Trusted Computing:
The TCG Trusted Platform Module Specification

Embedded Systems 2004

New Munich Trade Fair Centre in Munich, Germany

Hans Brandl, hans.brandl@infineon.com
Infineon Technologies AG
Trusted Computing

Introduction

In recent years, the necessity, functionality and new possibilities of trusted computing, especially the new standard from the Trusted Computing group (TCG) have been a hot topic of discussion in many forums. Also a lot of fears and misgivings in special for the use in PC environments have been provoked. In the meantime this technology has come of age, initial standards have been agreed upon and PCs are now equipped with it. The next area for trusted information processing are now amongst others the field of embedded systems. What is the content of the standard, the practical usability and the real issues here?

1 Why do we need Trusted Computing

One of the as yet unresolved problems of widely used security applications is to protect the hardware platform against attacks on its integrity or modification of the security software. Within the PC area typical incidents and attacks are well known and are endangering PCs for home banking, but also on servers within companies and other organizations which are used for sensible data, like personal, billing, e-commerce and others. But also on embedded systems such incidents happen. Typical examples are illegal changes or manipulations of data in controllers within automotive systems, e.g. odometer values for increasing the car value or vehicle theft protection systems but also other embedded system which handle goods of value. Current approaches for solving this problem purely at the software level are by their very principle unpromising. As has since been amply confirmed from experience and security trends in the smart card world, a trusted and tamperproof security basis cannot be implemented using software-based solutions alone. This of course applies equally to host systems such as PC platforms as well as embedded controllers.

2 The Trusted Computing Group

Major companies in the PC sector have therefore joined forces and begun working to solve this problem with the aid of a new hardware approach and the creation of an associated industry standard. In 1999 Compaq, Hewlett-Packard, IBM, Intel und Microsoft established the Trusted Computing Platform Alliance. The aim was to create Trusted Clients (e.g. PCs, but also PDAs or mobile telephones) in order to make important applications such as networks, communications and e-commerce much more trustworthy. At the same time, however, this standard was to be kept as open as possible in order to inform the technical and interested public in good time and to create confidence. The emerging Trusted Computing Standard employs a secure hardware structure whose main component, the Trusted Platform Module (TPM), is specified as an LSI security chip. This Standard is largely based on recent years' experience with high-security smart cards and their applications, important parts of whose

architecture and security characteristics have been consistently adopted. Similarly to the way in which we use the smart card's cryptographic mechanisms to protect sensitive and confidential personal data as well as critical processes in a security environment, these functions can also be used in the TPM to ensure not only the integrity of a platform but also to protect its user data. To restate clearly:
The TCG Standard provides authentication and accreditation of the platform, **not of the user**.

2.1 Standardisation: What is TCG?

The TCG is an industry standards group with a membership of more than 80 computing companies and offers standards that span computing devices, from PCs and servers to mobile devices and peripherals.

The TCG has since agreed three important specifications:

- Trusted Platform Module (TPM)
- PC Specific Implementation Specifications
- TCG Software Stack Specifications (TSS)

These parts set out the basic prerequisites for secure components on the new secure platforms. The corresponding member companies have simultaneously been investing considerable resources to ensure that the first implementations are available and that the first trusted motherboards with TPM or complete PC systems are ready for shipment.

At the same time, however, standardization work is continuing. In a total of around 20 working groups, the next application options and interfaces are being planned and standardized.

2.2 Security aspects in the TCG Specification

The generic TCG approach is producing new system structures: whereas until now security was to be achieved by means of additional levels of encryption or antivirus software, TCG begins at the very lowest level of the platform, and here right from the start of the booting operation of such a system, the TPM being trusted a priori as a certified HW security chip. At system startup an uninterrupted "chain of trust" extends from this lowest layer up to the applications. As soon as the lower level in each case has a stable security reference, the next layer can be supported on it. Each of these domains is built upon the preceding one and can therefore expect every transaction, internal link and device connection to be trusted, reliable, secure and protected.

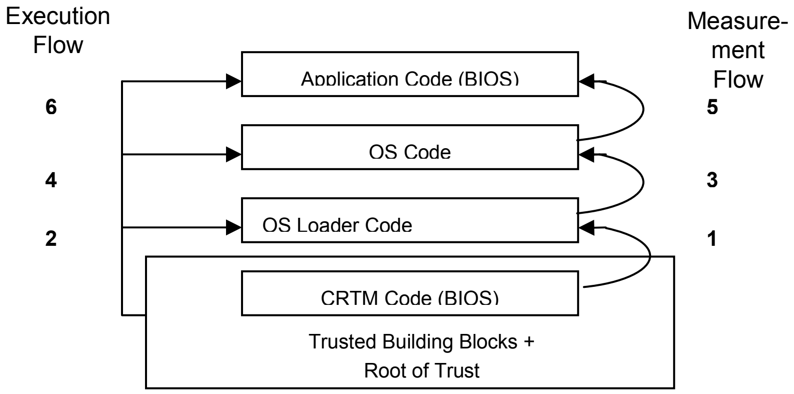


Fig. 1: Structure of the "chain of trust"

As a hardware security reference, the TPM constitutes the "root of trust" of the entire chain. Right at the start a check is performed to ascertain whether the signature (and therefore the constellation) of the platform components has changed, i.e. whether one of the components (disk storage, LAN connection, etc.) has been modified or even removed or replaced. Similar checking mechanisms supported by the TPM then successively verify e.g. the correctness of the BIOS, of the boot block and of the booting process itself, as well as the next higher layers at startup of the operating system. Throughout the startup process, but also later, the security and trust status of the system can therefore be interrogated via the TPM. However, this means that a compromised platform can also be securely identified by others and data exchange can be restricted to the appropriate extent. Trusted computing systems can create the conditions whereby for the first time modern, networked platform structures can also be significantly refined from the point of view of security and mutual trust.

