Trusted Computing as a universal security infrastructure

Chris Mitchell
Information Security Group
Royal Holloway, University of London
http://www.isg.rhul.ac.uk/~cjm

Acknowledgements

- The background material on TC was mostly produced as part of the OpenTC project by Eimear Gallery.
- The ideas on TC GAA and an OTP system based on GAA are joint work with Chunhua Chen and Shaohua Tang (South China University of Technology).

Contents

Background I

- We need to protect data in mission critical information systems, and to maintain and improve confidence in use of the Internet.
- Conventional secure computing is too difficult to use and too expensive.
- **Trusted Platforms** are computer platforms optimised for the protection and processing of private and secret data.

Background II

- Trusted computers have isolated environments that restrict access to the data in those environments.
- Data held within such an environment is not necessarily secure, but can guarantee that only the applications in the same environment can access the data.
- **Trusted Computing is a fundamental change to computers and computing.**

Trusted Computing

- 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.
- Windows Vista and 7 incorporate support for these chips, and use them as the basis of certain novel security functions.
- Open source software also exists capable of exploiting this hardware.
- However, the full potential remains to be exploited.

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.

Basic components and services

- **Integrity measurement** – cryptographic hash of a platform component (software executing on 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.

Fundamental requirements

- First we need 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 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 know 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).

Trusted platform components

- The TCG has specified platform components required in order to implement:
  - Integrity measurement;
  - Authenticated boot;
  - Sealed storage;
  - Attestation.
- Of fundamental importance are the three **Roots of trust**: 'components that must be trusted if the platform is to be trusted':
  - Root of trust for measurement (RTM);
  - Root of trust for storage (RTS);
  - Root of trust for reporting (RTR).

Roots of trust: RTM I

- The RTM is a computing engine which accurately generates integrity measurements representing a software component running on the platform.
- The measurement (a hash digest) is then recorded to a platform configuration register (PCR) in the TPM.
- Details of what was measured, i.e. of the software component, are then recorded to the stored measurement log (SML) outside the TPM.

