciding what types of task it can safely delegate to that node.
- That is, security policies can be automatically enforced.
- However, there is a long way to go …

Other applications

- A huge variety of applications have been suggested for trusted computing functionality.
- Examples include:
  - secure signature generation;
  - digital rights management (DRM);
  - secure identities for peer-to-peer computing;
  - control of personal information;
  - …
- However, what will actually happen is far from clear!

Resources

- www.trustedcomputinggroup.org
- http://www.microsoft.com/windowsvista/default.aspx
- http://www.intel.com/technology/security/
- http://www.cl.cam.ac.uk/Research/SRG/netos/xen/
- http://os.inf.tu-dresden.de/L4/LinuxOnL4/
- http://www.opentc.net/
- Trusted Computing Platforms – TCPA Technology in Context, Siani Pearson (editor), HP Invent.
- Trusted Computing – Chris Mitchell (editor), IEE.

Acknowledgements

- Must thank all members of the OpenTC project, and, in particular, the RHUL team: Eimear Gallery and Stéphane Lo Presti.
- Particular thanks to Eimear Gallery who produced the majority of the material for this talk.

Questions

- For further details on any topics addressed please contact me:
  - c.mitchell@rhul.ac.uk
  - http://www.isg.rhul.ac.uk/~cjm
  - Chris Mitchell
Information Security Group
Royal Holloway
University of London
Egham, Surrey TW20 0EX
UK

The Open-TC project is co-financed by the EC.

If you need further information, please visit our website www.opentc.net or contact the coordinator:

Technikon Forschungs- und Planungsgesellschaft mbH
Richard-Wagner-Strasse 7, 9500 Villach, AUSTRIA
Tel. +43 4242 23355 – 0
Fax. +43 4242 23355 – 77
Email coordination@opentc.net