Open Trusted Computing

Royal Holloway
University of London

Challenges for trusted computing

Chris Mitchell

Royal Holloway, University of London

c.mitchell@rhul.ac.uk

Contents

1. Trusted computing: A brief introduction
2. PKI requirements
3. Revocation issues
4. Attestation in practice
5. Attacks on the technology
6. Compatibility and usability
7. Mobile issues

1. Trusted computing: A brief introduction

- Objectives:
  - Give a brief overview of the history of trusted computing technology;
  - Review the main technological objectives and components of trusted computing.

TCG specifications

- The Trusted Computing Group (TCG) publishes its completed specifications freely on the web.
- Specifications under development are not freely available – they are for ‘members only’.
- However, there is a liaison programme for academic institutions, which gives access to documents (under NDA) without charge.
- The v1.2 TPM specifications (the current version) have recently been adopted as an international standard: ISO/IEC 11889 parts 1-4 (with the title *Information technology - Trusted Platform Module*) – published in May 2009 under auspices of ISO/IEC JTC1/SC27.

TCG trusted computing: Basic components and services

- **Integrity measurement** – a cryptographic hash of a platform component (i.e. software executing on the platform);
- **Authenticated boot** – process by which a platform’s state (the sum of its components) is reliably measured and stored;
- **Sealed storage** – process of storing data on a platform in such a way that the data can only be retrieved if the platform is in a particular state;
- **Attestation** – process of reliably reporting the platform’s current state;
- **Isolated execution** – enables the unhindered execution of software.

Trusted computing
Platform components

- Implementing Integrity measurement, Authenticated boot, Sealed storage and Attestation depends on the three **Roots of trust**:
  - Root of trust for measurement (RTM);
  - Root of trust for storage (RTS);
  - Root of trust for reporting (RTR).
- These are “components that must be trusted if the platform is to be trusted”.

Roots of trust I

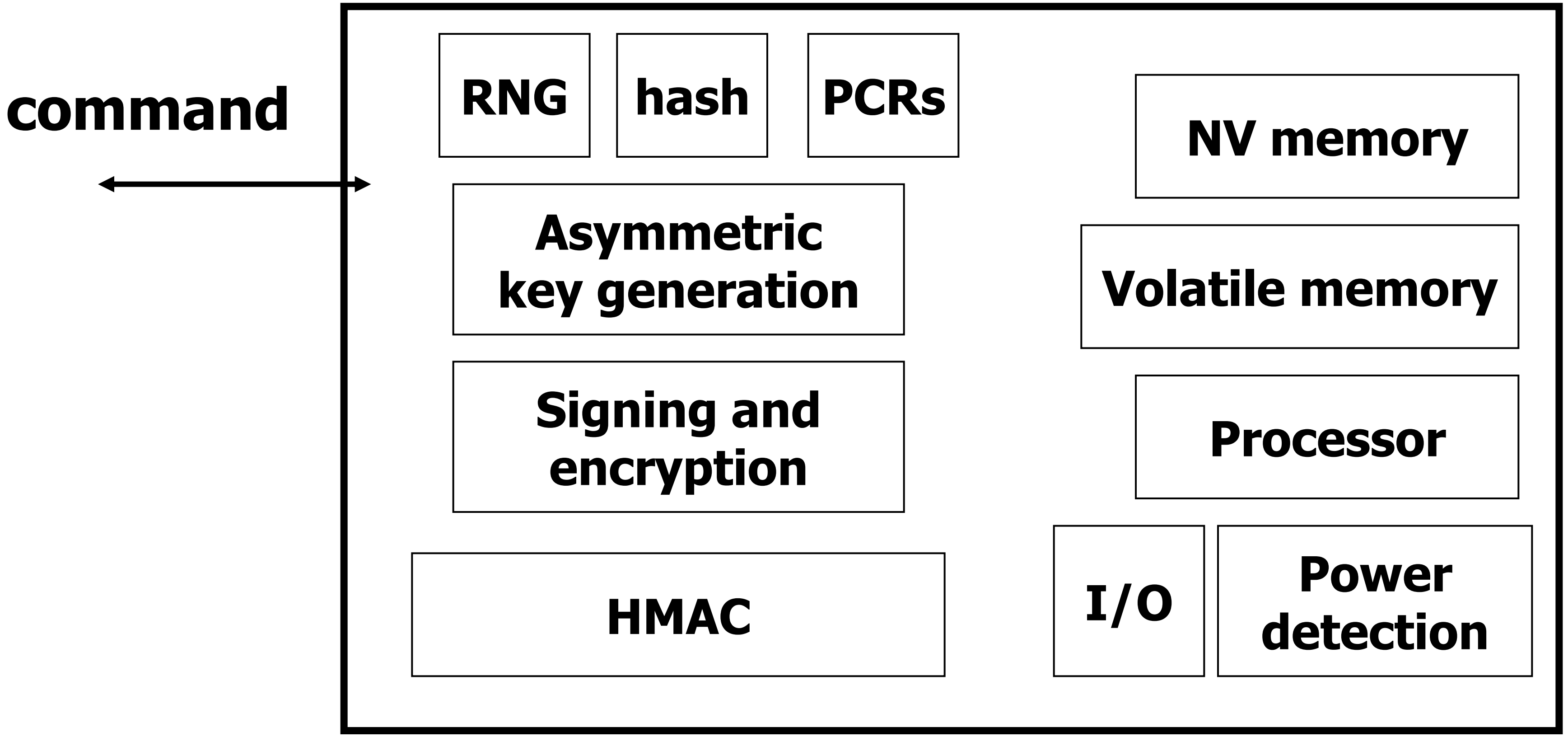
- **The RTM**:
  - The RTM is a computing engine which accurately generates at least one integrity measurement event representing a software component running on the platform;
  - For the foreseeable future, it is envisaged that the RTM will be integrated into the normal computing engine of the platform, where the provision of additional BIOS boot block or BIOS instructions (the **Core RTM** or **CRTM**) cause the main platform processor to function as the RTM.

Roots of trust II

- The **RTS** is a collection of capabilities which must be trusted if storage of data inside a platform is to be trusted:
  - Storing accurate summary of integrity measurements (platform state information);
  - Integrity and confidentiality protection of data;
  - Sealing.
- 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 RTS and RTR constitute the minimum functionality that should be provided by a **Trusted Platform Module (TPM)** – which is typically implemented as a hardware chip bound to the platform.

The TPM

A TPM is typically implemented as a chip mounted on the motherboard of its host platform.



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* (can only be executed by owner).
- **TPM user** (may be different to TPM owner).
- **Challenger** (wishing to verify platform state).
- **Protected object owner** (owner of data/software on a platform, which may be distinct from 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 several 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 trusted building blocks (TBBs) in 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, conformance credential(s), and a platform credential.

Certification entities II

- Two other certification entity types are defined:
  - A **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 a validation certificate.
  - A **Privacy-CA (P-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 keys include:
  - **Endorsement Key (EK)**, an asymmetric encryption key pair, unique per TPM, and typically generated at time of manufacture;
  - **Attestation Identity Keys (AIKs)**, i.e. signature key pairs, generated by the TPM during use – a TPM may have many;
  - **Storage Root Key (SRK)**, an asymmetric encryption key pair used to support secure storage of data external to the TPM.