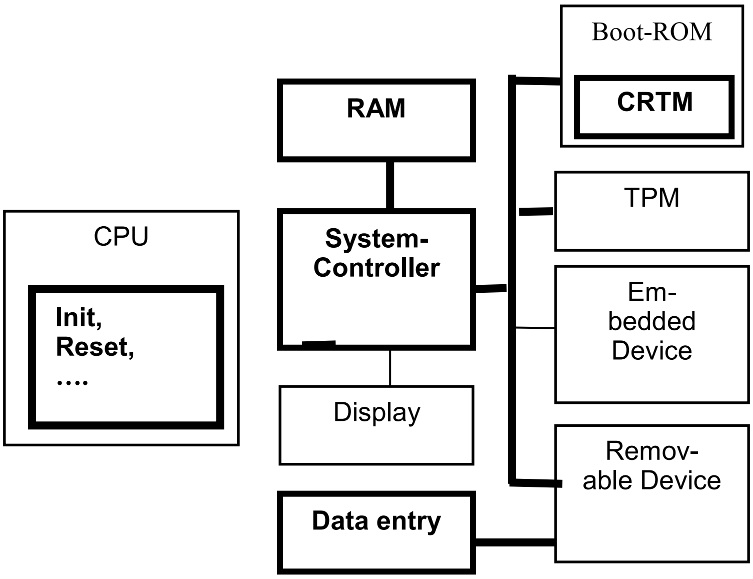


Fig. 2: Trusted platform: Trusted building blocks (core modules) accentuated

3 Background: Cryptographical and Security Mechanisms

For the realization of trustful and secure TC elements, all important functions have to use one of the usual and accepted security mechanisms. Therefore, the following background information recaptures the necessary basics of cryptographic theory and applications.

Cryptography is the science of information security. The word is derived from the Greek kryptos, meaning hidden.
Modern cryptography concerns itself with the following four objectives:

1) Confidentiality (the information cannot be understood by anyone for whom it was unintended).
2) Integrity (the information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected).
3) Non-repudiation (the creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information).
4) Authentication (the sender and receiver can confirm each others identity and the origin/destination of the information).

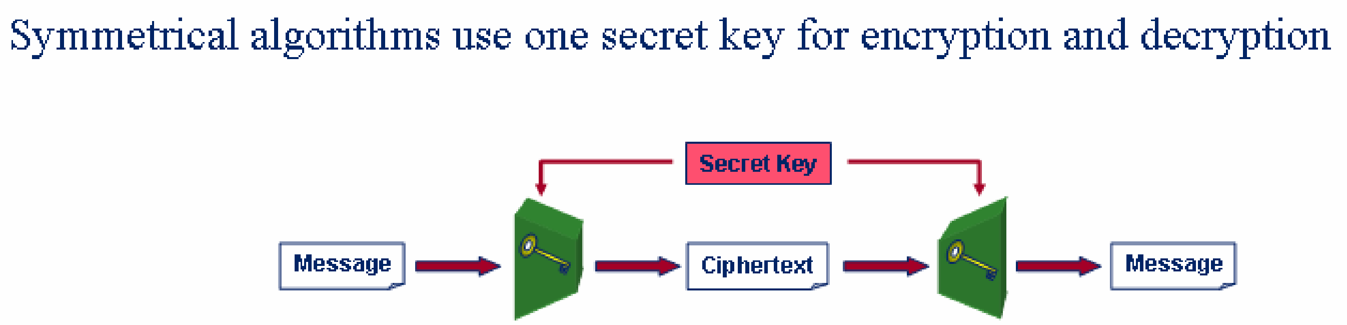
Procedures and protocols that meet some or all of the above criteria are known as cryptosystems. Cryptosystems are often thought to refer only to mathematical procedures and computer programs; however, they also include the quality of implementation and regulation of human behavior.

3.1 Cryptographical algorithms

The kernels of cryptosystems are in general cryptographic algorithms. They all perform the same basic function: They take an input – e.g. a message - and transform it with the help of special mathematical functions and a key, which controls these functions into a single output. There are two ways to perform this function. Encryption, as shown in Figure 3, uses the cryptographic key to transform the original message into an encrypted form. Decryption does the reverse; it uses a cryptographic key to transform an encrypted message back into its original (a.k.a. plaintext) form. A key is just a number, or a sequence of bits, which is large enough, so that it cannot be guessed or discovered in a systematic way. The length of this key depends on the type of algorithm and the targeted cryptographic strength (today from about 64 up to 2048 bits)

There are two basic types of cryptographic algorithms, which differ the number and types of cryptographic keys used.

Symmetrical key algorithms:

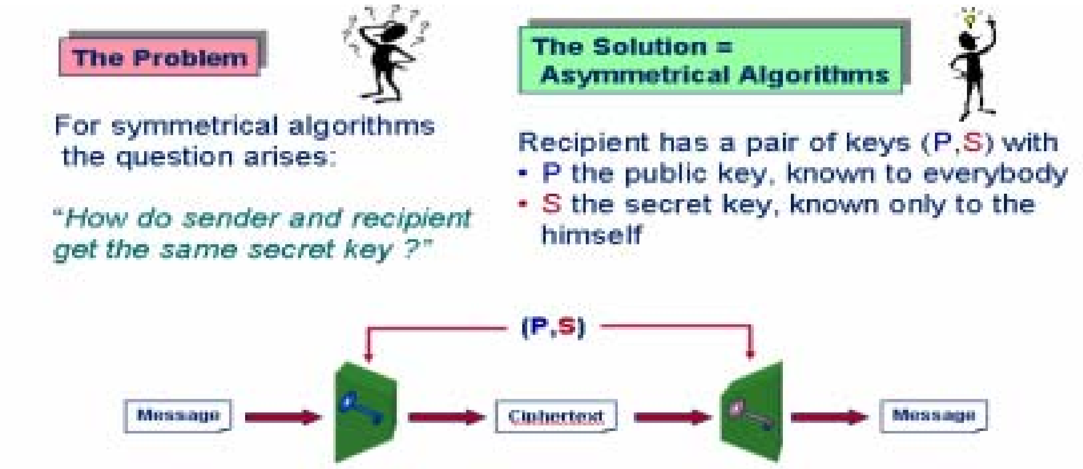


Symmetrical algorithms use one secret key for encryption and decryption

Fig. 3: Symmetrical cryptographic algorithm for en- and de-cryption

Each participant in a communication must have access to this key prior to initiating the communication.

Public key algorithms,
on the other hand, use related pairs of keys. Encryption can be performed with a first key. But decryption is only possible with the second key.



The Problem

For symmetrical algorithms the question arises:

"How do sender and recipient get the same secret key ?"