Roots of trust: RTM II

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

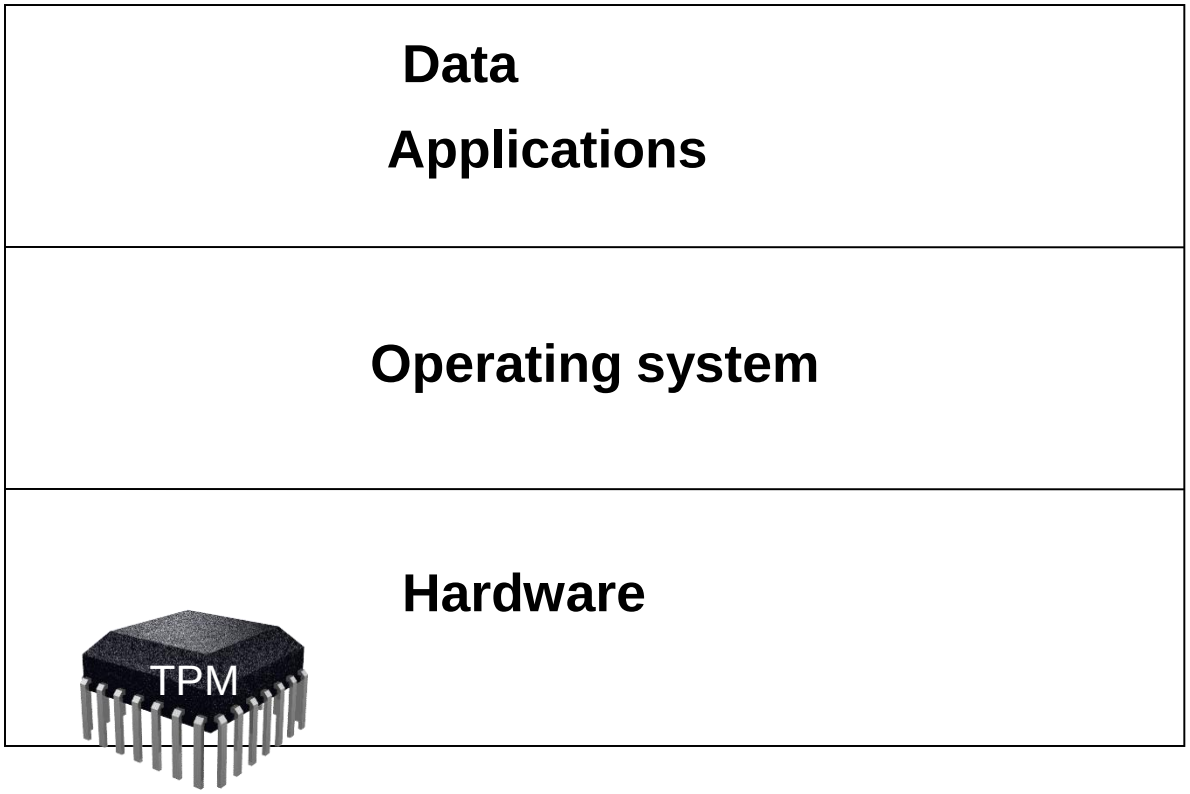
Roots of trust: RTS/RTR

- The RTS is a collection of capabilities which must be trusted if data stored inside a platform is to be trusted.
  - The RTS provides integrity and confidentiality protection to data used by the TPM, although data itself 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 RTS and RTR constitute the minimum functionality that should be provided by a **Trusted Platform Module (TPM)** – implemented as a chip bound to the platform.

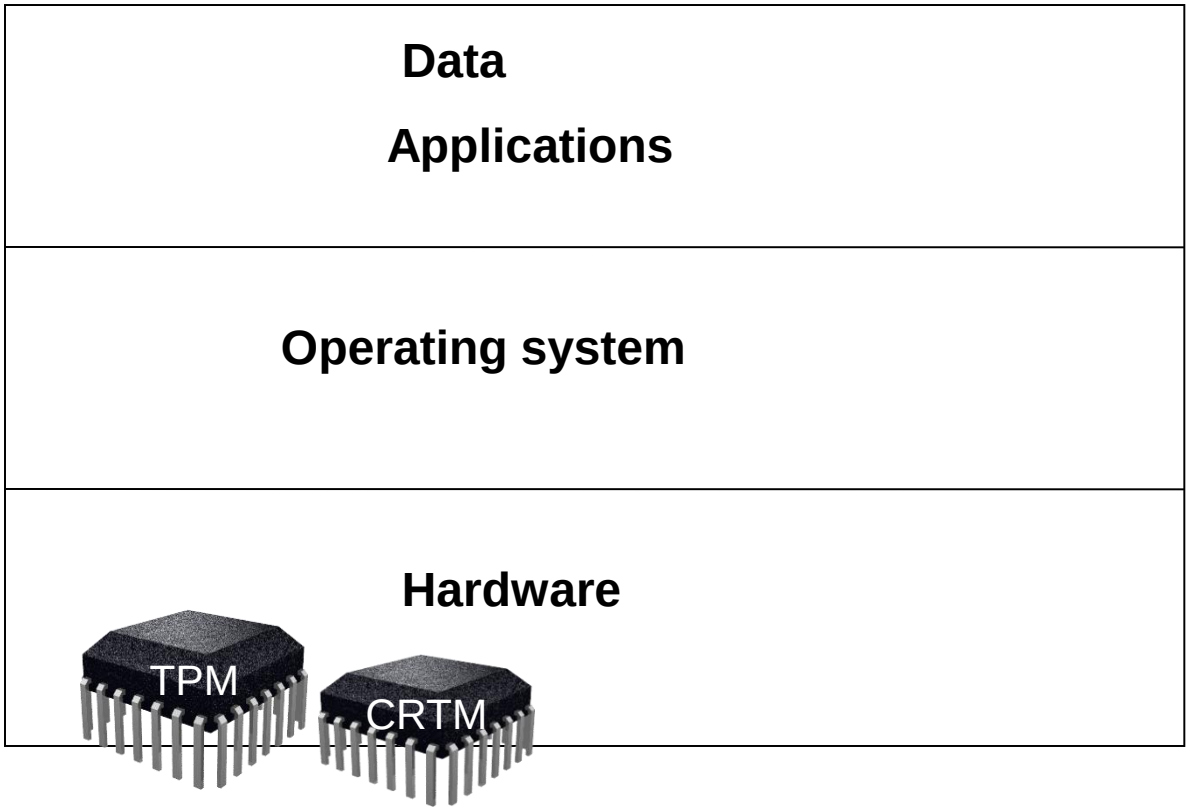
The TSS

- 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 resources to support access to the TPM:
  - the TPM Device Driver;
  - TPM Core Services;
  - TPM Service Provider.