Endorsement Key Pair (EK)

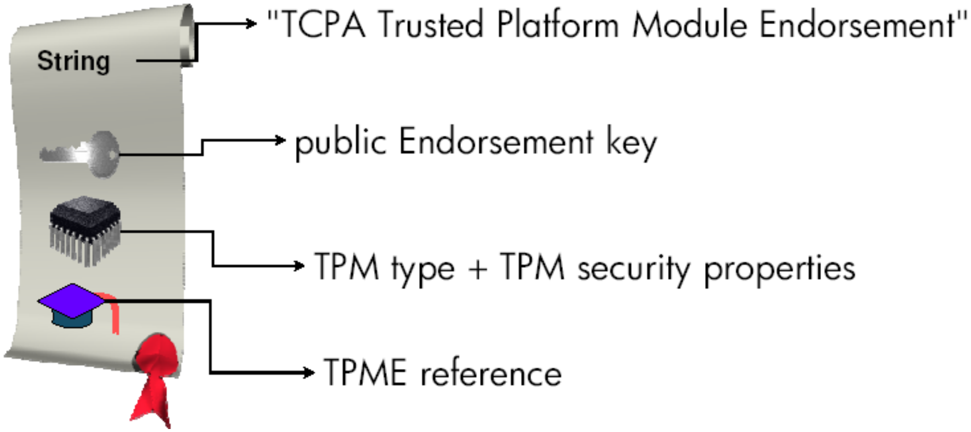
- It is a fundamental requirement that:
  - Each TPM has an endorsement key pair stored in it;
  - The public part of the endorsement key pair is certified by the TPME (e.g. the TPM manufacturer) in the form of the endorsement credential.
- The private part of the EK is used by a TPM to prove that it is a genuine TPM. It is never used for signing.
- It is only ever used 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 attest 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;
  - is signed by a Trusted Platform Management Entity.

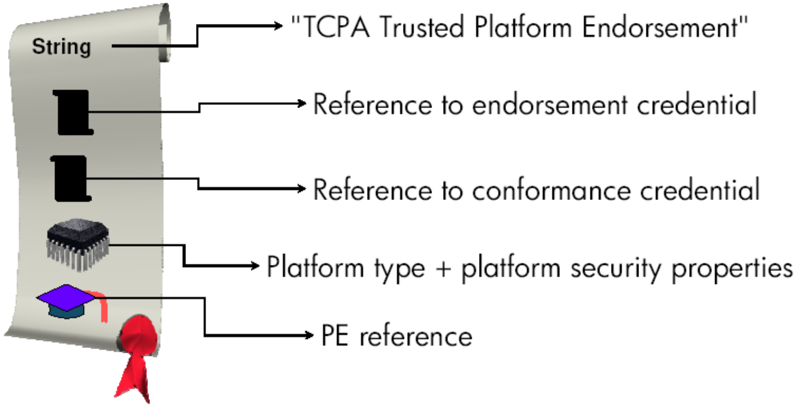

String
"TCPA Trusted Platform Module Endorsement"
public Endorsement key
TPM type + TPM security properties
TPME reference

Credentials II

- A **Conformance credential** is:
  - a document that vouches that the design and implementation of the TPM and the trusted building blocks (TBBs) 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.


String
"TCPA Trusted Platform Endorsement"
Reference to endorsement credential
Reference to conformance credential
Platform type + platform security properties
PE reference

Attestation Identity Key Pairs (AIKs)

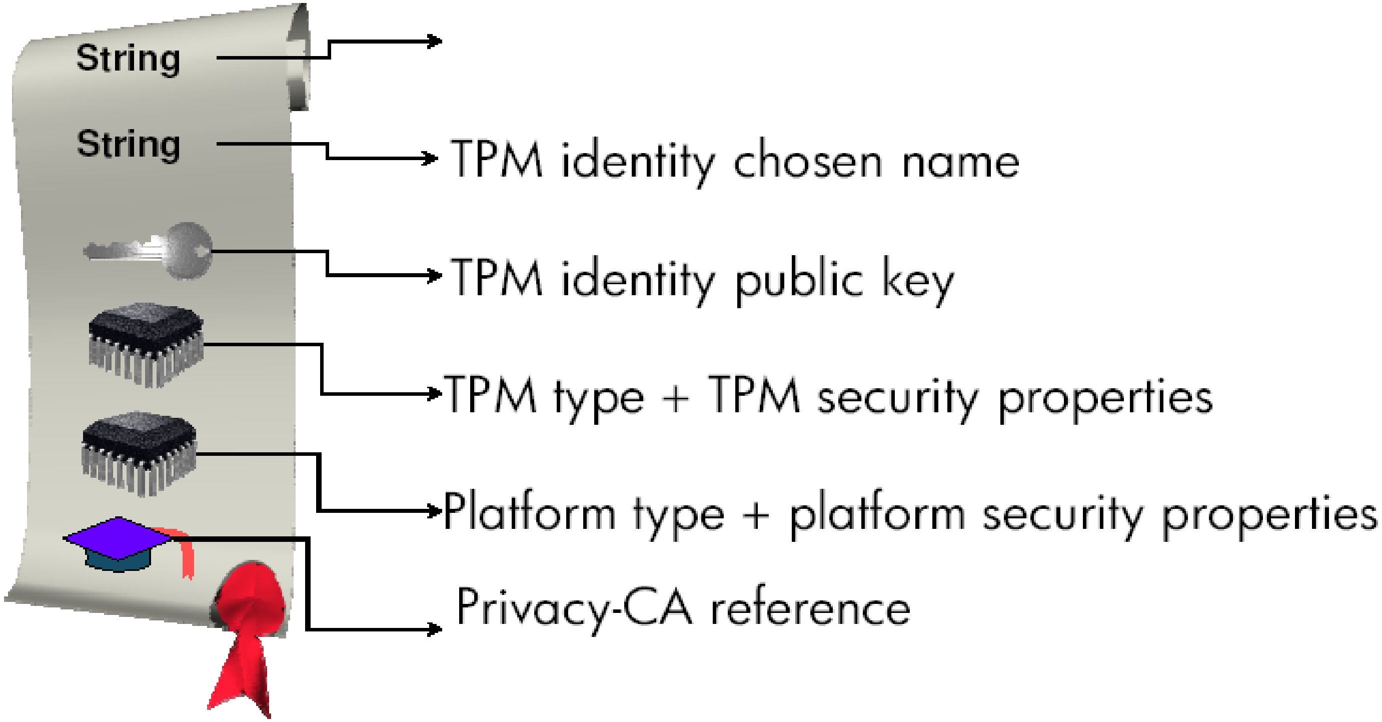
- These signature key pairs are used by a TPM to attest to platform properties to external entities.
- Used by a ‘challenger’ of the platform to verify that a TPM is indeed genuine, without identifying a specific TPM.
- A special trusted third party called a Privacy-Certification Authority (P-CA) supports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

Generation of AIKs

- TPM chooses a new AIK pair, an ‘identity’, and a P-CA which will be requested to attest to this new identity.
- The TPM signs the public key, the chosen identity, and the identifier of the chosen P-CA, using the newly generated AIK private key.
- 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