The Solution = Asymmetrical Algorithms

Recipient has a pair of keys (P,S) with
- P the public key, known to everybody
- S the secret key, known only to the himself

Fig. 4: Asymmetrical cryptographic algorithm with two keys for en- and de-cryption

Asymmetrical algorithms are broadly used for securing communication. The receiver for a message makes its first key public (therefore also called public key) and holds its second key secret under his control (called secret key). Each participant who wants to send him an encrypted message uses his public key for encryption of the message. As the next step, decryption can, by definition, only be carried out by the second key (secret, still at ownership of the receiver) and not by the public key. The receiver

then takes its secret key out of its protected store and uses it for decryption. If someone intercepts the message along the way, they have no way of reading it without access to the secret key.
The big advantage of this algorithm is that no key material has to be exchanged in a secure way before starting communication, which makes such a system much easier for handling and operating.

Digital signing of messages or data

The second goal of cryptography is to ensure the integrity of messages transmitted between two parties. Integrity provides communicating parties with the assurance that a message or a data sequence like a program was not modified.
To ensure integrity, the sender of a message uses a hash function, a mathematical algorithm that creates a unique summary of a message known as a ‘message digest’ and transmits it along with the message. When the recipient decrypts the message, he uses the same hash function to create his own version of the message digest and then compares it to the digest transmitted with the message. If the two digests match, the recipient knows that the integrity of the message is preserved. If the digests differ, something altered the message along the way. (This alteration could be the result of intentional mischief or chance, such as electrical interference, faulty networking equipment or similar failures but also of attacks on the integrity of the data material.)

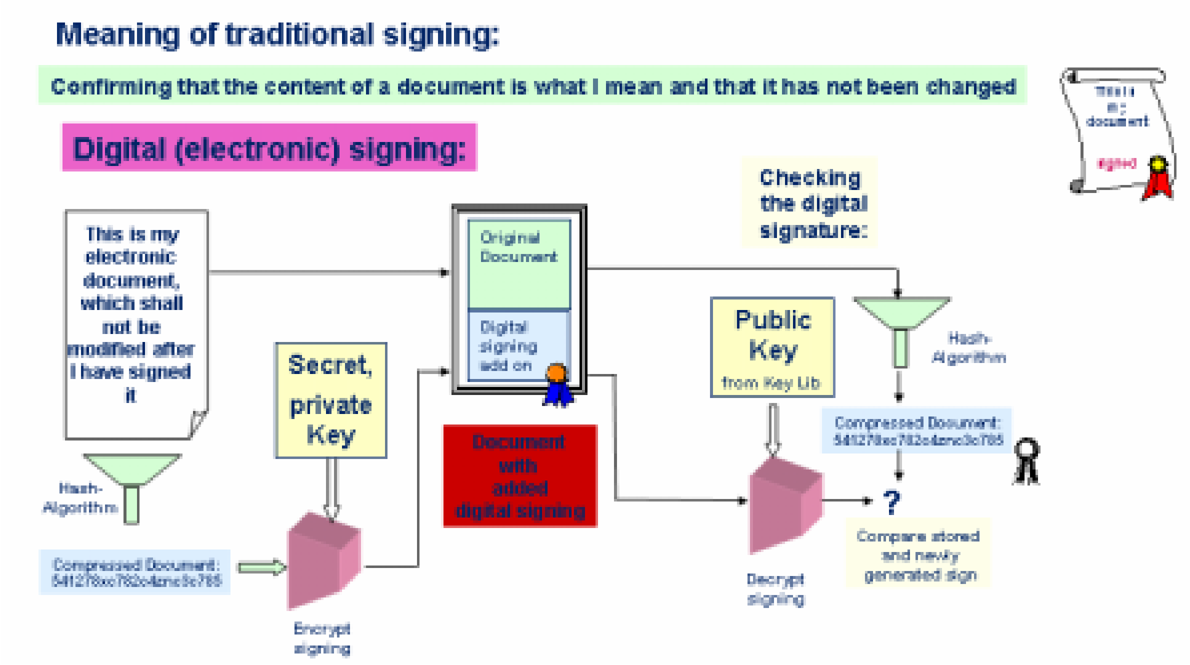

Meaning of traditional signing:
Confirming that the content of a document is what I mean and that it has not been changed
This is my document
signed
Digital (electronic) signing:
This is my electronic document, which shall not be modified after I have signed it
Hash-Algorithm
Compressed Document: 541278xc782c4zne3c785
Secret, private Key
Encrypt signing
Original Document
Digital signing add on
Document with added digital signing
Checking the digital signature:
Public Key
from Key Lib
Hash-Algorithm
Compressed Document: 541278xc782c4zne3c785
Decrypt signing
?
Compare stored and newly generated sign

Fig. 5: Digital signing of messages or documents by using asymmetrical cryptographic algorithm with two keys

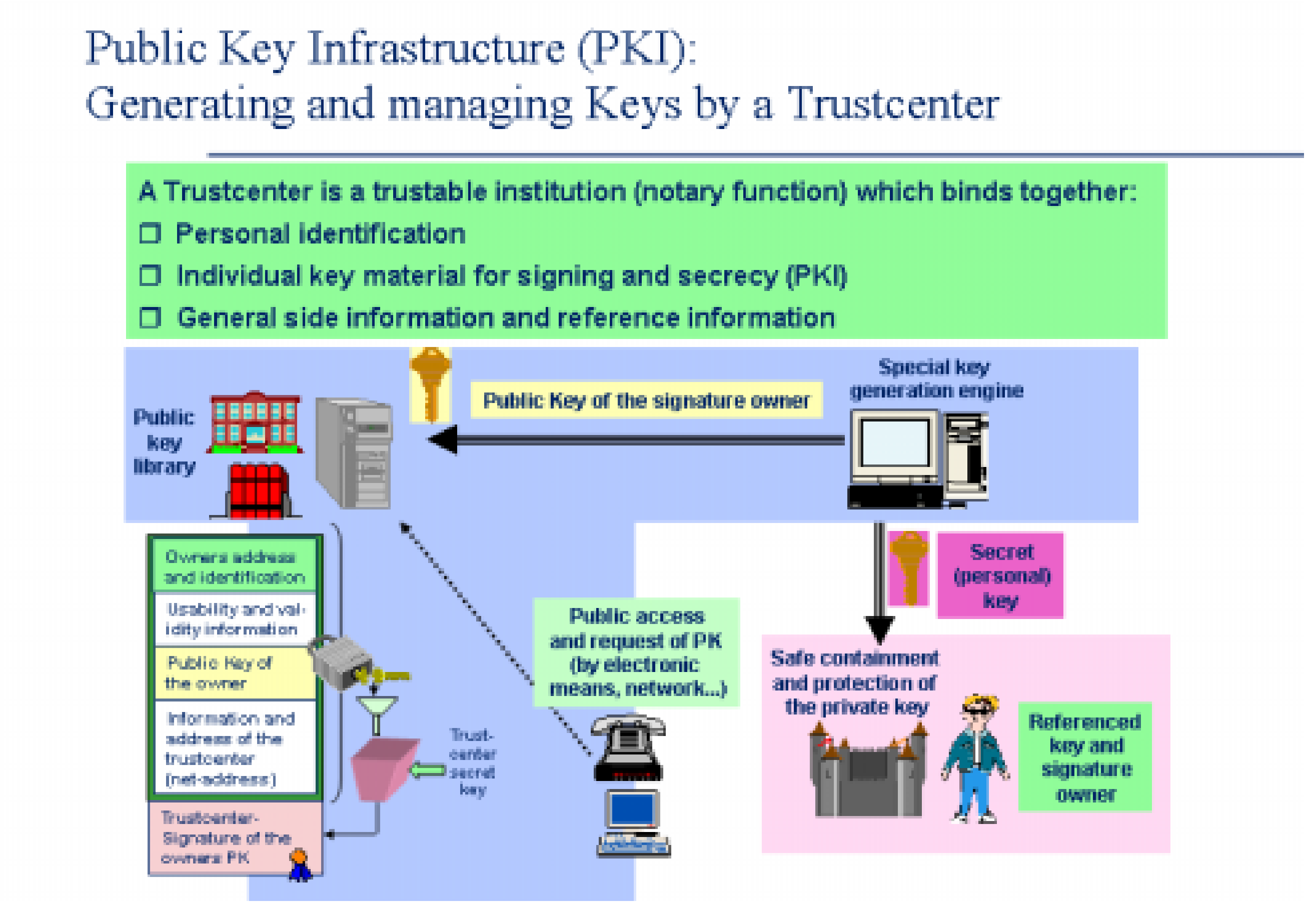
To perform all the necessary security basic mechanisms mentioned at the beginning of this chapter, we have now all the right tools:

