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?
- Academic example: Secure software download
- Industry example: OMA DRM v2 implementation – MPWG

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?
- Academic example: Secure software download
- Industry example: OMA DRM v2 implementation – MPWG

A trusted system

- A trusted system or component is one that behaves in the expected manner for a particular purpose.

  [Trusted Computing Group – www.trustedcomputinggroup.org]
- This is difficult to achieve this 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 any 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

- First and foremost we 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).
- Need a way of checking the boot process.
- The component that checks the initial boot must be trusted – the 'Core Root of Trust' – this is hardware-based.
- If the loaded software has been checked (and hence is reliable), it can check the next software to be loaded, and again there is a solid basis for trust – this process is iterated.

Monitoring the checking

- As well as performing checks during the boot process, there needs to be a reliable way of recording the results of each of these checks.
- 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

- 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

Microsoft's additional components:

- Process isolation, whereby an integrated isolation kernel facilitates 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

- 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 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

- ***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***.

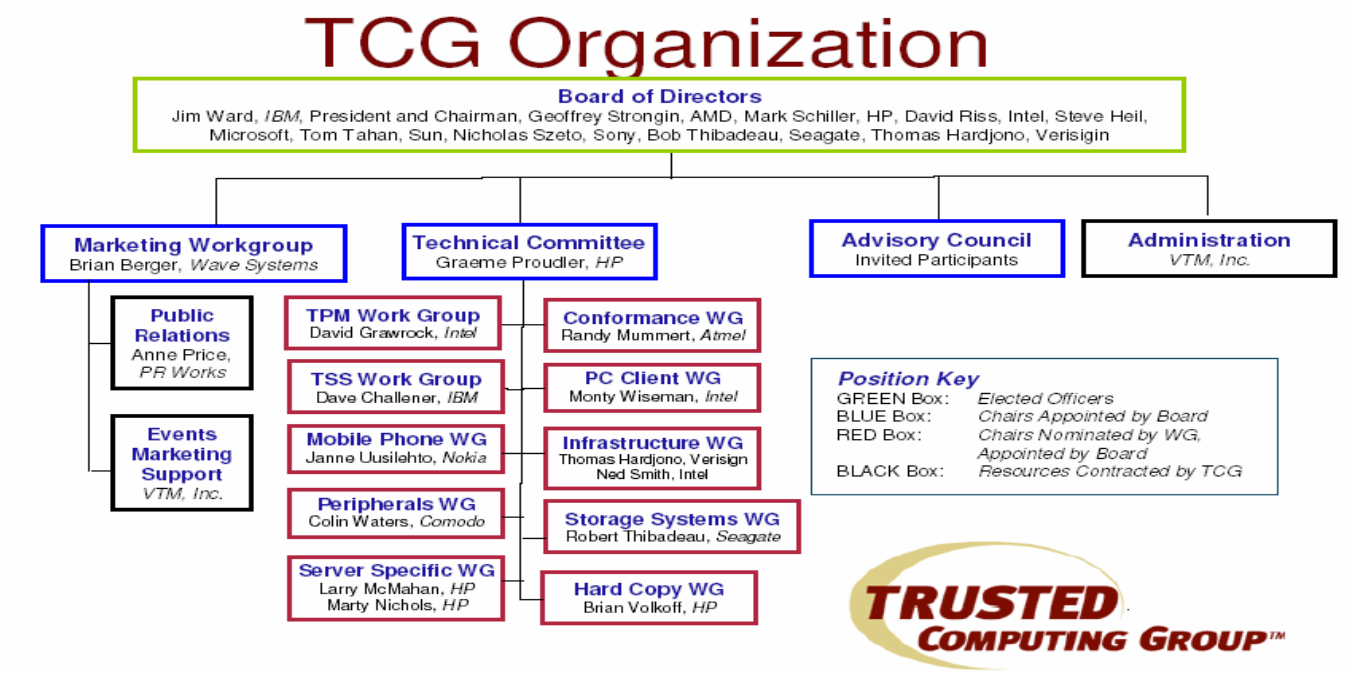
Contents

The TCPA

- TCPA (Trusted Computing Platform Alliance): An industry working group.
- Focus: 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.



- 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.

- Specifications available at:

  www.trustedcomputinggroup.org

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?
- Academic example: Secure software download
- Industry example: OMA DRM v2 implementation – MPWG

The 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

- The RTM
  - The RTM is a computing engine which accurately generates at least one integrity measurement event representing a software component running on the platform.
  - The measurement digest is then recorded to a platform configuration register (PCR) in the TPM.
  - Details of the measuring process, namely the measured value, is then recorded to the stored measurement log (SML) outside the TPM.

Roots of trust: RTM

- For the foreseeable future, it is envisaged that the RTM will be integrated into the normal computing engine of the platform, where the provision of additional 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 and 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 data used by the TPM but that is stored externally;
    - It also provides a mechanism to ensure that the release of certain 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 the software on the platform which supports the TPM.
- The challenger must determine whether TSS functions can be trusted by examining integrity metrics.

- The TSS architecture consists of a number of software modules, which provide fundamental resources to support the TPM.
  - The TPM Device Driver;
  - TPM Core Services;
  - TPM Service Provider.