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



Sending the platform identity certificate to the TPM

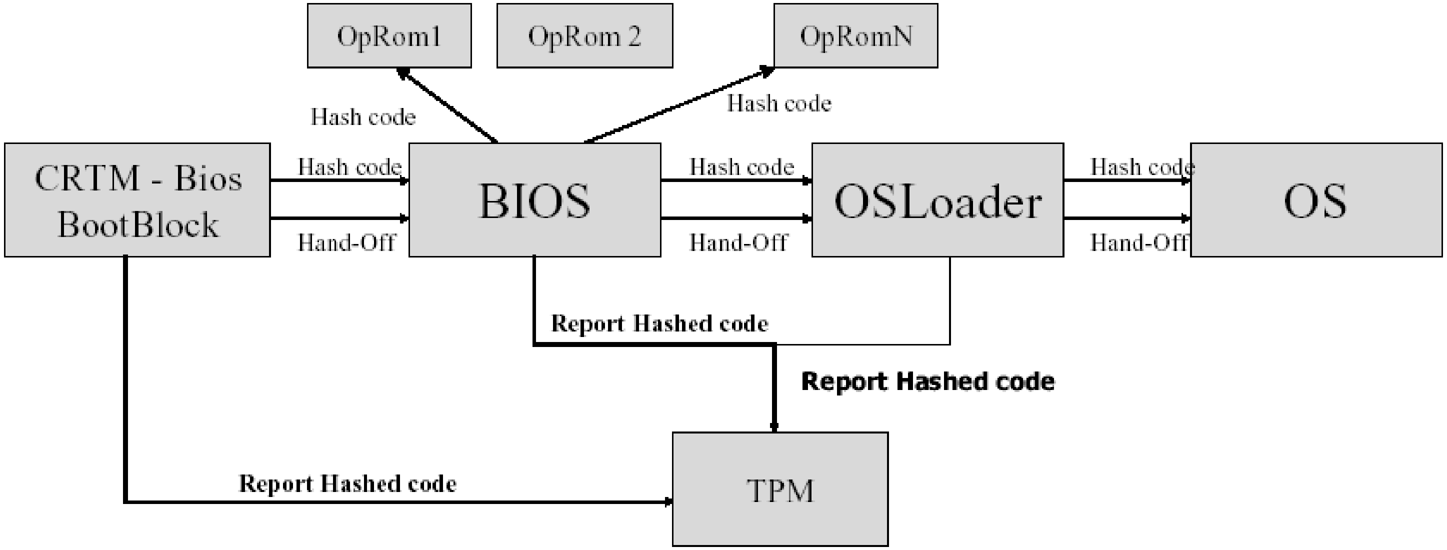
- The P-CA generates a random secret encryption key.
- The platform identity certificate is encrypted using this secret key.
- The secret key is encrypted using the TPM’s public EK.
- The encrypted certificate and key are then sent back to the requester, thus ensuring that only the appropriate TPM can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

Issues with use of 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 its 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.
- As a result, an alternative protocol called DAA (Direct Anonymous Attestation) has been devised which avoids this problem.

Authenticated boot

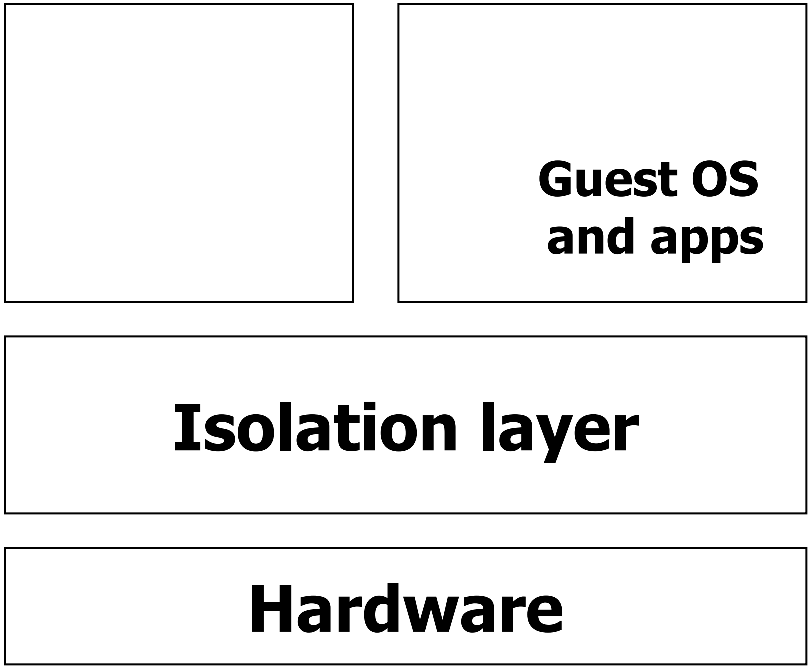
The Authenticated boot process



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 of a signed integrity report 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.

Trusted computing fundamentals
Isolated execution environments



Example implementations include: OS-hosted VMM (VMWare workstation), Stand-alone VMM (Terra), Hybrid isolation layer (XEN 2.0), Hardware supported isolation layer (NGSCB).

- Protection from external interference
- Observation of isolated environment activity only by controlled inter-process communication
- Secure communication between isolated environments
- Trusted path between a program running in an isolated environment and I/O devices

2. PKI requirements

- In the introduction we mentioned a number of certificates that are used by trusted computing technology.
- These are generated by a variety of entities, and constitute a potentially huge and highly complex PKI.

Credential issue

- For a platform to be considered trusted, it must first obtain the following core credentials from CAs of specific types, namely:
  - An endorsement credential from an endorsement CA;
  - A platform credential from a platform CA; and
  - One or more conformance credentials from conformance CAs.
- In order to address privacy concerns resulting from routine use of an EK, the TCG introduced the ability for a TPM to generate and use an arbitrary number of pseudonyms, in the form of AIKs.
  - A Privacy-CA (P-CA) verifies core credentials.
  - Provides assurance that an AIK is bound to a genuine TP in the form of AIK credentials.