- To protect the confidentiality of a message, encrypting the message with the recipient's public key.
- To read an encrypted message sent to you, decrypting the message with your private key.
- To create a digital signature for a message, encrypting the message digest with your private key.
- To verify the digital signature for a message, decrypting the digital signature with the sender's public key and comparing the result to the message digest you compute.

3.2. Cryptographic Certificates

As we have seen, the main element of the abovementioned algorithms and cryptographic systems is the right key, its use and handling. Especially for use with asymmetric functions it is mandatory that everyone knows for whom the used key material is designated and that this is no fake key from a suspicious source. To deliver security also for this step, a special trusted key box, the certificates, is now part of all relevant standards.

Digital certificates are data structures, which contain key material and some descriptive information (From which source is this? for which owner, application …), which are for proof of authenticity digitally signed by a so-called trustcenter (public key infrastructure's (PKI), a trustful third party). So they provide the ability to facilitate communication among large groups of individuals previously unknown to each other. Servers or users seeking to obtain a digital certificate contact one of the well-known Certificate Authorities (CAs), such as well known VeriSign, Thawte, German Telekom or also generate certificates by its own IT- organization. The CA then requires the applicant to prove his identity using a procedure that varies according to the level of trust assigned to the certificate.



Public Key Infrastructure (PKI):
Generating and managing Keys by a Trustcenter

A Trustcenter is a trustable institution (notary function) which binds together:

- ❒ Personal identification
- ❒ Individual key material for signing and secrecy (PKI)
- ❒ General side information and reference information

Public key library
Public Key of the signature owner
Special key generation engine
Secret (personal) key
Owners address and identification
Usability and validity information
Public Key of the owner
Information and address of the trustcenter (net-address)
Trustcenter-Signature of the owners PK
Trust-center secret key
Public access and request of PK (by electronic means, network...)
Safe containment and protection of the private key
Referenced key and signature owner

Fig. 6: Distributing key material with trustful and secure methods: The Trust Centre or PKI

Once the CA is satisfied that the user is authentic, the CA issues a digital certificate. This certificate is structured according to the X.509 standard and contains the name of the CA, the name of the certificate's owner, the certificate's effective and expiration dates, details of the algorithm used by the

CA to sign the certificate and, most importantly, the subject's public key. The CA then signs the certificate using its private key and provides the signed certificate to the certificate subject.

The subject (person or computer) may then use the certificate to prove his identity before entering secure communications with another user or system. The subject merely sends the certificate to the other party who then uses the CA's public key to validate the signature made with the CA's private key. If this validation process is successful, the certificate recipient may be certain that the subject's public key contained within the certificate is authentic. The recipient may then use that public key to initiate a secure communications session with the certificate subject.
The recipient may be confident in the security of the process due to the nature of public key encryption. There is nothing secret about a subject's digital certificate, and it may be distributed freely. If an imposter attempted to use the certificate to impersonate the certificate's true subject, he wouldn't be able to participate in the communications session initiated with the subject's public key, because he wouldn't have access to the subject's private key.

4 Trusted Platform and the Trusted Platform Module (TPM)

4.1 Objects of the TCG Specification

A trusted platform as defined by the TCG consists of trusted hardware and software on the platform, and the connection and integration of external Certification Authorities (CAs, Trust Centres) for enabling cryptographic proof mechanisms. This is also in order to enable the processes and the platform to be identified to the outside world. The platform in turn logically comprises:

- Core Root of Trust for Measurement (CRTM).
  CRTM consists of the routines executed right at the start of booting of the platform (while the operating system is not yet available) in order to achieve secure startup conditions, essentially by measuring and monitoring the integrity of the boot operation. In technical terms, this is accomplished by forming hash values of the critical parts which are then sent to the TPM for checking. These functions have already been implemented in the more recent BIOSs (e.g. AMIBIOS8 with TCG support, [AMI01]).

- Trusted Platform Module (TPM).
  The TPM is the central hardware security device (implemented as a chip) in which all the basic trusted operations and particularly the cryptographic functions are securely handled. Its structure corresponds approximately to that of the well-known high-security smart cards of the type used e.g. for digital signatures or also in payment transactions.

- Trusted Platform Support Service (TSS).
  TSS provides the operating system with a standardized high-level interface (API, referenced in C) to the TPM via which it handles the security functions of the OS or of applications.

- Initial Program Loader (IPL) as the link between BIOS and OS ensures the integrity of the OS.