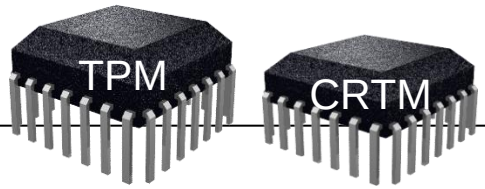
Current platforms with integrated TPMs



Envisaged trusted platforms (stage 1)

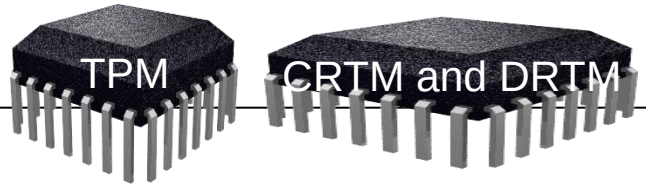


Envisaged trusted platforms (stage 2)

Data-1 **Application-a** **Application-b** **Guest OS**	**Data-2** **Application-c** **Application-d** **Guest OS**	
Virtual machine monitor/ Hypervisor/ Isolation layer		
Hardware 		

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
- Core trusted computing functionality
- Security infrastructures
- TC GAA – a universal security infrastructure
- Applying TC GAA
- Conclusions

TCPA I

- TCPA (Trusted Computing Platform Alliance): industry working group.
- Focus: enhancing trust and security in computing platforms.
- Original alliance of promoter companies (HP, IBM, Intel and Microsoft) – founded in 1999.
- Initial draft standard unveiled: late 1999.
- Invitation then extended to other companies to join the alliance.

TCPA II

- TCPA released first specifications in early 2001, defining a fundamental component of a trusted platform, namely the Trusted Platform Module (TPM).
- A TPM is typically implemented as a chip mounted on a PC motherboard, and provides a foundation for all trusted functionality on the PC (in combination with the BIOS).
- By 2002 the TCPA had over 150 member companies.