Privacy CAs and DAA

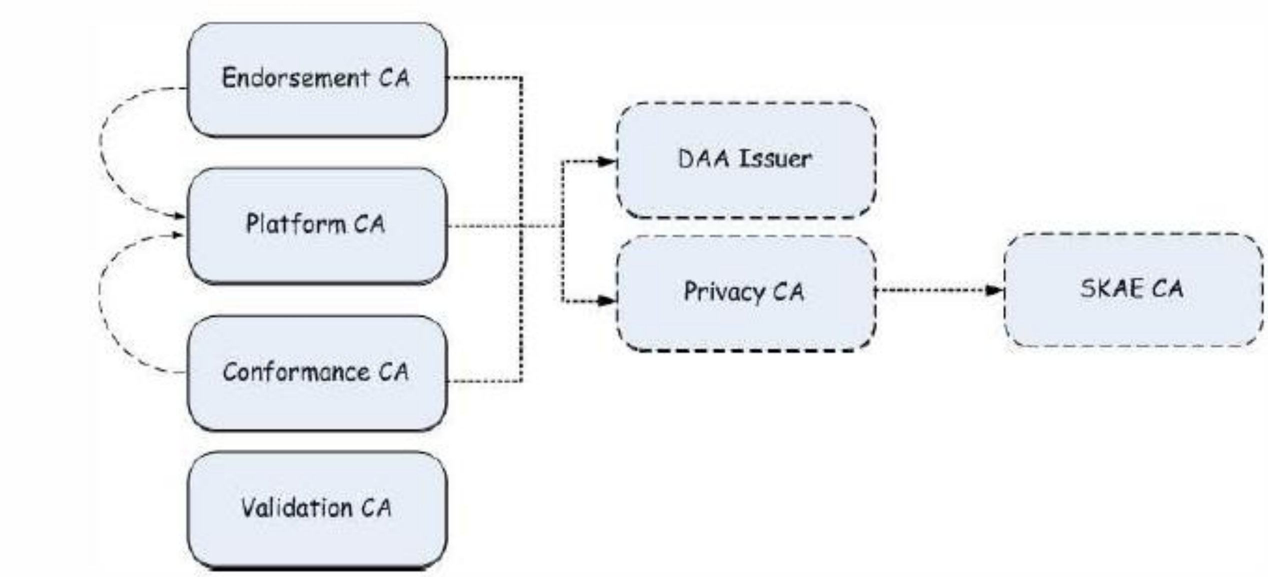
- P-CAs have been criticized as a “point of weakness”.
  - Capable of linking EK – AIK pairs.
- Direct Anonymous Attestation (DAA), was introduced as a response to this criticism:
  - A DAA-CA can produce a DAA credential for a TP.
  - In turn can be used to sign AIK credentials.
  - DAA enables TP attestation with no P-CA linkage possible between EK-AIK pairs

‘Translation’ CAs

- Yet another class of CA has been introduced to attest to the usage, mobility and authorisation constraints associated with private keys held by a TPM.
  - A Subject Key Attestation Evidence (SKAE) CA is responsible for issuing X.509 certificates which allow a verifier to ascertain that an operation involving a private key can only be performed within a TCG-compliant TPM environment.
  - Enables TCG keys to be integrated into traditional protocols, e.g. SSL/TLS, IPsec.

PKI dependencies

- A trusted computing PKI is complex, not just because of the sheer number of CAs, but also because of a series of implicit dependencies amongst many of these CAs.



- A platform CA relies on the due diligence of an endorsement CA and one or more conformance CAs in accrediting components of a trusted platform.

Dependencies and liability

- Both privacy-CAs and DAA CAs rely on platform CAs, endorsement CAs and one or more conformance CAs.
- Furthermore, SKAE CAs rely on the due diligence of Privacy CAs or DAA CAs in evaluating the accreditation evidence provided by a trusted platform.
- Where will liability will lie? Certificate Policies and Certification Practice Statements are notoriously difficult/costly to create and so act as a barrier to entities wishing to provide CA services.
- If we move away from islands of trust, trusted computing relies heavily on a global PKI infrastructure.
- The development of any functional PKI requires a sophisticated combination of organisational, policy-oriented, procedural, and legislative approaches

3. Revocation issues

- Any use of a PKI, raises issues of revocation.
- That is, how do we disseminate information about compromised private keys, and/or discredited third party service providers.

Credential revocation

- Given the complex dependencies between many of the TC-PKI credentials, the compromise of an individual key and the subsequent revocation of its associated public key certificate may result in a cascading revocation of all dependent TPM credentials.
  - For example, in the event of AIK revocation, all SKAE certificates associated with the newly revoked AIK must also be revoked.
  - This implies that multiple CAs, potentially in independent domains, must be contacted in a timely manner and informed about a revocation decision.
    - Potentially time-consuming and costly endeavour.
- Further problems are introduced with DAA.

Revocation infrastructure

- Who pays and who gets paid?
  - In the SET PKI the issuing bank carries the cost (and associated risk) of handling certificate enrolment and issue while the acquiring bank obtains all the benefits.
  - In Sweden, certificates issued to citizens are free, but a validity check costs €0.25.
  - Low assurance certificates makes serious revocation handling financially infeasible.

TPM revocation

- It remains unclear how TPM compromise can be detected.
- TPM compromise could give rise to an excessive number of certification requests originating from a single TPM host platform (‘excessive’ determined by risk-management policy).
  - CAs may specify different thresholds for determining what is meant by ‘excessive’.
  - Once detected, revocation information needs to be globally propagated to prevent the compromised TPM host platform from being (mis)used elsewhere. This requires the establishment of a global revocation infrastructure.
  - For legal reasons, a CAs may be reluctant to announce suspected compromises.
  - To alleviate the risk of a malicious P-CA issuing falsified revocation statements, a means of assessing the credibility of CAs in issuing such statements is needed. It is currently unclear what form such a mechanism might take.

PKI and PKI-dependant applications

- At present most of the PKI infrastructure doesn’t exist.

4. Attestation in practice

- Attestation is great in principle, but has many problems in practice.

Attestation

- The exact parameters to be considered when performing integrity measurements on platform components have yet to be standardised.
  - At a minimum, the parameters must be chosen so that each software component's integrity measurement can be uniquely identified.
  - These measurements must also remain consistent to allow ease of verification.
- As the number of platform components increases, so does the complexity of third party verification of attestation statements.
- It also becomes difficult for a challenger to verify a single component running on a platform.
- Isolation technologies simplify the task of state verification.

Attestation

However, binary state information remains problematic:

- It says little about program behaviour.
- It is not expressive.
- Upgrades and patches to programs are difficult to deal with.
- Solution = property based attestation??

Delegation-based property attestation	Derivation-based property attestation
Use a trusted third party to map states to properties. Problem: Moving the goalposts. Who provides this facility and what exactly is a property?	Send code to remote VM to test for properties. Problem: Possible limitations on what can be tested.

Virtualisation support

- There is very limited support for hypervisors, necessary to exploit the full power of trusted computing on general purpose computing platforms.
- Microsoft Windows Server Hyper-V does not (yet) support use of TPM.
- Xen hypervisor does support use of TPM, and provides virtualised TPM support to hosted operating systems.

Virtualisation architectures

- However, both Hyper-V and Xen provide limited support for true high security environments.
- Both rely on an architecture in which a ‘domain 0’ (root) partition has full control over the hardware, and in which the full set of drivers are run.
- Domain 0 will typically contain a copy of the full operating system (Windows Server 2008 or Linux).
- The attack surface may well be smaller than when an operating system is running on an ‘unvirtualised’ environment, since user applications may not run there, but vulnerabilities in the operating system and/or drivers may still be exploitable.

5. Attacks on the technology

- We next consider some of the shortcomings that have been identified with the technology itself.
- Considering the complexity of the hardware, it is perhaps surprising that more serious issues have not been identified.

Use of cryptography

- 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 new TPM specification in the next couple of years (TPM.next), which looks likely to use crypto in a more flexible way (e.g. with algorithm identifiers, as in X.509, instead of fixed algorithms).

An anonymity attack

- Rudolph (SEC 2007) showed how a malicious DAA issuer could compromise the anonymity properties of DAA.
- The corrupt DAA issuer uses a different key pair for every TPM to which it issues a credential.
- Credential use can then be linked back to the particular DAA issuer key pair, and hence to a particular TPM.
- In practice, such an attack would probably be readily detectable, unless it was only used in a very targeted way.