4.2 TPM: hardware, software, functionality

In accordance with the TCG architecture, the TPM provides the security functions requiring particular protection and which are therefore also implemented in a secure hardware environment. The TPM is designed as a passive part. It has no means of actively influencing program execution of the central processor or the boot operation. It receives only control and status measuring data from the central processor which it processes, stores and reads out again from its secure structure, and feeds these results back to the central processor.
Only access to particular data (such as key material) is made dependent by the TPM itself on the presentation of appropriate authentication patterns.

The main security functions handled by the TPM are:

- **Protection of key material**
  The various key classes are stored in a protected manner in the TPM. The access method is selected according to key type (TPM-bound, migratable, signature, identity (AIK), binding keys).
- **System authentication**
  Authentication and validation of the platform to third parties.
- **Communication of the system's security status (attestation)**
  Trusted communication of the security-relevant (platform-user-defined) configuration.
- **Random number generator**
  Generation of genuine hardware-based random numbers for secure key generation.
- **File sealing**
  Binding of data to the system configuration and signing of the data when storing with the hash value of the configuration. Access to the data is then only possible if the configuration remains unchanged.
- **Secure saving of configuration changes** in the Platform Configuration Registers (PCR). Status changes are detected, safeguarded by the SHA-1 hash algorithm.

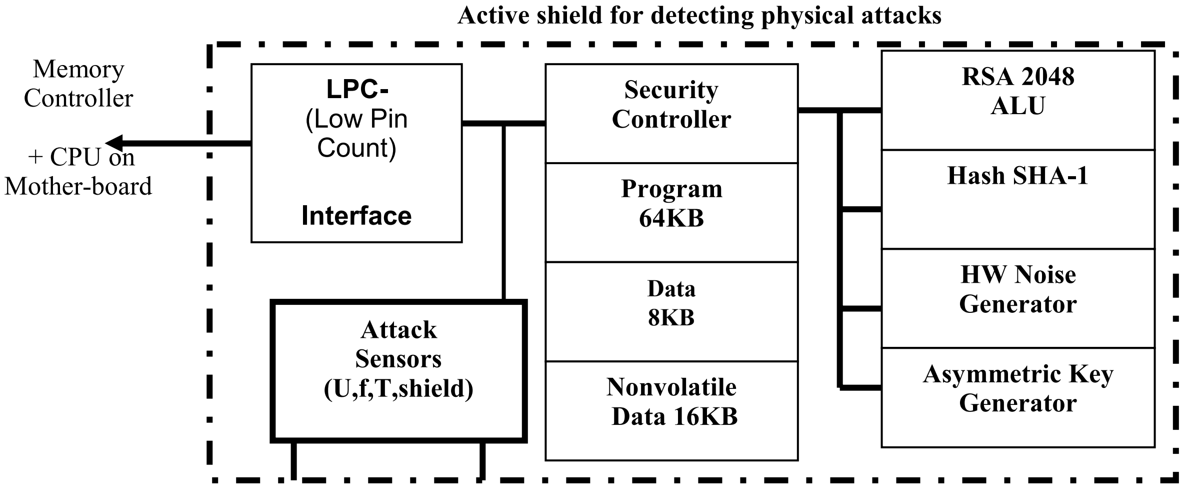


Fig. 7: Block diagram of Trusted Platform Module (TPM)

In addition, the TCG has also placed emphasis on some general, but at least equally important characteristics :

- Protection against attacks on the integrity of the TPM, particularly against physical attacks
- Inexpensive implementation in order to allow widespread use.
- Compliance with global export control regulations in order not to restrict international trade with TC platforms (PCs).

Skilful system design allows implementation with a minimal amount of cryptographic and security hardware in the TPM:

- Specialized crypto arithmetic unit for rapid computation of RSA cryptography up to 2048 bits.
- Key generation for RSA keys up to 2048 bits
- Hardware hash unit for the SHA-1 algorithm
- Genuine hardware noise generator as input for key generation
- Internal processor with the appropriate hardware for computing the critical functions (e.g. RSA with the secret key part) on a trusted basis in a secure environment.
- Monotonic, protected counter and time meter in order to prevent reply attacks.
- Nonvolatile memory (EEPROM) to retain the data even if the operating voltage is switched off
- Sensors and internal security structures (e.g. active screen over the top wiring layer of the chip) in order to detect physical attacks and counteract them.
- TPM self-test function

Extensive internal firmware implements the interface protocol defined by the Standard to the overlying layers of the host software (TSS) and uses the above hardware functions for this purpose. In addition, this firmware also checks and administers the various security sensors and reacts appropriately to detected physical tampering or alterations to the chip or its environment. The correctness of the implementation is checked and confirmed by an independent test institute by means of a complex certification process.

5 The Trusted Computing Software Stack (TSS) for the Host

Like any other hardware element, the TPM requires a special driver and service provider interface in order to enable it to be addressed from the operating system. This Trusted Platform Support Service (TSS) constitutes a security API which provides the TPM functions for the relevant operating system. TSS consists, at the lowest level, of the hardware-based device driver (in kernel mode) which initializes the interfaces and exchanges data with the TPM via the LPC bus. The next higher level consists of the System Service with:

- TPM Device Driver Library
- TSS Core Services
- TSS Service Provider

which handle the following functions:

- Coordination and management of multiple accesses to the TPM
- Converting the abstract API commands to the data stream for the TPM
- Readying the System Service (even if no user is active) for remote access
- A Cache Manager securely stores the data exceeding the memory area on the external mass storage, thereby providing a storage capacity for keys and security data which is limited only by the size of the disk storage. A similarly operating Authentication Cache Manager is additionally provided.