TPM functional components

- The TPM incorporates the following functionality:
  - Key generation
    - Asymmetric key generation
    - Nonce creation
  - Cryptographic co-processor
    - RSA engine
    - Signing operations
    - Symmetric encryption engine
  - Execution engine
  - HMAC engine
  - SHA-1 engine
  - Power detection
  - Random number generation
  - Non-volatile memory
  - Volatile memory
  - Opt-in
  - 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:
  - Owner authorised commands (can only be executed by owner).
- TPM user.
- Challenger.
- Protected object owner.
- Intermediaries – migration.

Certification entities I

- The ***trusted platform module entity*** (TPME) attests to the fact that the TPM is genuine:
  - Digitally signs an endorsement credential containing the public endorsement key belonging to a particular TPM;
  - The TPME is likely to be the TPM manufacturer.
- The ***validation entity*** (VE) certifies integrity measurements, i.e. measured values and measurement digests, which correspond to correctly functioning or trustworthy platform components, for example embedded data or program code, to create validation certificates.
- The ***conformance entity*** (CE) guarantees, through the generation of signed conformance credentials, that the design and implementation of the TPM and trusted building blocks (TBB) within a trusted platform meet established evaluation guidelines.

Certification entities II

- The ***platform entity*** (PE) offers assurance, in the form of a platform credential, that a particular platform is an instantiation of a TP design, as described in conformance credentials, and that the platform's TPM is indeed genuine.

- A ***Privacy-CA*** attests that an identity (and an attestation identity key) belongs to a trusted platform.

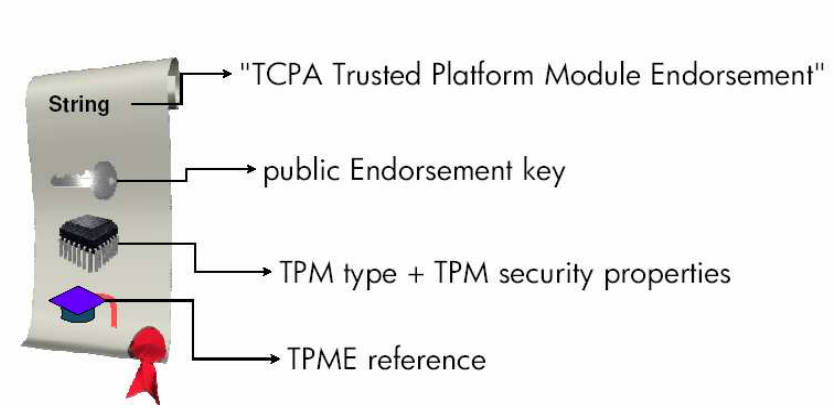
Endorsement Key Pair (EK)

- It is a fundamental requirement that:
  - Each TPM has a private endorsement key embedded in it;
  - The public half of endorsement key pair is certified by the TPME/manufacturer (in the endorsement credential).
- The EK is used by a TPM to prove that it is a genuine TPM.

- It is never used for signing.
- It is not a platform identity.
- It is only ever used for decryption in two scenarios:
  - To take ownership of a TPM;
  - To derive platform attestation identities/platform identities.

Credentials I

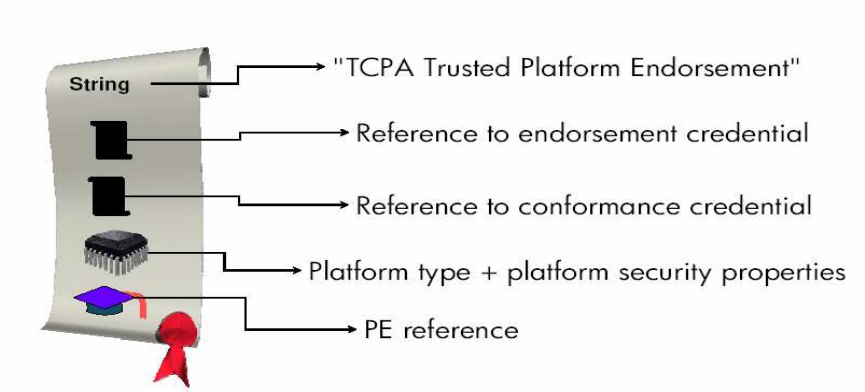
- ***Endorsement credential***:
  - Certifies that a public encryption key (the public endorsement key) belongs to a genuine TPM;
  - Constructed by a Trusted platform management entity.



Credentials II

- ***Conformance credential***:
  - 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.

- ***Platform credential***:
  - 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
  - Constructed by: Platform entity