Corrupt administrator attacks

- Smyth, Ryan and Chen (ESAS 2007) described another possible attack on the use of DAA.
- DAA is designed to enable two interactions by the same platform with the same verifier to be linked (this, in turn, enables platforms to be revoked).
- This function is based upon the identifier used by the verifier.
- If a verifier an the DAA issuer used the same identifier, then a platform could be identified, breaking anonymity.
- Such an attack may be detectable in practice.

DAA variants

- A number of variants of DAA, each with its own particular properties, have been proposed to address identified performance and privacy issues.
- However, whether any of these will make it into the next set of TPM specifications is rather unclear.

Application vulnerabilities

- Bitlocker (part of Vista) is one of the few applications of trusted computing out there in the wild (and in use).
- It has been designed to provide full volume encryption, e.g. to protect against loss of data from stolen/lost laptops.
- It is, however, only designed to protect against opportunistic attacks.
- As shown by Tuerpe et al. from Fraunhofer Darmstadt (in Trust 2009), Bitlocker, even when using trusted computing, does not prevent targeted attacks.

Authorisation data attack

- Chen and Ryan (FTC 2008) showed how ‘weak’ choices for authorisation passwords could be compromised.
- This attack arises because of the way in which authorisation data is encrypted.

6. Compatibility and usability

- We next look at issues arising from potential compatibility and usability problems.

Backwards Compatibility

- As a consequence of the piecemeal roll-out of Trusted Computing technologies, current trusted platforms do not come equipped with fully-integrated RTMs, isolation technologies, processors or chipset extensions.
- Indeed, many platforms don’t come equipped with any credentials vouching for the ‘trustworthiness’ of the platform (or its components).
- This has the potential to create an awkward backwards compatibility issue, as and when fully equipped trusted computing platforms become available.

Usability

- Prevailing wisdom suggests that it is prudent to hide the complexities of security technology from end-users.
  - Applications that have relied on a PKI have failed in cases where security functions have been too unwieldy to be usable by non-experts.
  - In one example, the PKI experience was considered so painful by some users that they refused to use the technology if it involved handling certificates.
- By contrast, using a TPM currently requires a detailed understanding of how the underlying technology works.
  - For example, the very act of enabling a TPM prior to its use is a non-trivial task, requiring a user to understand and edit BIOS settings.
  - Once enabled, a user is further confronted with setting a TPM owner password, selecting key types fit for purpose, and enrolling certain keys within a PKI.
  - Further problems may arise from password use and management.

Non-Compliance and Interoperability

- Unfortunately, many of the technological building blocks required to instantiate a trusted platform are not standardised, nor does the TCG dictate implementation specifics to its adopters.
  - As a result, a number of currently available TPMs do not comply with the TPM specifications.
  - The current absence of conformance testing facilities suggests that the production of non-compliant TPMs may very well continue for the foreseeable future.
  - In turn, discrepancies in implementation between TPM manufacturers may limit future interoperability between different trusted platforms.

A parallel trusted computing world?

- China seems intent on devising its own set of trusted computing specifications, based on Chinese cryptographic techniques,
- This will result in a completely parallel (and incompatible) set of trusted platforms being manufactured solely for the Chinese market,
- This has the potential to seriously damage the chances of success for trusted computing elsewhere, since it will limit the possible economies of scale, and potentially cause major interoperability issues.

7. Mobile issues

- The **Mobile Phone Working Group** (**MPWG**), one of a number of platform-specific working groups within the TCG, works on the extension and adoption of trusted computing concepts for the mobile device.
- The group builds on existing specifications and concepts to address specific characteristics of mobile devices, such as:
  - connectivity; and
  - limited capability.
- Have defined a Trusted Mobile Platform (TMP).

Possible mobile-specific issues

- As we now outline, the mobile specifications rely on an even more complex PKI.
- The realisability of this PKI in practice seems rather questionable.

TMP stakeholders I

- **Users**, who store their data in the platform:
  - There may be multiple user stakeholders in a platform;
  - For example, an employee or a consumer may be a user stakeholder.
- **Service providers**, who provide services consumed in a platform:
  - There may be multiple service provider stakeholders in a platform;
  - Examples of services include: corporate services for employees; content distribution services for consumers; an address book; a diary.

TMP stakeholders II

- **Communications carriers**, who are specialist service providers providing cellular radio access for the platform:
  - There may be multiple communications carrier stakeholders in a platform.
- The **device manufacturer**, who provides the internal communications within a platform and typically provides all the hardware resources within a platform:
  - There is a single device manufacturer stakeholder in a platform.

A TCG TMP

- Conventional TCG-enabled platforms enforce:
  - the rights of a single platform Owner (who has exclusive control over the data protection mechanisms in the platform); and
  - the rights of multiple data owners (who use the data protection mechanisms, with permission from the platform Owner).
- If a cellular-radio enabled platform was just a conventional TCG-enabled platform, it follows that an Owner or User who turned off the platform TPM would prevent the radio from operating.