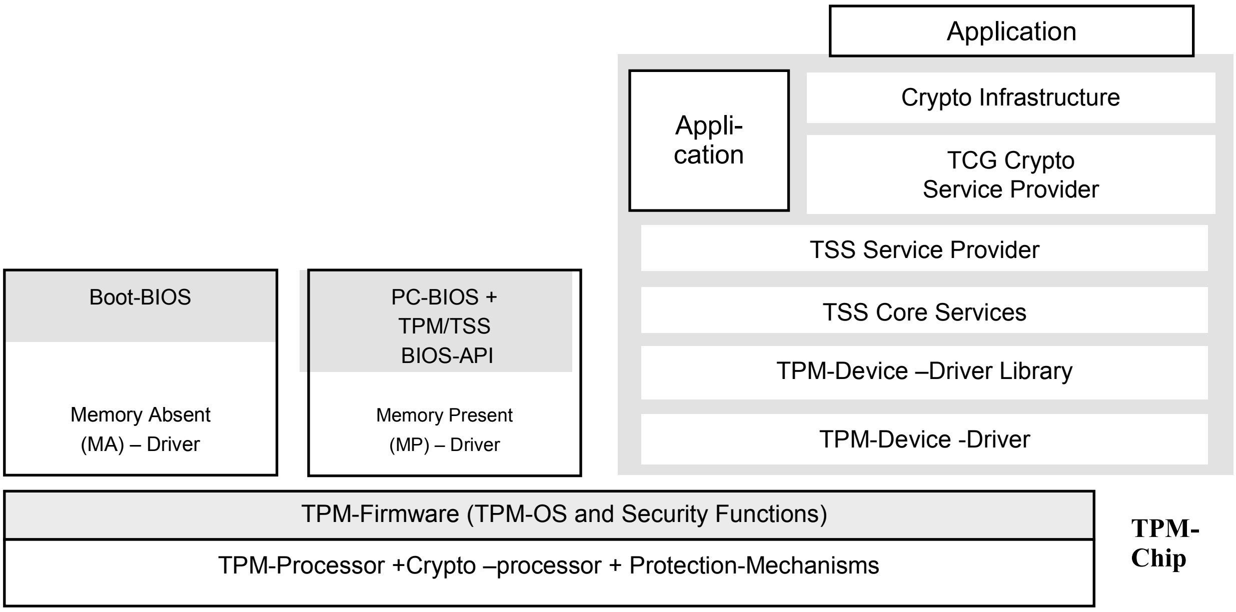


Fig. 8: Trusted Platform Support Services (TSS)

5.1 Cryptographic interfaces of the TSS

Since the TSS, as an API, makes its security functions available to be operating system, it seems reasonable to provide this interface also for other security applications via an adaptation module, thereby enabling in particular secure storage and signature services of the TPM to be made available to the normal applications. Two standard APIs are available, which considerably increases the usability of the platform.

5.1.1 Microsoft Cryptographic Service Provider (MS-CSP)

Various application programs under Windows® (such as Outlook®, Explorer, Word ...) contain security functions such as encryption and signing and handle these functions via the so-called MS-CAPI (Microsoft Cryptographic Application Programming Interface) as a proprietary crypto interface. MS-CAPI can be directed to the TPM by accessing various security providers such as software modules, cryptographic tokens (smart cards, etc.) or again via the TSS. It is therefore relatively easy to port existing security applications already using MS-CAPI to the higher security levels of the TPM merely by selection of a different CSP.

5.1.2 PKCS#11

developed by RSA is the most widely used universal crypto interface standard. It is used e.g. by the Netscape browser. Once again the conversion of the PKCS#11 security calls to the TSS-API facilitates the adaptation of existing standard applications quite considerably. Thus highly secure solutions can be achieved with minimal implementation cost/complexity by securely storing certificates for browsers in the TPM.

5.2 Secure PC-Type platforms: Hardware equipment, Intel's LaGrande and AMD's SEM

As has already emerged from the deliberations concerning the implementation of trusted digital signatures, in addition to making the platform secure, a trusted interface to the user is also required. A trusted platform must of course also be able to satisfy the basic paradigm "What you see is what you get" (WYSIWYG). Although they TPM alone can check security statuses or digital signatures or even generate signatures, it cannot safeguard communication with the outside world or assume the security functions of the main processor. Additional security functions are therefore required in the other building blocks of a platform (secure input, WYSIWYG display, compartmentalized memory areas for the various process domains). The two major PC chipset manufacturers AMD and Intel are founder members of the TCPA and have since been engaged in incorporating the relevant security functions in their chip sets. Information particularly from Intel (LaGrande chip technology) concerning the current status and availability of such trusted PC peripheral devices is currently available [INT01] or advanced [INT02].

6 Secure Key and Data Hierarchy

6.1 Key and certificate chain in the TPM as starting point of the "chain of trust"

As the TPM Specification in accordance with the trust requirements of the TCG is completely public and accessible to all, someone could clone their own TPM on a processor in conformance with this Specification. If e.g. secure e-commerce processes are then transacted, the owner of this "special" TPM could easily modify the internal data to his advantage: such a module would certainly enjoy the full trust of its owner, but would be totally unsuitable for exchanging trusted processes. A trust structure has therefore been implemented which is already known from high-security bank cards:

Endorsement Key

At the end of TPM chip fabrication (after final testing), the manufacturer generates a 2048 bit private/public key pair in the TPM, the so-called Endorsement Key. This is stored in such a way that the private key (PK) can no longer be read out, but can only be used internally in the TPM. The EK is additionally protected by a special certificate. The manufacturer thereby confirms electronically that this TPM has been produced in a trusted process by an inspected manufacturer and meets the requirements of the Specification. The trustworthiness of the entire TPM system is based for the most part on this process and the uniqueness of the EK. The user must trust the manufacture that the private part of the key is not stored anywhere, and that it is not accessible to anyone else. This aspect is also intensively tested as part of the security evaluation. In the case of qualified manufacturers such as Infineon, this fundamental process is performed in the same certified high-security area as for smart cards.

Storage Root Key (SRK)

The SRK forms the root of a key hierarchy in which other lower-order keys, but also data (blobs), are securely stored, their trustworthiness therefore depending on the SRK. The SRK is automatically generated by the owner in a “Take Ownership” operation. If the owner of a TPM gives up this ownership, this also deletes the SRK and also makes all the keys protected by it completely unusable, which is welcome for data protection purposes.

Attestation Identity Key (AIK)

The term attestation implies both authentication and integrity. Using AIKs and external Trust Centres, (even anonymous) identities can be created. AIKs are derived from the SRK and can also be subsequently created or deleted based on their use. Several AIKs are possible for each platform and for each user. Typical applications are:

- Server authentication
- Platform-bound digital contents (DRM)
- Anonymous identities in procurement and tender platforms

Direct Anonymous Attestation (DAA)

In the latest TCG Specification V1.2, another attestation based on zero knowledge methods (direct proof) has additionally been defined, enabling platforms without external CA to mutually attest one another and improved anonymity to be achieved without an external third-party [INT05], [TCG01].
This feature is extremely useful, if one wants to check, if the system on the other side of a communication link is fitted out with a TPM for trusted computing purposes. By use of DAA this can be done in direct data exchange without any third instance.