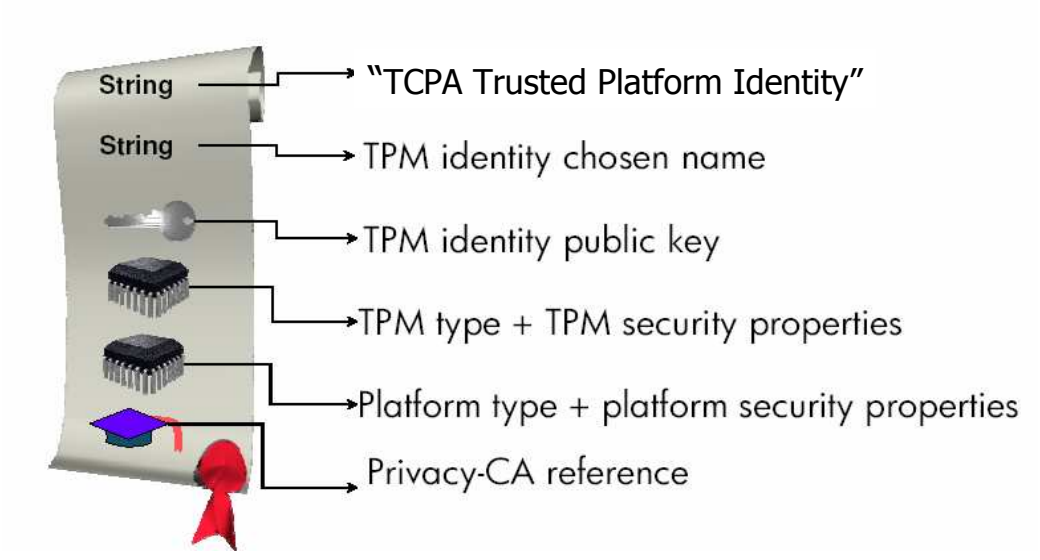
- These key pairs are used by a TPM to attest to platform properties to external entities.
- Used by a 'challenger' of the platform to verify that a TPM is indeed genuine, without identifying a specific TPM.
- A special trusted third party called a Privacy-Certification Authority (P-CA) supports the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

Generation of Attestation Identity Key Pairs (AIKs)

- TPM chooses an arbitrary key pair, an 'identity', and a P-CA which will attest to this new identity.
- The TPM signs the public key, the chosen identity, and the identifier of the chosen P-CA.
- The 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, verifies the credentials and the signature.
- The P-CA generates the ***Platform Identity Certificate***, a statement that the AIK and the identity being to a genuine trusted platform with the specified properties.

Platform identity certificate

Platform identity certificate:

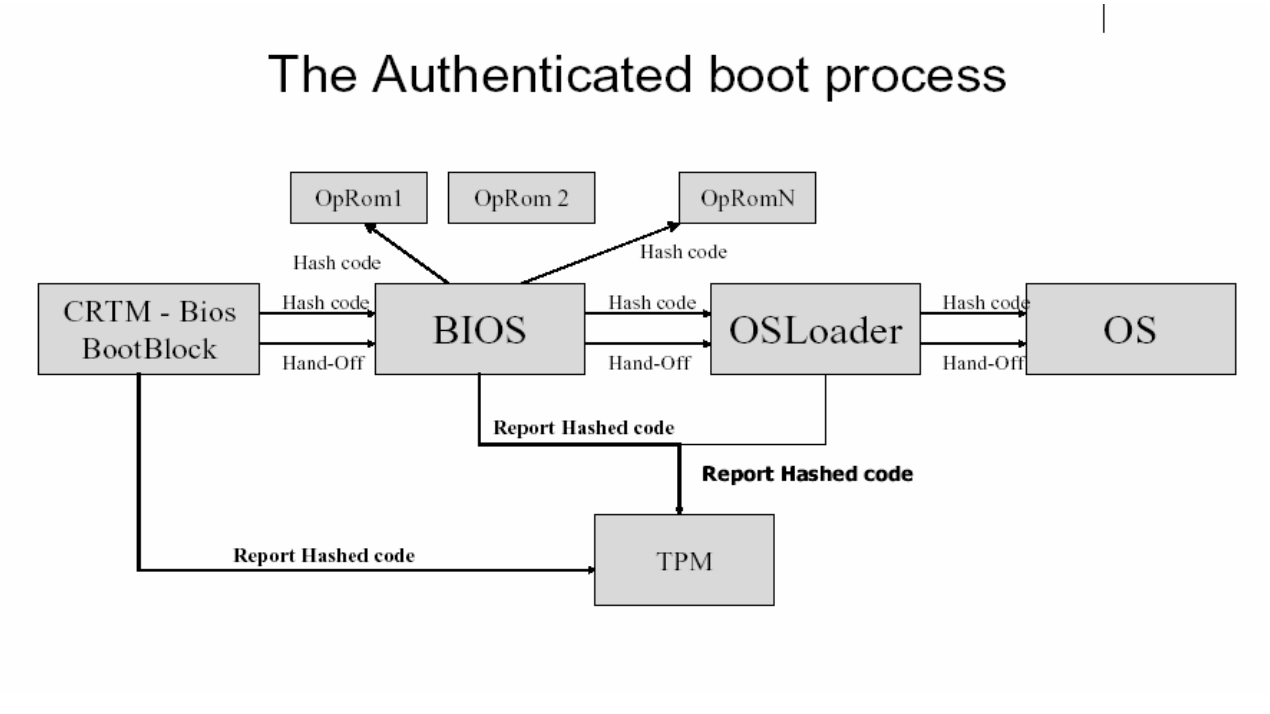


Sending the platform identity certificate to the TPM

- 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 key are then sent back to the requester, thus ensuring that only the appropriate TPM can access the certificate.

Issues with use of P-CA

- The P-CA gets to see all the platform credentials, including the endorsement credential (and 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.