The TCG

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

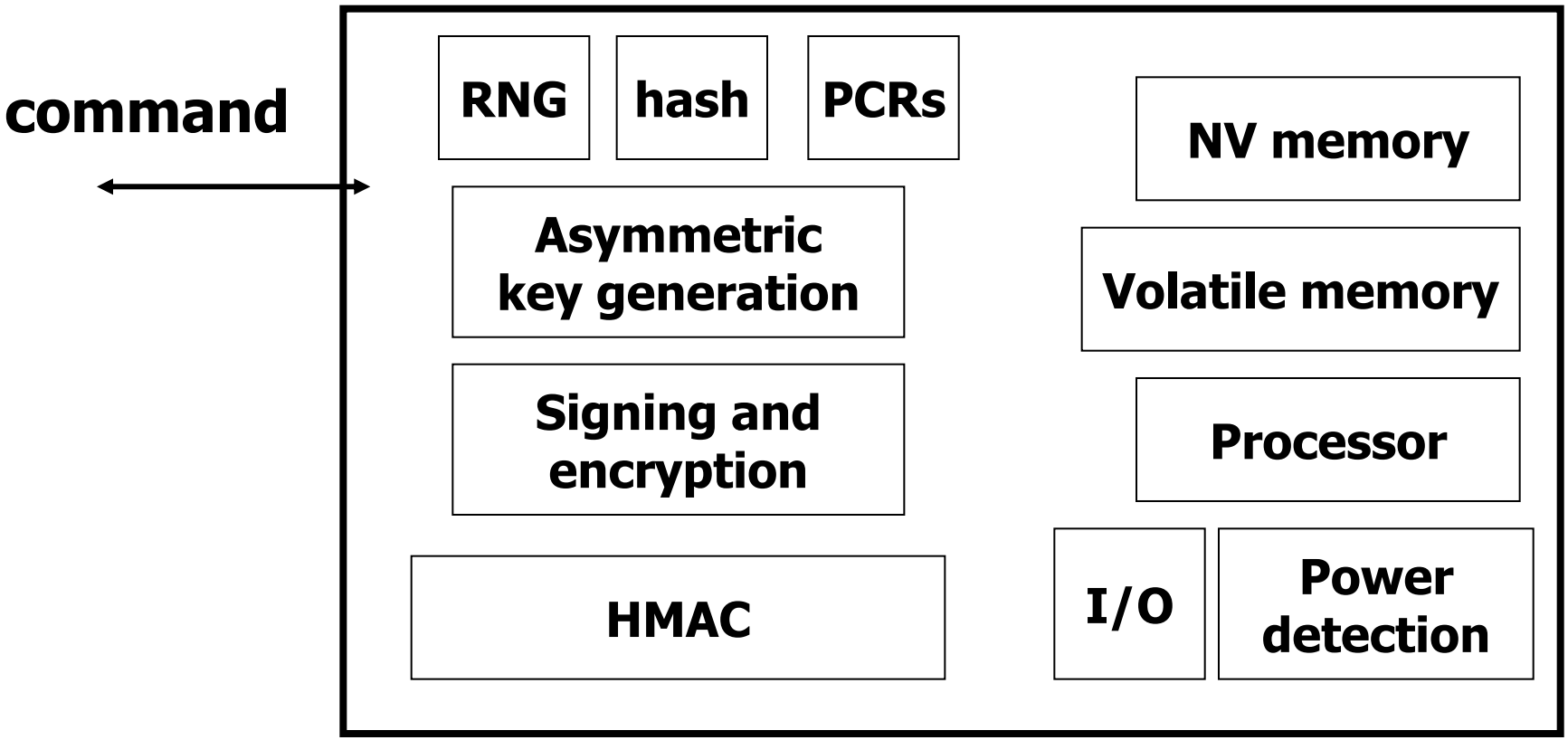
TCG specifications

- The 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 (May 2009) been adopted as an international standard: ISO/IEC 11889 parts 1-4 (with the title *Information technology - Trusted Platform Module*).

Contents

- What is trusted computing?
- The TCG
- Core trusted computing functionality
- Security infrastructures
- TC GAA – a universal security infrastructure
- Applying TC GAA
- Conclusions

The TPM


command
RNG
hash
PCRs
NV memory
Asymmetric key generation
Volatile memory
Signing and encryption
Processor
HMAC
I/O
Power detection

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 engine;
  - HMAC engine;
  - SHA-1 engine;
  - 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 new TPM specification in the next few 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 (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** (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** (an asymmetric encryption key pair used to support secure storage of data external to the TPM).

Endorsement Key Pair (EK)

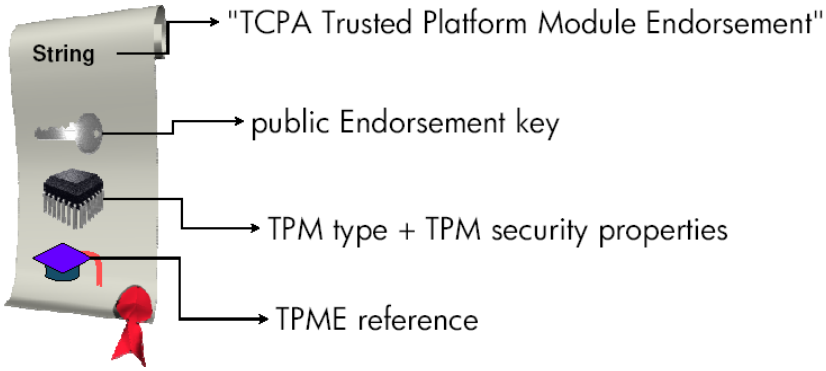
- It is a fundamental requirement that:
  - Each TPM has an endorsement key pair stored in it;
  - The public part of the endorsement key pair is certified by the TPME (e.g. the TPM manufacturer) in the form of the endorsement credential.
- The private part of the EK is used by a TPM to prove that it 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 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;
  - is signed by a Trusted Platform Management Entity.

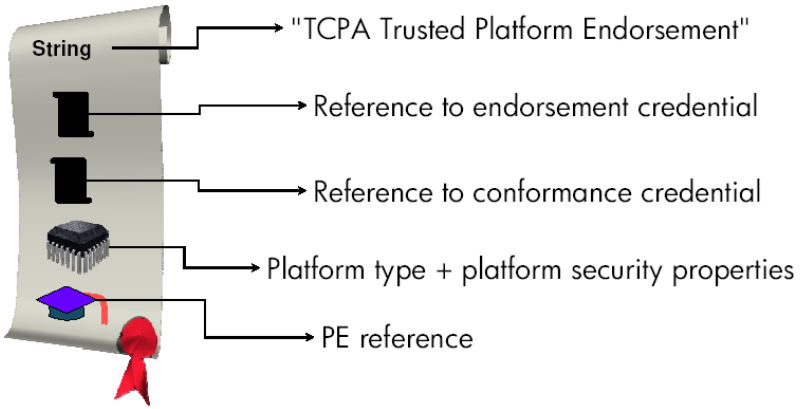


Credentials II

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



Attestation Identity Key Pairs (AIKs)