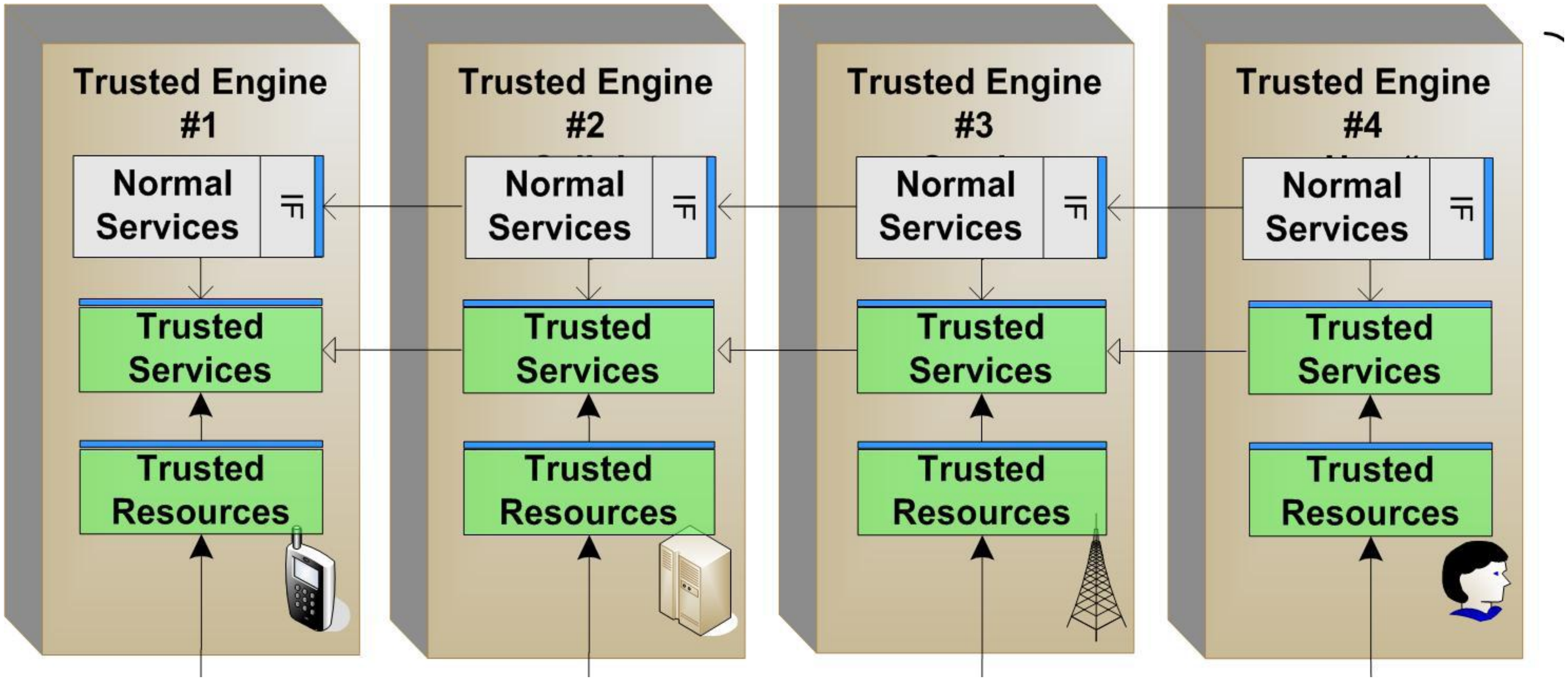
A TCG TMP

- To maintain the right of an Owner or user to turn off his TPM, the TMP specification generalises the concept of a platform to mean a set of trusted “engines”.
- A TMP, as defined by the TCG, is made up of a set of such engines.
- An engine is defined as a construct capable of:
  - manipulating data;
  - providing evidence that it can be trusted to report the current state of the host platform; and
  - providing evidence about the host platform's current state.
- Each stakeholder on a trusted mobile platform has its own engine.

A TMP engine

- Each engine provides platform services on behalf of its stakeholder, and also incorporates functionality similar in many ways to a ‘traditional’ TCG trusted platform.
- An engine can:
  - access a set of trusted resources;
  - obtain and use an endorsement key and/or attestation identity keys;
  - provide evidence of its trustworthiness as a trusted platform;
  - report evidence regarding its current state;
  - import and/or export services, shielded capabilities and protected functionality;
  - implement arbitrary software functionalities such as trusted and/or normal services.

A TCG TMP



Trusted Engine #1
Normal Services
IF
Trusted Services
Trusted Resources

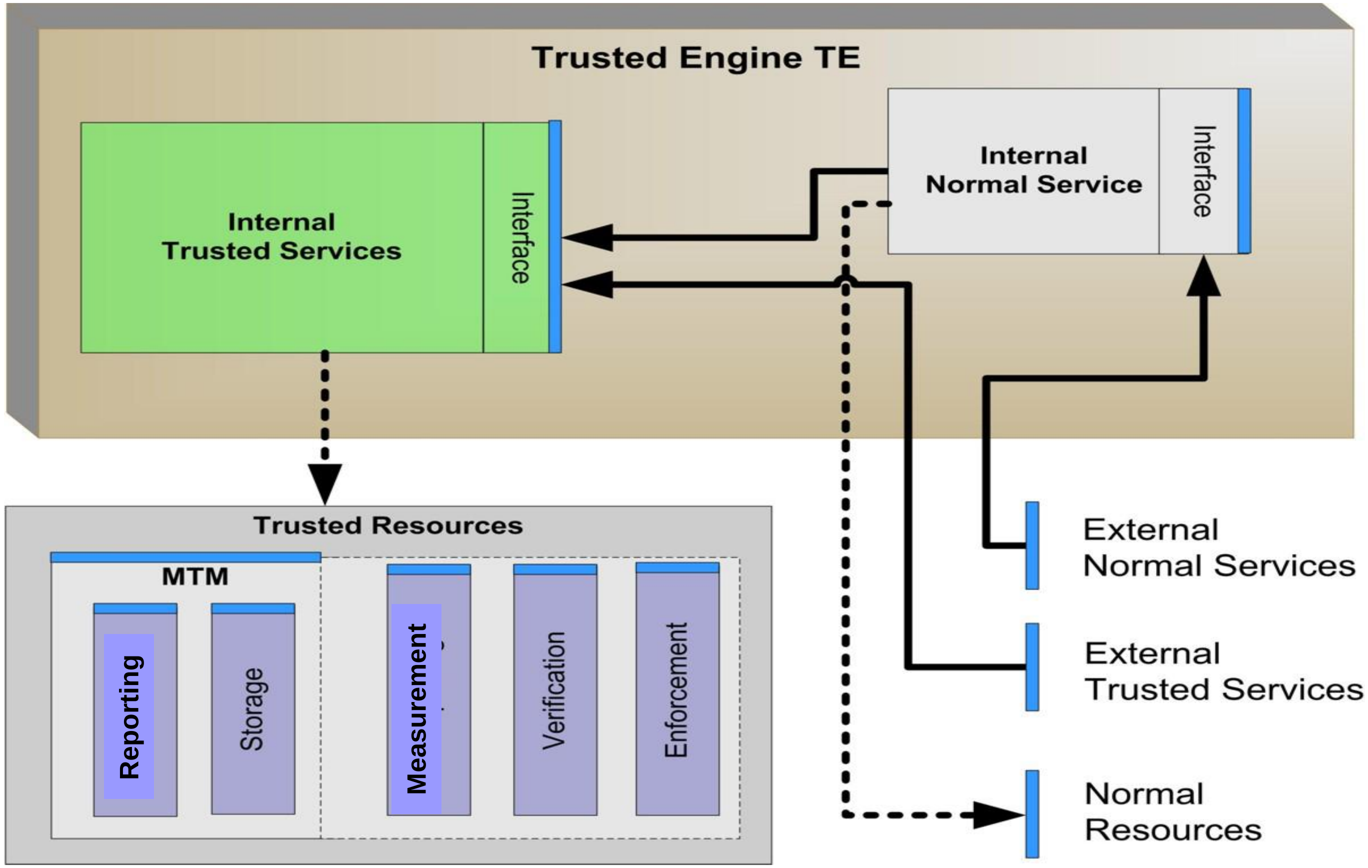
Trusted Engine #2
Normal Services
IF
Trusted Services
Trusted Resources

Trusted Engine #3
Normal Services
IF
Trusted Services
Trusted Resources

Trusted Engine #4
Normal Services
IF
Trusted Services
Trusted Resources

[Schmidt, Kuntze and Kasper]

A TMP engine – overview



[Schmidt, Kuntze and Kasper]

[Just as a TPM provides trusted resources (reporting and storage) to a PC platform, a Mobile Trusted Module (MTM) provides trusted resources to a mobile platform]