- A TPM incorporates a set of Platform Configuration Registers (PCRs).
  - They are used to store platform software integrity metrics.
  - Usually a TP has several PCRs (a minimum of sixteen) and uses them to record different aspects of the state of the trusted platform.
  - Each storage register 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:
  - This is less expensive than holding all 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:
  - How trustworthy **the information is** using the signature of the platform on the measurements and the platform identity certificate;
  - How trustworthy **the software configuration** of the platform is using the reported measurements.

Secure boot – potential (1)

- The DIR (Data Integrity Register) is a TCG v1.1 function.
- It provides a place to store information using the TPM 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 – potential (2)

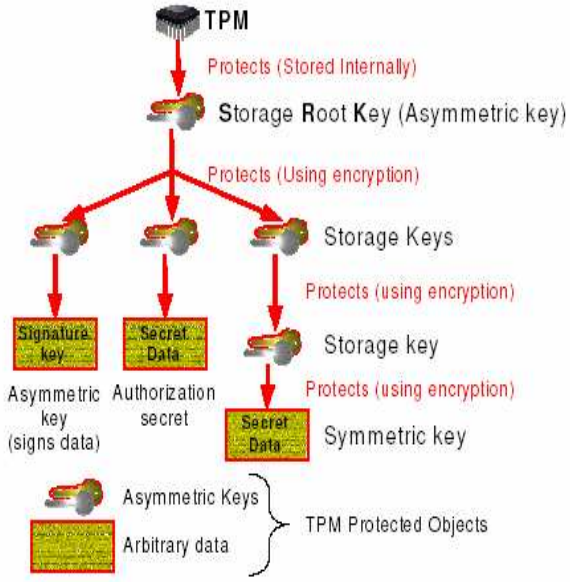
Proposed solutions:

- The TPM has the same number of DIRs as PCRs.
- The expected PCR values are written by the TPM owner to the DIRs.
- 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 halted.

Proposed solutions:

- 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.

- 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 such that they can only be decrypted by the TPM.
- Asymmetric encryption is used.



Secure storage I

- Binding (data):
  - This capability allows for external data to be encrypted under a public TPM parent key such that it can only be decrypted by the TPM.

- Wrapping (keys):
  - TSS Wrap Key: This capability allows an externally generated key to be encrypted under a parent key.

  Wrapping variants:
  - TSS Wrap key to PCR: Similar to above 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 II

- Sealing (data / symmetric 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 integrity metric sequence at the time the seal operation is performed and encrypted under 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)

- Access control applies:
  - Owner authorised commands;
  - Protected objects;
  - Before a TPM is owned, the TPM is unavailable

- 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 exist a number of authorisation protocols which protect against:
    - Man in the middle attacks
    - Replay
    - The exposure of the authorisation data

- Physical presence

Direct Anonymous Attestation, DAA

- P-CA:
  - Point of weakness: A P-CA is capable of:
    - User/TPM activity tracking; or
    - Making unwanted disclosures of platform information.

- DAA 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 the disclosure of the TPM public endorsement key, thereby removing the threat of a TTP collating data which may jeopardise the privacy of the TPM user.

Locality

- Allows TPM owner to assign privileges to external processes based on their locality.
- 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 any TPM resources.

Delegation

- Allows an owner to have fine-grained control over the use of specific owner authorised TPM commands.
- In the v1.1b TPM specifications, 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

- Implemented to improve the security of the communication channel between the TPM and trusted processes.
- Transport session provides integrity and confidentiality protection to commands sent to the TPM:
  - Integrity is provided by the MACing; and
  - Confidentiality is provided by the encryption of the command sent using a stream cipher, with keystream generated inside the TPM.
- The logging of commands sent to the TPM within a transport session is also facilitated.

Monotonic counters

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

Migration

- Non-migratable keys:
  - A non-migratable key is locked to a particular TPM and never duplicated;
  - It must be created by the TPM.
- Migratable key:
  - 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).
- A certifiable migratable key:
  - Keys created on 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

- Provides proof of a time interval not a time instance.
- It is the responsibility of the caller to associate the ticks 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

- TPM audit;
- Maintenance;
- Context management.