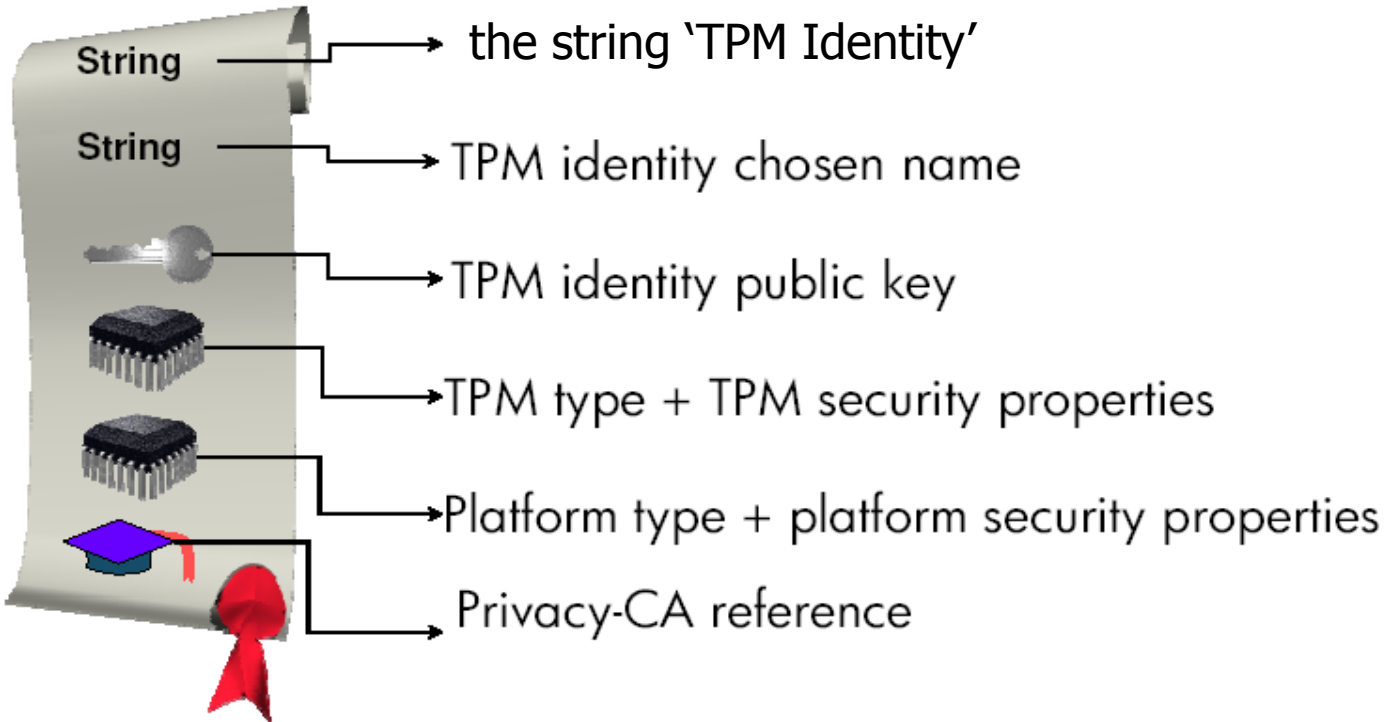
- These signature key pairs are used by a TPM to attest to platform properties to external entities.
- Used by a ‘challenger’ of the platform to verify that a TPM is indeed genuine, without identifying a specific TPM.
- A special trusted third party called a Privacy-Certification Authority (P-CA) supports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

Generation of AIKs

- TPM chooses a new AIK pair, an ‘identity’, and a P-CA which will 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 chosen 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 EK.
- The encrypted certificate and key are then sent back to the requester, thus ensuring that only the appropriate TPM can access the certificate.