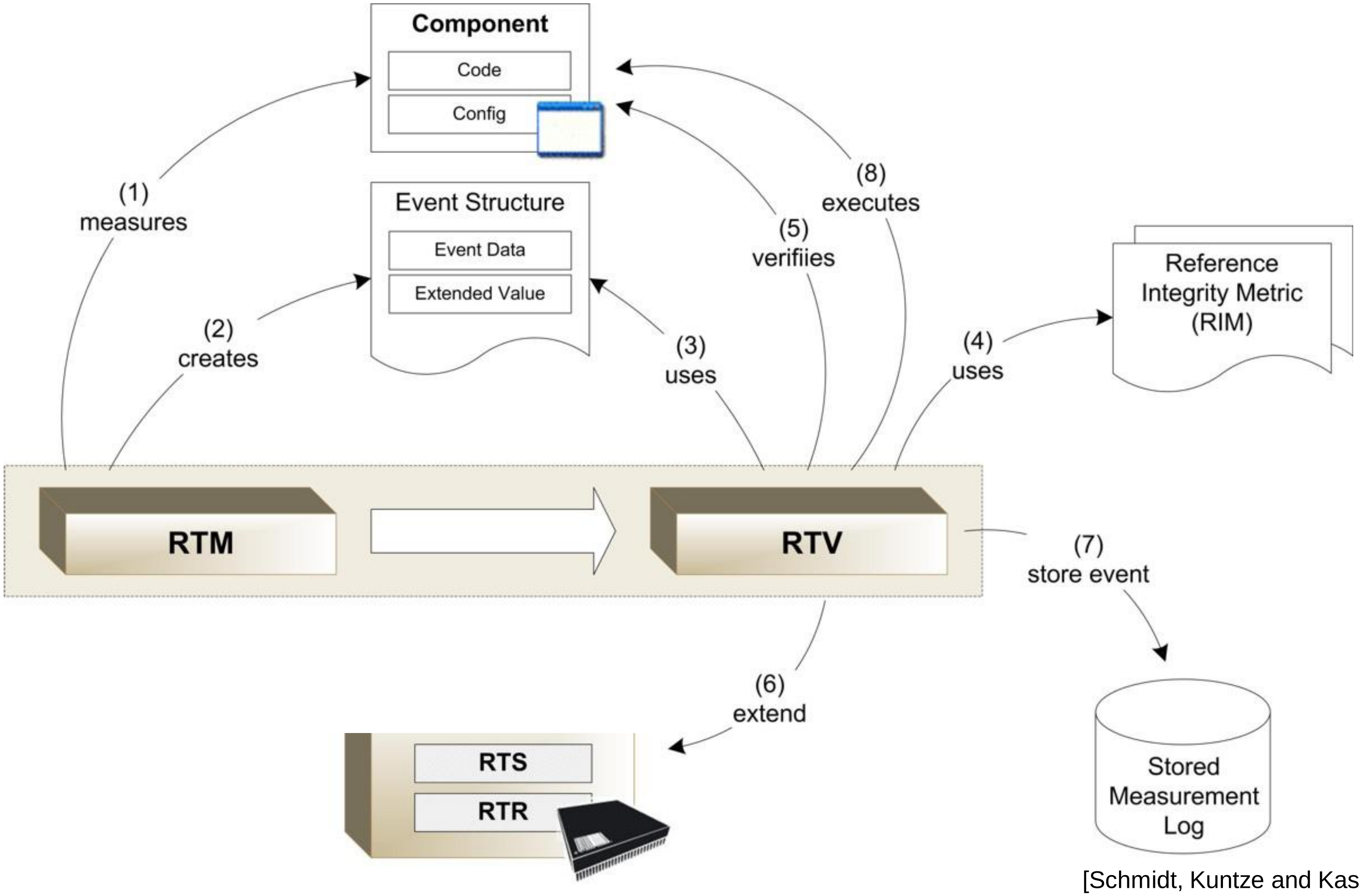
A TMP engine – components

- An engine is made up of trusted resources:
  - The following Roots of Trust are defined for the mobile domain:
    - Root of Trust for Storage (RTS);
    - Root of Trust for Reporting (RTR);
    - Root of Trust for Measurement (RTM);
    - **Root of Trust for Verification (RTV);**
    - **Root of Trust for Enforcement (RTE)**.
  - Each root of trust Provides evidence of its trustworthiness:
    - directly, by proving knowledge of secrets (EK, AIK) and associated credentials that can only be accessed by authenticated subjects of the stakeholder; or
    - indirectly by providing measurements.

Secure boot

- The TCG MPWG has defined a secure boot process.
- Rather than measuring and recording – as is the case with authenticated boot – a secure boot process allows each platform component to be:
  - **Measured**,
  - **Verified** (by RTV), in which the measured value is compared against a reference value (which indicates what the measurement 'ought to be'), and
  - **Acted upon**, where, if it is discovered that a platform's component measurement is not what is 'ought to be', then the boot process can be aborted (by the RTE).

Secure boot
The process (mandatory engines)



[Schmidt, Kuntze and Kasper]

Secure boot
RIMs and TIMs

- A Target Integrity Metric (TIM) is the actual measurement of a software component taken by the RTM or measurement agent
- A Reference Integrity Metric (RIM) is a reference value used to compare with a TIM.
- A RIM provisioning method needs to:
  - authenticate source;
  - verify authorisation of source to provide RIMs;
  - verify integrity, freshness and validity of RIMs.
- A RIM_Cert is an authenticated and integrity protected structure containing a RIM and some auxiliary information.