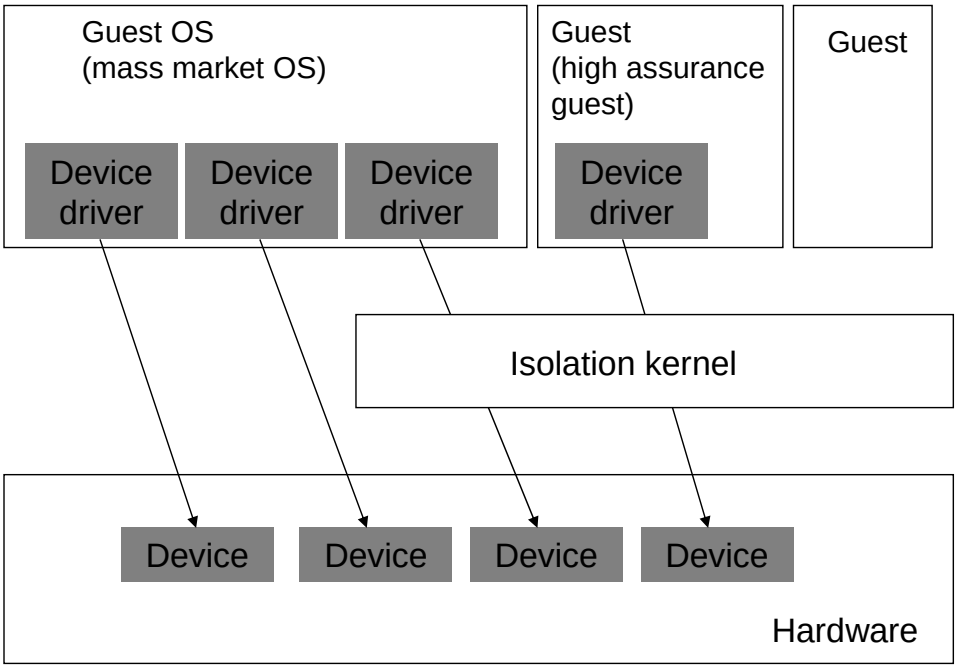
Contents

- 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?
- Academic example: Secure software download
- Industry example: OMA DRM v2 implementation – MPWG

Microsoft's Trusted Computing Initiative

- The Microsoft trusted computing initiative was originally introduced under the name Palladium.
- January 2003: The name Palladium was dropped:
  - Officially because the name had already been trademarked by another company
- The work has continued under the name NGSCB, for Next Generation Secure Computing Base.

NGSCB architecture



NGSCB architecture

- A TPM v1.2 (crypto chip/security support component (SSC)).
- The isolation kernel.
- Mass market OS + untrusted applications.
- High assurance components.

Crypto chip/SSC

- Required to provide the following services:
  - Authenticated boot
  - Persistent protected storage
    - Seal/unseal
    - Monotonic counter
  - Attestation
    - Quote
    - PkSeal/PkUnseal
- A TCG compliant TPM provides a concrete implementation of the abstract crypto chip.

The isolation layer – approaches: device support

1. Virtual machine monitor (VMM)
   - Expose devices to their guest OSs by virtualising them
     - A VMM intercepts a guest OS's attempt to access a physical device, and performs the actual device access on behalf of the guest with possible modifications of the request and /or access control checks
     - VMM co-ordinates access requests from guests to share devices among them.
   - This approach requires a driver for each virtualised device to be part of the isolation layer.
2. Export device to guest OSs
   - Isolation layer controls which guest can access a device
   - Device access by guests are made directly
   - DMA devices have unrestricted access to the full physical address space of the machine
   - Thus a guest in control of a DMA device can circumvent isolation layer protections

The isolation layer – approaches: OS compatibility

1. VMM
   - Exposes the original hardware interface
     - Support for off the shelf OSs
   - Increase the complexity of the isolation layer
     - Particularly on PC hardware where x86 CPU is not virtualisable
2. Exokernels / microkernels
   - Expose different interfaces
   - Require new OSs to be written or existing OSs modified

The isolation kernel I

- The isolation layer exposes the original hardware interface to one guest.

The CPU

- The x86 CPU has four protection rings (rings 0-3)

- Upcoming versions of x86 processors – new CPU mode
- More privileged than existing ring 0 (effectively ring -1)
- The Microsoft isolation kernel will execute in this ring
  (virtualisability problems thwarted)

The isolation kernel II

Memory

- In order to partition memory among guests: virtualisation
- Instructions that execute on the CPU, address memory through virtual addresses
- Each virtual address is translated to a physical address, which is then used to access physical resources
- Mapping: page map
- The page table edit control (PTEC) algorithm is used to partition physical memory among its guests
- Any attempt by a guest to edit its page map traps to the isolation kernel which consults its security policy
- This provides isolation between guests

Devices

- On 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

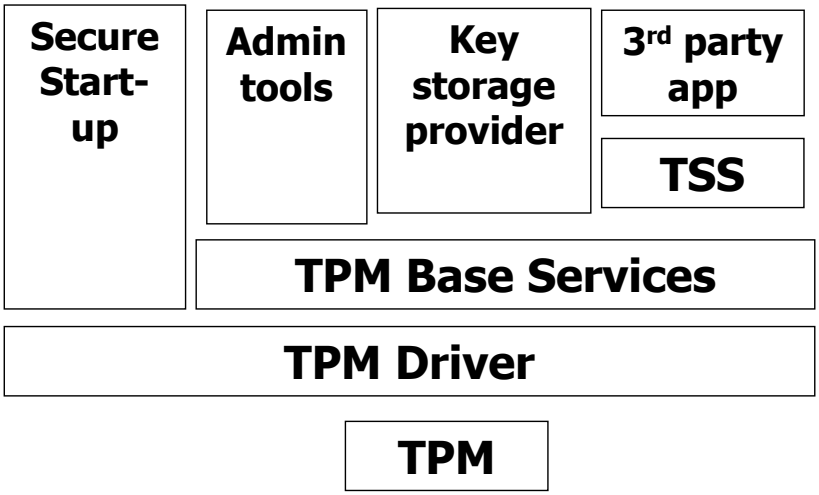
Direct Memory Access (DMA) Devices

- On existing PC hardware, they have access to the full physical address space
- Therefore a guest in control of a DMA device could circumvent the virtual memory protections described
- 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

The isolation kernel

- 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



- The TPM device driver designed for the TPM v1.2 chip
- The TPM base services – a service that provides sharing of limited resources on the TPM
- The Secure Startup, Admin Tools, and Key Storage Provider components are Microsoft applications and services that rely on TPM Services.
  - Secure start-up
  - TPM admin tools (example – curtail use of TPM commands that may reveal privacy information about user or workstation)
  - Key storage provider
- The “3rd-party Application” and TSS components are third-party components that rely on TPM Services
  - No plans for v1.2 compliant TSS for Longhorn
  - Work with TSS vendors to create TSS products that interface with TBS infrastructure

Secure Startup – Full Volume Encryption I

- Renamed BitLocker Drive Encryption (final feature release name).
- BitLocker Drive Encryption provides full volume encryption of the Windows volume, which helps protect data from being compromised on a lost or stolen machine.
- 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.