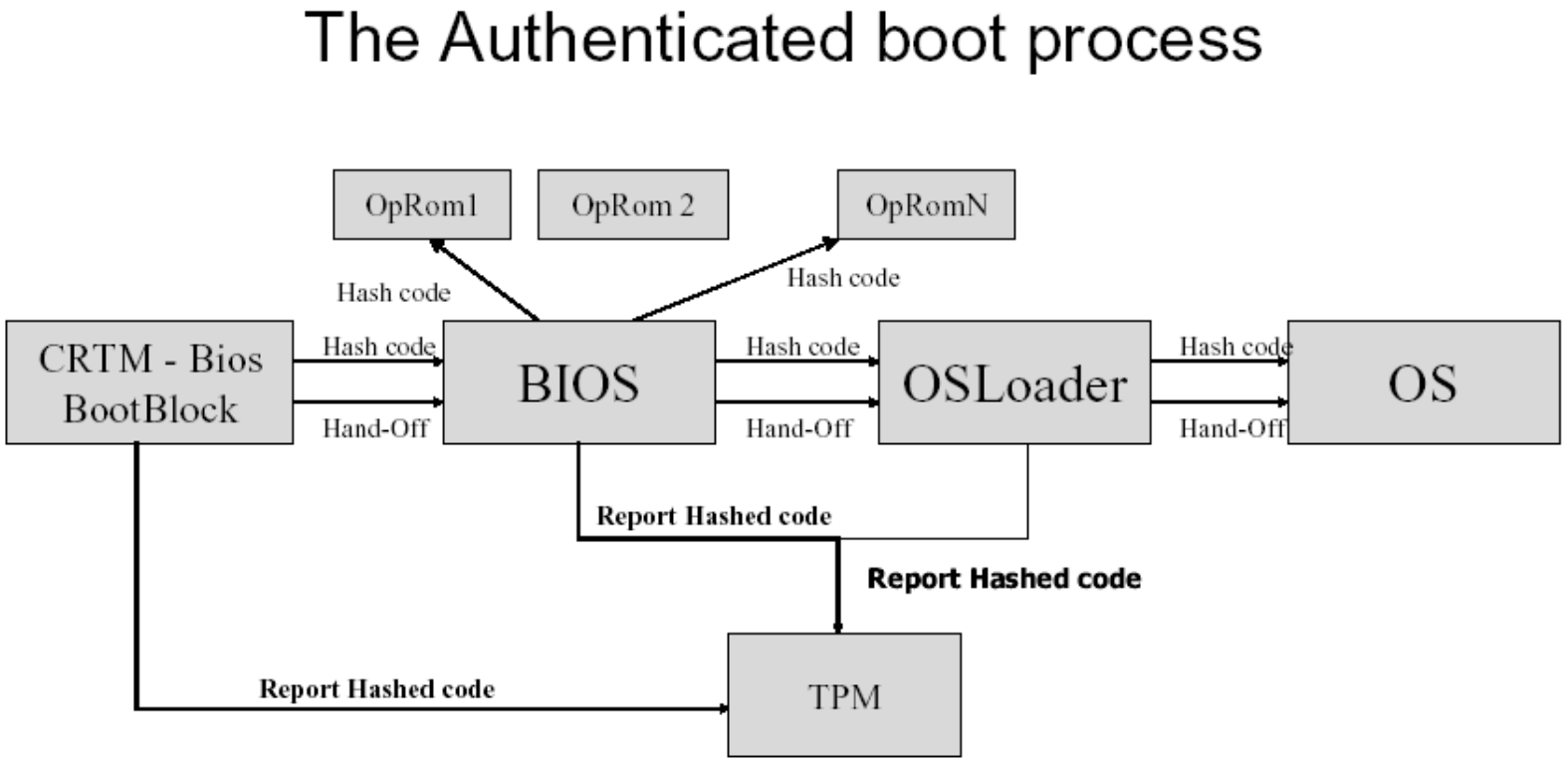
Issues with use of a P-CA

- 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.

Direct Anonymous Attestation (DAA)

- 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 need 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.

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 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 from boot time:
  - 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 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.

Contents

Security infrastructures

- In order to use cryptography to protect communications, some kind of security infrastructure needs to be in place.
- In its simplest form, this will just be a means to set up shared secret keys between communicating parties.
- Traditionally, e.g. in banking networks, this can be achieved using one or more Trusted Third Parties (TTPs).
- One type of TTP for this purpose is known as a Key Distribution Centre (KDC).
- A KDC shares a secret key with every party, and these keys can be leveraged (using an appropriate protocol) to set up a secret key between any two parties.