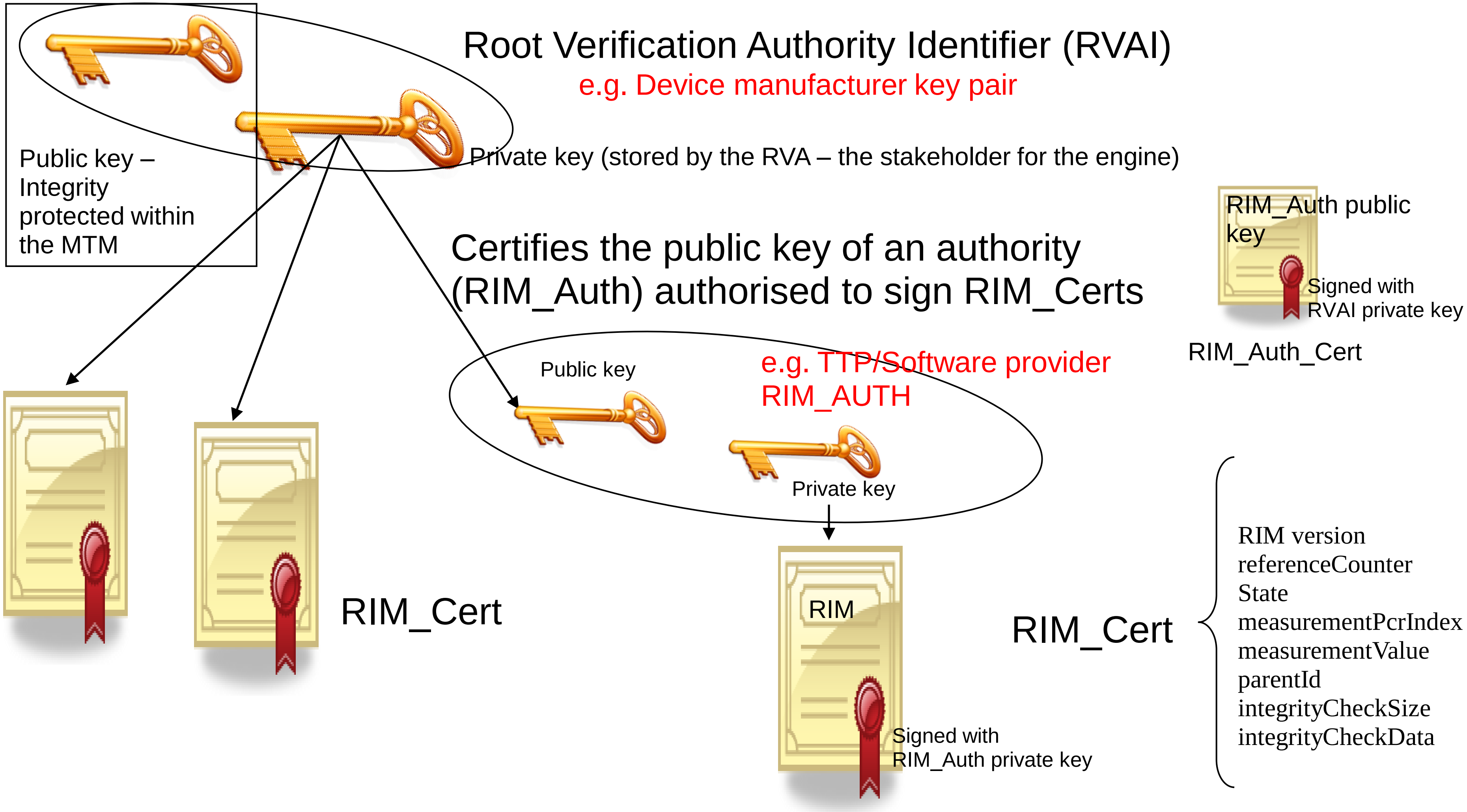
Secure boot
RIM creation/verification

- The parties which create authentic and authorised RIM_Certs are called RIM_Auths:
  - An external RIM_Cert may be authenticated using a digital signature;
  - An internal RIM_Cert may be authenticated using a message authentication code.
- The keys used to verify RIM_Certs are called TPM verification keys.
- For each MTM, a key hierarchy used to authorise RIM_Certs can be set up.

Secure boot
Supporting PKI

- Each engine must be pre-configured with a public key called the Root Verification Authority Identifier (RVAI).
- This key must be integrity protected, e.g. via ROM or NV storage in the MTM (it can also be signed using a key stored within the MTM).
- The Root Verification Authority (who owns the private key) is the stakeholder for the engine. It acts as the root CA, and can:
  - directly sign RIM_Certs;
  - delegate the authority to sign RIM_Certs to RIM_Auths in the form as RIM_Auth_Certs;
  - delegate the authority to authorise RIM_Auths in the form of RIM_Auth_Certs.

Secure boot – RIM provisioning (external RIM_Certs)



Root Verification Authority Identifier (RVAI)

e.g. Device manufacturer key pair

Public key – Integrity protected within the MTM

Private key (stored by the RVA – the stakeholder for the engine)

Certifies the public key of an authority (RIM_Auth) authorised to sign RIM_Certs

RIM_Auth public key

Signed with RVAI private key

RIM_Auth_Cert

Public key

e.g. TTP/Software provider RIM_AUTH

Private key

RIM_Cert

RIM

Signed with RIM_Auth private key

RIM_Cert

RIM version
referenceCounter
State
measurementPcrIndex
measurementValue
parentId
integrityCheckSize
integrityCheckData

RIM_Auth_Cert

tag	TPM_TAG_VERIFICATION_KEY
usageFlags	defines the capabilities for the key defined in key data, i.e. whether it can authorise RIM_Auths or sign RIM_Certs
parentID	Identifier for the key used to authorise the key contained in keyData If parentID = none then this is a “root key”
myID	Identifier for the key structure
referenceCounter	Defines the validity of the structure
keyAlgorithm	Identifier for the algorithm to be used with the key held in keyData
keyScheme	The method by which the integrityCheckData can be verified
extensionDigestSize	Length in bytes of the buffer extensionDigest
extensionDigest	Contains a hash of proprietary extension data
keySize	Length of the buffer KeyData
keyData	Contains the key for verifying the integrityCheckData field
integrityCheckSize	The length of the integrityCheckData buffer
integrityCheckData	An integrity check for the TPM_Verification_Key The method by which to verify is defined in the object referenced by parentId

RIM_Auth_Validity_Lists

- If the Root Verification Authority (or other RIM_Auth acting as a CA) wishes to revoke delegated authorisation, then it SHOULD do so by signing a periodic RIM_Auth_Validity_List indicating the identifiers of its delegates that are still valid.
- Every RIM_Auth which signs Validity Lists MUST ensure it always has signed a Validity List whose “valid from” and “valid to” fields in UTCtime format enclose the current date and time.
- Whether or not a RIM_Auth signs RIM_Auth Validity Lists MUST be indicated by a usage flag in the TPM_Verification_Key structure.

RIM_Cert

tag	TPM_TAG_RIM_CERTIFICATE
rimVersion	A version number for the RIM certificate
referenceCounter	Defines the validity of the structure
state	Defines the state the system must be in at the time of use
measurementPcrIndex	The PCR index that is to be extended using the defined measurement value
measurementValue	The measurement value to be extended to the specified PCR
parentId	The identifier for the key used to verify this structure
extensionDigestSize	Length in bytes of the buffer extensionDigest
extensionDigest	Contains a hash of proprietary extension data
integrityCheckSize	The length of the integrityCheckData buffer
integrityCheckData	An integrity check for the TPM_RIM_CERTIFICATE The method to be used to verify it is defined in the object referenced by parentId

RIM_Validity_Lists

- RIM_Auths that are able to sign RIM Certificates SHOULD be able to revoke such certificates (typically also issuing a replacement).
- If a RIM_Auth is able to revoke its RIM_Certs, then it SHOULD do so by signing a periodic RIM_Validity_List indicating the serial numbers of its certificates that are still valid.
- Every RIM_Auth which signs Validity Lists MUST ensure that it signs Validity Lists whose “valid from” and “valid to” fields (in UTCtime format) enclose the current date and time.
- Whether or not a RIM_Auth signs RIM Validity Lists MUST be indicated by a key-usage flag in the TPM_Verification_Key structure.

Internal and external RIM_Certs

- The full set of external RIM_Certs, RIM Validity Lists, RIM_Auth_Certs and RIM_Auth_Cert revocation information (RIM_Auth Validity Lists etc.) defines a complex privilege structure.
- It is not required that each verification agent (especially the RTV) is able to process this whole structure during each boot and hence determine what is really a valid RIM.
- This problem is addressed by using a special RIM Conversion Agent to process all of the external RIM_Certs and map from External RIM_Certs to Internal RIM_Certs.
- These Internal RIM_Certs can then be more easily handled by the RTV and other verification agents.

Acknowledgements

- I would like to thank all the partners in the OpenTC project who helped us develop a wide range of teaching materials.
- I must particularly thank Eimear Gallery, who worked on OpenTC almost from the beginning to the end, for preparing a significant proportion of the material presented in this lecture (along with Shane Balfe).

The Open-TC project was co-financed by the EC (11/05-4/09).

If you need further information, please visit the 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

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.