Secure Startup – Full Volume Encryption II

- BitLocker also stores measurements of system volume in a TPM chip
  - Sealing mechanism
- Every time the computer is started, Windows Vista verifies that the system volume has not been modified in an offline attack
- An offline attack is a scenario in which an attacker boots an alternative operating system in order to gain control of the system
- If the system volume has been modified, Windows 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?
- Academic example: Secure software download
- Industry example: OMA DRM v2 implementation – MPWG

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.

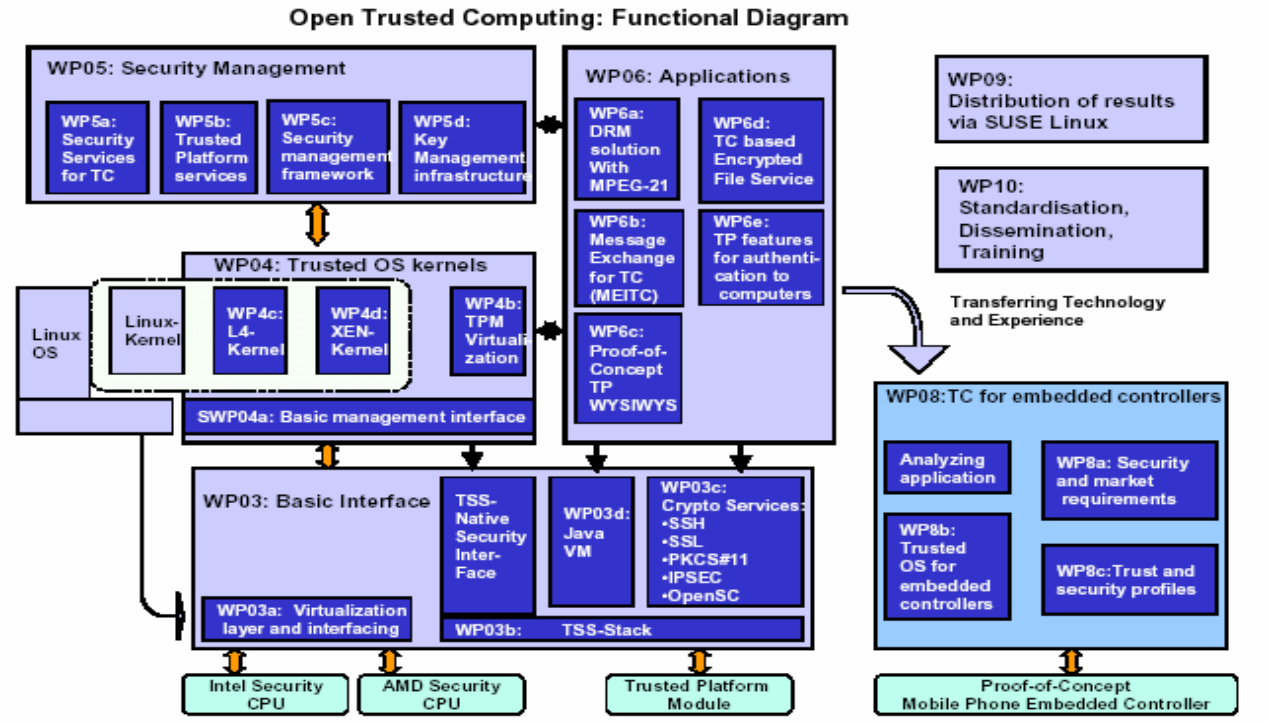
- A v1.2 TPM must also be added.

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



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

- 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?
- Academic example: Secure software download
- Industry example: OMA DRM v2 implementation – MPWG

Software security – What TC will not do!!

Software vulnerabilities result from both:

- Design errors
- Coding errors – TC technologies will not prevent or aid in the development of secure software without vulnerabilities

Vulnerabilities are waiting to be attacked by viruses and worms

- Viruses/malicious code – will not stop them being written or circulated

The isolation layer - What TC will do!!

Security of the isolation layer code itself

- Size of isolation layer – lines of code
- Provably secure
- Separation of the isolation layer –> ring -1
  - (helps prevent a potential attack from 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 from 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
    - (helps prevent a potential attack from malicious software)
- Secure channel to output devices to ensure integrity of output can be assured
    - (helps prevent a potential attack from malicious software)

The TPM - What TC can do!!

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

The TPM - What TC can do!!