Public Key Infrastructures (PKIs)

- A PKI is another type of security infrastructure, based on digital signatures.
- A Certification Authority (CA) creates digitally signed certificates for user public keys, binding a user name to a public key.

The promise of a universal PKI

- A few years ago, PKI was the subject of huge hype.
- Companies producing PKI products (e.g. CA software) or providing PKI services suddenly (and temporarily!) became hugely valuable.
- In many cases the vision sold as part of this hype was of some kind of universal PKI, whereby every PC in the world would have a public key certificate, which could then be used for a huge range of purposes, e.g.:
  - secure e-commerce;
  - universal secure e-government;
  - secure home banking;
  - electronic signatures for all;
  - …

PKI – what happens in practice I

- Of course, this has not happened.
- There are many PKIs, each set up for a specific purpose.
- For example:
  - companies have their own PKIs, used to support internal secure communications;
  - MasterCard and Visa (and card issuing banks) have PKIs set up to support EMV (used to support smart card based credit/debit card transactions, e.g. in parts of Europe);
  - Internet web sites have certificates used for SSL/TLS security.
- There are, of course, many explanations for this – one being the fact that the policies under which certificates are issued will depend on the context of use.

PKI – what happens in practice II

- More generally, PC users do not have the expertise or motivation to generate a signature key pair, and obtain a certificate for their public key.
- This can be seen from the failure of the SET e-commerce secure payment system, one of the major obstacles to the adoption of which was the need for every user to generate a key pair, and take a copy of their public key to their bank.
- End users cannot be expected to understand the operation of public key cryptography.
- Moreover, current PCs typically do not have a means for secure key storage (needed for the private key).

Contents

- What is trusted computing?
- The TCG
- Core trusted computing functionality
- Security infrastructures
- TC GAA – a universal security infrastructure
- Applying TC GAA
- Conclusions

TC – a universal security infrastructure?

- Trusted computing may give us a universal security infrastructure ‘by the back door’.
- Every PC owner will have a crypto-capable PC, will have an asymmetric key pair in their TPM, and will have a public key certificate for the public key.
- Moreover, the TPM is capable of generating signature key pairs on demand, of using generated private keys to sign arbitrary data, and of providing secure storage for private keys.

Possible problems

- The key pair provided in every TPM (when shipped to user) is not suitable for use as a general purpose key pair:
  - although it is an RSA key pair, it is intended for use as an encryption/decryption key pair;
  - the TPM does not enable its use for signing arbitrary data.
- The TPM is capable of generating an RSA key pair designed for signing (known as an AIK – Attestation Identity Key), and can also obtain an X.509 certificate for the public part of the AIK from an entity known as a Privacy-CA.
  - However, the private part of the AIK cannot be used to sign arbitrary data.

Solutions to problems

- Get the TPM to generate another signature key pair, and use an AIK to sign a ‘certificate’ for the public key.
- The private key of this key pair can be used to sign arbitrary data.
- This means that the PC now has a means of generating arbitrary numbers of signature key pairs (essentially automatically) and obtaining certificates for them.
- Only problems are:
  - There is a need to associate two certificates with each key pair (the Privacy-CA certificate for the public AIK, and the AIK-signed certificate for the public key in use);
  - The AIK-signed certificate is not in the standard (X.509) format.

Certificate 'translation'

- A means of addressing these last two problems has been proposed by the TCG.
- Proposed special extension to PKCS#10 (PKCS#10 is a format for submitting certification requests to a CA).
- This extension (SKAE) allows a PC to submit a PC-generated certificate (signed using AIK) for signature public key, with other evidence, as part of a cert request.
- CA verifies the certificate and evidence, and would then generate a new certificate for the PC public key.
- All these processes could be performed by a Java applet, which would give the PC user a secure and automatic means of joining a global PKI.

Example 1 : SSL client side authentication

- Currently, SSL is only used for unilateral authentication i.e. of the server to the client, mainly because client PCs typically do not have key pairs and certificates.
- Precisely the procedure just described could give a means for a PC user to acquire a signature key pair and a public key certificate in order to support SSL client side authentication.
- This is described in greater detail in:
  - A. Alsaid and C. J. Mitchell, 'Preventing phishing attacks using trusted computing technology', in Proc. INC 2006, 6th International Network Conf., Plymouth, July 2006, pp.221-228.
- Related work, including implementations, has been conducted by the OpenTC project.