Certificates

Additional confidence in the correctness of the platform is created using further cryptographic certificates which are likewise stored in the TPM:
The **Endorsement Certificate** confirms that the TPM originates from a trusted source. It contains the public key (PK) of the EK and is used for forming the AIK.
The **Platform Certificate** is brought in by the motherboard/PC manufacturer and confirms that a valid TPM has been mounted in a correct platform. It is likewise used for forming AIKs.
The **Conformance Certificate** is issued by a test laboratory and confirms that the security functions of the TPM and motherboard have been positively checked and are compliant with the protection profile of the TCG.

The concatenation of these various certificates, credentials and keys in order to be able to make various security declarations constitutes a highly sophisticated logical system. The interested reader is referred to [TCG01] TCG Specification Architecture Overview, Sect. 4.2.5.

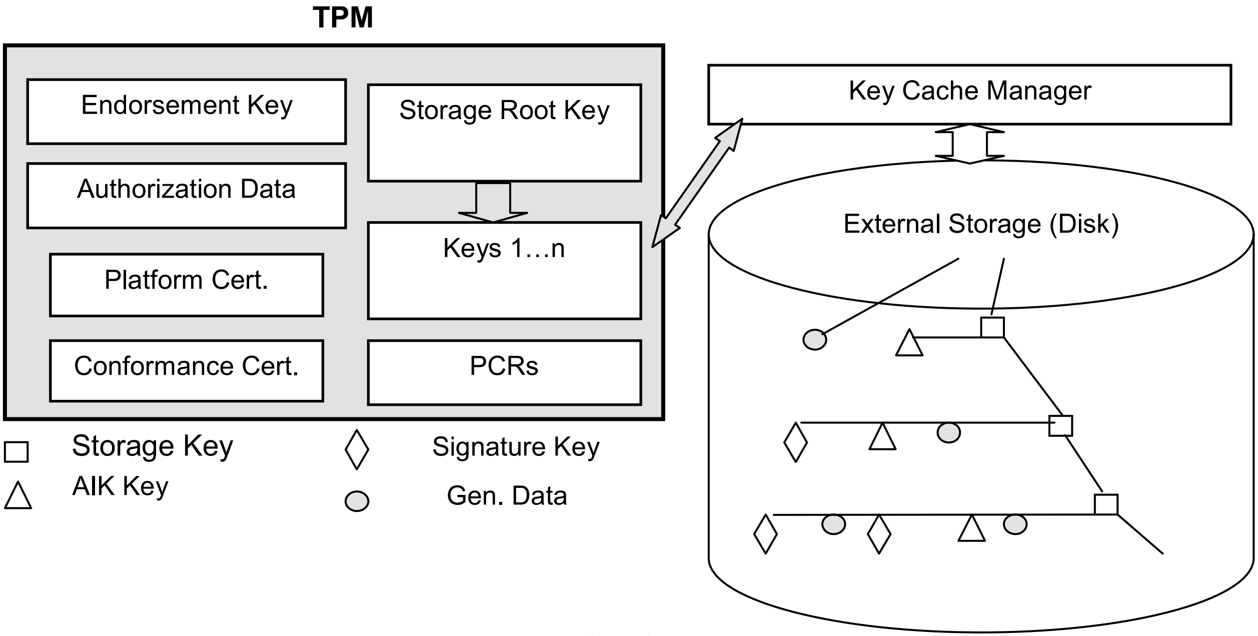


Fig. 5: Certificate chain

Key migration

As both platform-related and person-related keys (user keys) can be stored via the TPM, the need arises to transfer the user keys securely to other platforms. For this purpose the TCG has defined a set over rules under the term *migration*:

Non-migratable keys

(bound to the platform)

Examples :

- EK: Endorsement Key of the TPM manufacturer
- SRK: Storage Root Key

These keys basically cannot be transferred to other TPMs as they are platform-specific. Under the rules, backup (maintenance) is possible as provided for in the Specification.

Migratable keys

(transferable to other platforms)

Example :

- All the keys (if generated in a migratable manner) and data employed by the user and stored under the storage key.

Transport modes

Migration

(transfer to other TPMs)

For migration a special authorization process is used and the data material is transported in a password-protected container for security purposes. In practical terms this can be accomplished by a special migration server

Maintenance

(backing up and restoring key material)
This feature is used for data/key backup in the event of a hardware defect (recommendation only). Implementation by backup server or e.g. smart cards

6.2 Management of TC platforms

Whereas until now only the general logistical requirements have been created for handling standard PCs, new requirements arise for the handling of devices with unique identity and safety elements derived from it. In particular the porting and backup of rights, keys and identities require new terms of reference for use in the field. In the event of defects, it must be possible to migrate the sensitive data securely or re-import the backup data securely [INT07].

6.3 CC certification of chip and firmware

In order to ensure quality and confidence in the TPM module and also to reassure the user of this, it undergoes an official certification process in accordance with ISO15408 (Common Criteria, CC) similarly to the normal practice with smart cards. In the TCG a Protection Profile (PP) has been worked out for certification in accordance with CC EAL3 basic. Like all the other relevant TCG standards it is openly available on the TCG web site [TCG01]. In addition, hardening to CC EAL4 medium is already currently under discussion as an improvement. In any case, however, the prerequisite for a trusted system is successful evaluation and the existence of a valid certificate for hardware and software.

6.4 Operating systems and applications

Contrary to the assumption in some publications, the TCG standard is not bound to any operating system and in particular not to Microsoft. The Standard essentially governs trusted hardware platforms and consequently contains no requirements for the operating system. Only the open TSS-API is provided, which can of course be used by any operating system.

On the other hand, it is only an operating system on a TCG hardware platform that can make flexible trusted computing possible. The TCG Standard with its tamperproof and trusted hardware creates the basis to which the security mechanisms of the OS can relate. It is specifically for that purpose that the TPM provides its signature but also its secure storage functions.