- Software may be built to leverage 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

Contents

Protection of broadcast content

- New business model:
  - Delivery of broadcast services to *mobile* receivers, with services available from many broadcasters
- Current protection mechanisms:
  - Designed for relatively *static* receivers and services available from a small number of broadcasters
- Common scrambling algorithm
- Conditional access systems
- Common Interface:
  - Consumers require multiple PC-Card modules – cost, inconvenience, unsuitable for mobile devices
- Simulcrypt:
  - Broadcasters install and maintain multiple CA systems – cost, maintenance issues
- Current mechanisms not designed for mobile receivers

Potential Solution

- Download proprietary applications to mobile devices on demand
- Problem:
  - Applications, and providers, are security sensitive
  - Lack of trust in the mobile host:
    - Piracy: protection of proprietary algorithms, keys
  - Host needs to demonstrate that it can be trusted:
    - Application needs protection – not the host
- Trusted Computing provides the mechanisms to demonstrate trust

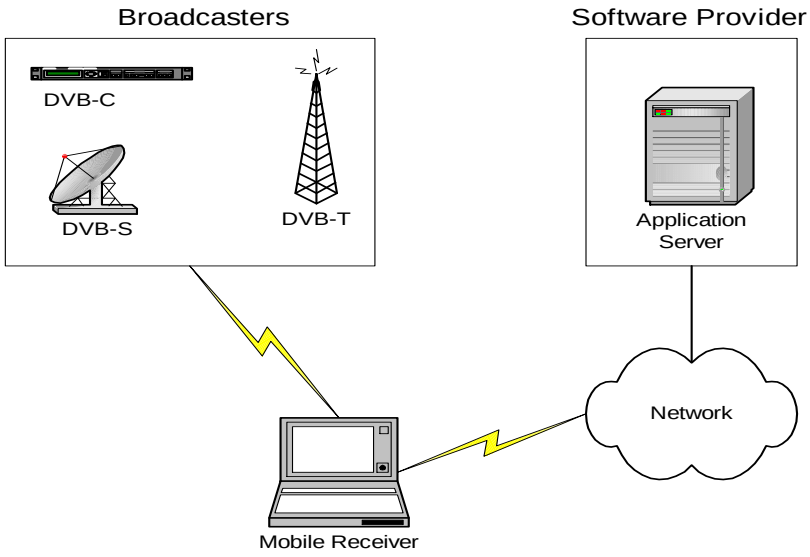
Required security services

1. Confidentiality of application in transit
2. Integrity of application in transit
3. Entity authentication:
   - Host
   - Application provider
4. Origin authentication of application
5. Freshness of messages
6. Confidentiality and Integrity of application in while in storage on the device (AC mechanisms to protected the application on the device)
7. Confidentiality and Integrity of application in while executing on the device

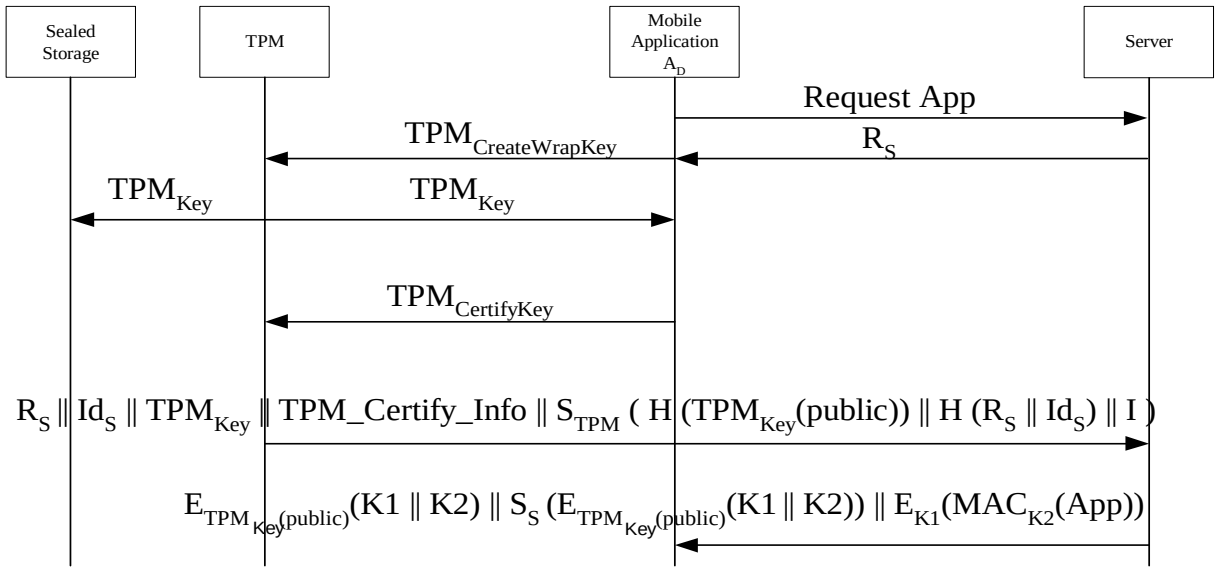
Corresponding security mechanisms

1. Symmetric encryption
2. MACing of the application
3. Entity authentication protocol runs as described in ISO 9798-3 (Host and application provider)
   - Attestation (Host) as described within TCG TPM specification set
4. Digital signature of the application provider on the symmetric keys used in 1 and 2
5. Nonces/ timestamps
6. Protected/secure storage, as described in TCG TPM v1.2 specification set
7. Memory isolation techniques, as described by Microsoft with respect to NGSCB, for example