Example 2: Secure PC-based electronic signatures

- A considerable amount of work has gone into developing legislative and commercial frameworks for electronic signatures.
- However, such frameworks typically require a cumbersome registration procedure for users, and also some means of storing private keys securely.
- The possibility exists that, with the aid of the TPM in a PC, the PC itself can become a trusted platform for the implementation of a personal electronic signature capability, since it can provide the secure storage and also automatically perform the registration procedures.

Portability and privacy issues

- The problem remains that PCs are not typically in one-one correspondence with human users.
- Users have multiple PCs (transferring secrets between TPMs is difficult), and PCs may have multiple users.
- In the latter case. issues may arise in holding a single user accountable for the behaviour of a PC.
- However, TPMs are ‘owned’ by a single user, which typically means that only one individual will be able to use the TPM-protected keys.
- If users want multiple ‘unlinkable’ identities, TPM can generate new key pairs. (Privacy-preserving certification and use of cryptography is key feature of TCG specs.).

Using the TC infrastructure

- It is perfectly possible to design applications building directly on the trusted computing infrastructure.
- Substantial literature now exists.
- However, secure application protocols are non-trivial to design.
- Trust relationships can be very unclear.

Third party support

- We propose the creation of a third party based service to enable the provision of security services building on the TC infrastructure.
- The definition of standard security services, e.g. for key establishment, will enable the TC infrastructure to be exploited in a simple and uniform way.
- It will also provide an opportunity for trusted computing aware third parties to provide novel security services.

Using the GAA architecture

- The *Generic Authentication Architecture* (GAA), standardised by 3GPP, builds on the mobile phone security infrastructure to provide generic security services including authenticated key establishment.
- We have designed a version of GAA (which we call **TC GAA**) which enables TC to be used to provide generic security services in a simple and uniform way.

What GAA provides

- GAA has two main protocols:
  - **Bootstrapping** – establishes a secret master key between a mobile phone and a **BSF** (a TTP provided by mobile operator)
  - **Session key establishment** – enables an application-specific session key (derived from the master key) to be established between the phone and a GAA-aware server (which communicates with the BSF).

TC GAA – a sketch

- A Privacy CA takes the role of the BSF in **TC GAA**.
- A simple extension to Privacy CA functionality enables it to:
  1. perform bootstrapping, to set up a shared master key with a TPM;
  2. use this master key to derive application-specific session keys which it provides to specific servers.

GAA as a general framework

- GAA was originally designed to provide a way of exploiting the mobile phone security infrastructure.
- We have shown how it can be used to build on the TC infrastructure.
- Could also be used as a framework for providing general purpose security services building on other pre-existing security infrastructures.

Contents

- What is trusted computing?
- The TCG
- Core trusted computing functionality
- Security infrastructures
- TC GAA – a universal security infrastructure
- Applying TC GAA
- Conclusions

GAA-based one-time passwords I

- We consider one possible application of TC GAA, namely to enable the simple derivation of one-time passwords (OTPs).
- These passwords are based on a (potentially weak) long-term user password.
- The TC GAA session key provides protection against brute force password searches.

GAA-based one-time passwords II

- The OTP is computed as a function of the long-term user password and the short term application-specific session key.
- Compromise of the OTP does not enable a brute-force search for the password without knowledge of the session key.
- The TP used in the protocol does not need to be registered to the user – only needs to be trusted not to compromise the password.

GAA OTP – other instantiations

- The notion of using a GAA session key to help generate an OTP from a long-term weak password applies to all instantiations of GAA.
- Indeed, in parallel work we have designed a series of simple OTP schemes using a GAA-enabled mobile phone.

Contents

- What is trusted computing?
- The TCG
- Core trusted computing functionality
- Security infrastructures
- TC GAA – a universal security infrastructure
- Applying TC GAA
- Conclusions

Building the TC infrastructure

- There is a major problem with rolling out trusted computing applications.
- The envisaged complex infrastructure does not yet exist.
- TC GAA may help with providing the business case necessary for the emergence of the wide range of third party security services necessary to fully realise the goals of trusted computing.

TPM.next

- The scheme we have proposed is built on the current generation of TPM (v1.2) functionality.
- A new set of TPM specifications (with working name TPM.next) is due to be released shortly.
- Whilst backwards compatible, these allow a richer range of functions, and may make certain tasks simpler.