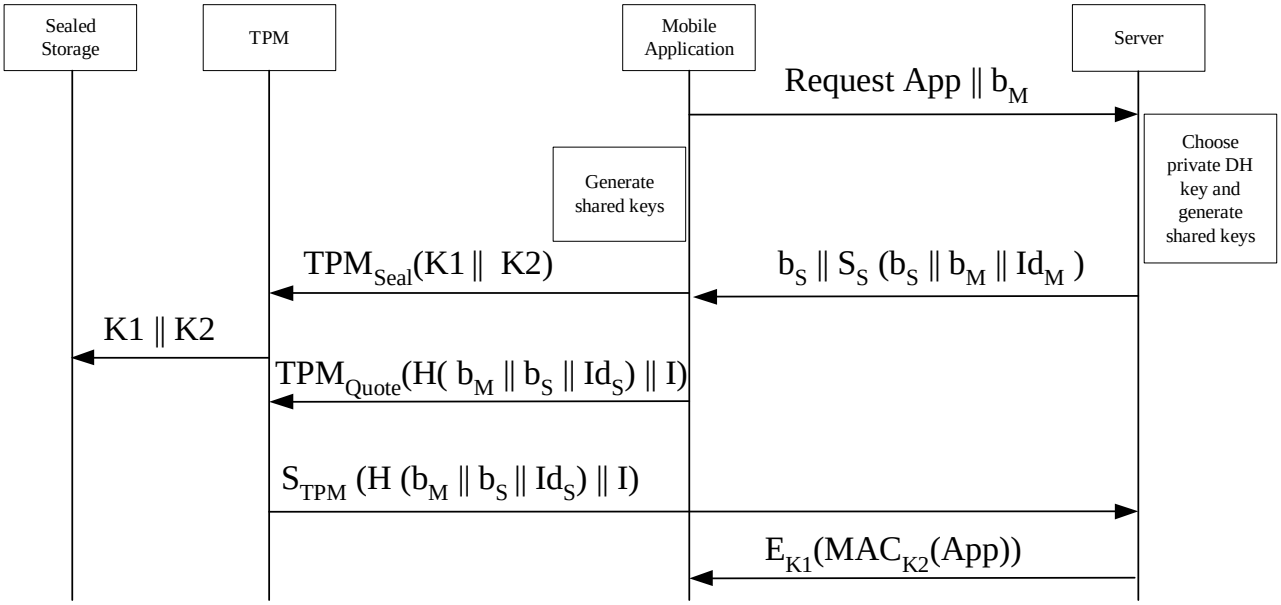
Model



Protocol 1



Protocol 2



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?
- Academic example: Secure software download
- Industry example: OMA DRM v2 implementation – MPWG

OMA

- The Open Mobile Alliance (OMA) was founded in June 2002
- One of the original objectives of the OMA was to define a DRM specification set for use in the mobile environment
- OMA DRM v1 was published as a candidate specification in October 2002, and in 2004 was approved as an OMA enabler specification after full interoperability testing had been completed

- Following this, in 2004, work on OMA DRM v2 was completed and OMA DRM v2 was published as a candidate specification in July 2004
- OMA DRM v2 builds upon the version 1 specifications to provide higher security and a more extensive feature set

OMA DRM v1

- Main goals:
  - Timely and inexpensive to deploy
  - Easy to implement on mass market mobile devices
  - Finally, it was required that the initial OMA DRM solution did not necessitate the roll-out of a costly infrastructure

- Three classes of DRM functionality:
  - Forward lock
  - Combined delivery
  - Separate delivery

OMA DRM v1

- Weaknesses in OMA DRM v1:
  - A rights issuer has no way in which to determine whether the requesting device supports DRM
  - In the separate delivery DRM class, where the content is encrypted, the content encrypting key is not protected
  - The device has no way of authenticating the rights issuer and therefore may be sent bogus rights objects from an entity claiming to be the legitimate rights issuer

OMA DRM v2

- Both device authentication and rights issuer (RI) authentication are provided
- Mechanisms are deployed in order to protect the confidentiality of media objects
- Mechanisms are also deployed such that the OMA DRM v2 agent can determine whether a media object received from a RI has been modified in an unauthorised way

- Also support for an extended feature set: subscription, streaming content, reward schemes, domains, unconnected devices

Supported security features

- The rights object acquisition protocol (ROAP) suite
  - The 4-pass registration protocol
  - The 2-pass rights acquisition protocol
  - The 1-pass rights acquisition protocol
  - The 2-pass join domain protocol
  - The 2-pass leave domain protocol
- A trust model enables an RI to obtain assurances about DRM agent behaviour and the robustness of the DRM agent implementation
  - It is the responsibility of the Content Management Licensing Administrator (CMLA), or a similar organisation, to provide a trust model, i.e. robustness rules, and to define actions which may be taken against a manufacturer who builds devices which are not sufficiently robust

Lifecycle of a OMA DRM v2 agent

- OMA DRM v2 agent installation
- The rights object acquisition protocol (ROAP) suite
  - The 4-pass registration protocol
  - The 2-pass rights acquisition protocol
  - The 1-pass rights acquisition protocol
  - The 2-pass join domain protocol
  - The 2-pass leave domain protocol
- Threat analysis
- Requirements extraction
- Mapping to TPM and TSS specifications

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 vast 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

Open_TC EC Contract No: IST-027635

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