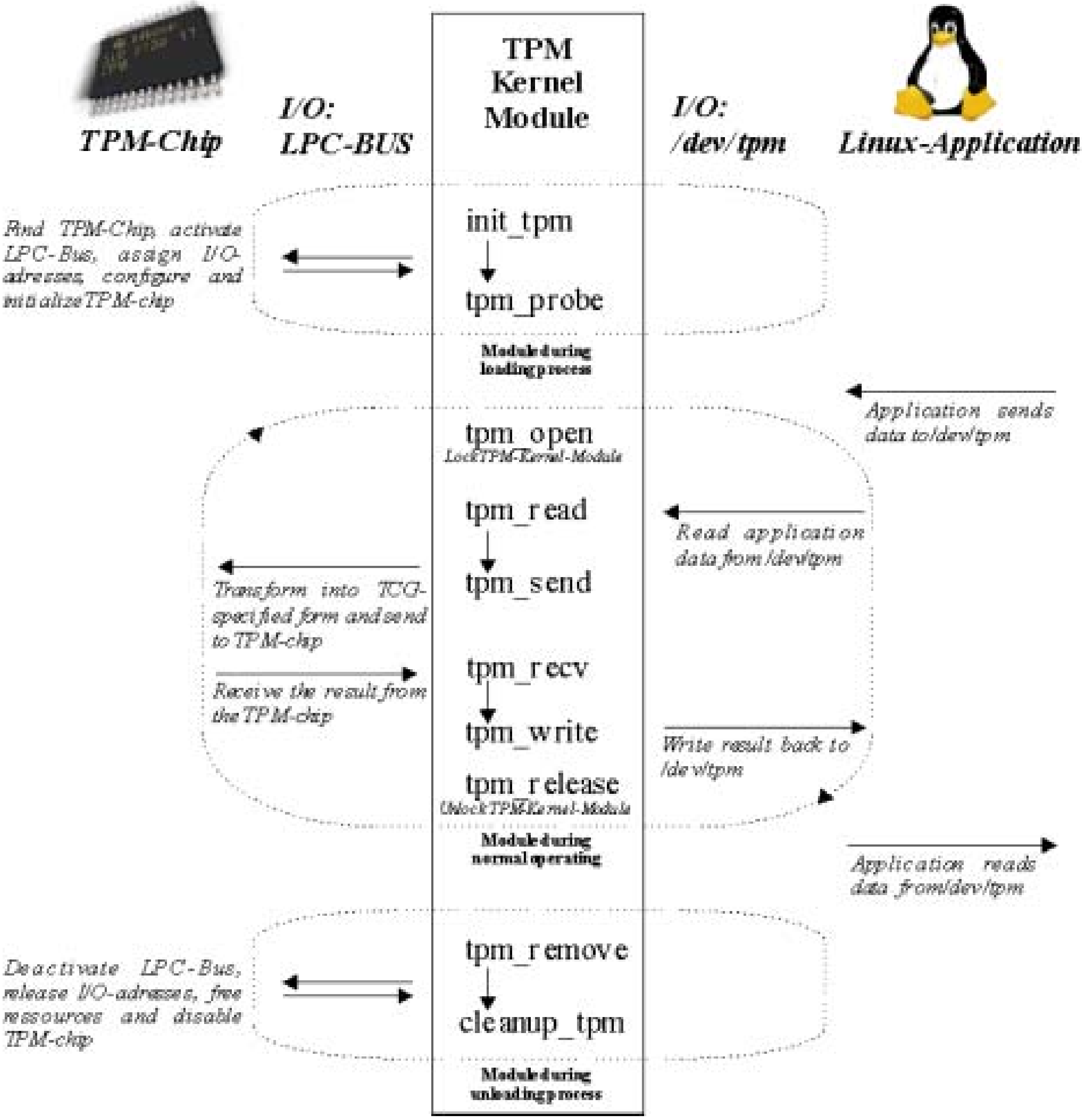
Secure operating systems for TC platforms are mainly designed for separated compartments and domains. The basic considerations here are described e.g. in [TCG03], [INT04].

7 Example for using and interfacing a TPM

In a public available Example (Linux Opensource) we will show how a TPM can be integrated. At the Institute for Applied Data Security at University of Bochum, Department of Electrical Engineering], some TC/TPM applications have been developed and the source code is available under GPL License in the Net [ADS01]. This includes a driver for the Infineon SLD 9630 TPM and also a special trusted bootloader for Linux (Trusted GRUB).

The user interface is provided by different applications (UNIX commands) based on a library called libtcpa that converts the commands entered by the user into TPM commands according to the TCG spec. A Linux kernel module (tpm_infineon.ko) then sends them over the Low Pin Count I/O (LPC) bus to the TPM using a proprietary protocol. Both, user-level applications and kernel module communicate using the special character node devices on the Linux file system */dev/tpm*. The communication with the TPM is unidirectional, so that either writing or reading is possible. The kernel module reads the data from */dev/tpm*, transforms it into the TCG-specific data format and send it to the TPM-chip which stores it into its FIFO. The TPM will then perform the requested operation and send the data back to the kernel module, which writes it back to */dev/tpm*. The detailed process of this communication, including kernel initialization and removal, is shown in the following figure.

Source: [ADS01] Institute for Applied Data Security



TPM-Chip
I/O: LPC-BUS
TPM Kernel Module
I/O: /dev/tpm
Linux-Application
Find TPM-Chip, activate LPC-Bus, assign I/O-adresses, configure and initialize TPM-chip
init_tpm
tpm_probe
Module during loading process
Application sends data to/dev/tpm
tpm_open
Lock TPM-Kernel-Module
tpm_read
Read application data from /dev/tpm
tpm_send
Transform into TCG-specified form and send to TPM-chip
Receive the result from the TPM-chip
tpm_recv
tpm_write
Write result back to /dev/tpm
tpm_release
Unlock TPM-Kernel-Module
Module during normal operating
Application reads data from/dev/tpm
Deactivate LPC-Bus, release I/O-adresses, free ressources and disable TPM-chip
tpm_remove
cleanup_tpm
Module during unloading process

8 Outlook: Fields of Applications

Although the TCG began its activities with PC security in mind, the idea of the secure platform is transferable to other devices and applications. Within the framework of the TCG there now exist workgroups for a wide variety of application fields. Topics of discussion include:

- PDAs and smart phones
  PDAs now have similar features and functions to PCs, are operated continuously on the Internet and are at considerable risk of theft or loss due to their portability. The operating systems which they typically employ mainly have no or only minimal security functions. Losing a device generally results in all the data contained in it being compromised. TPM and its use in the OS can bring about a considerable improvement here by means of implicit authentication and encryption methods.

- Mobile communication applications
  Modern devices such as UMTS are in operation and connected to the Internet 24*7h, and so far no particular security precautions have been taken. With TPM not only can data security be improved, but the applications possible can also be significantly increased. In particular the possibility of securely storing certificates in the device makes new trusted applications possible. Even the increasing risk due to harmful software (a virus/worm which dials the emergency number is now quite conceivable) can be better countered using a TC-based operating system.

- Communication (WLAN security, network remote access ...)
  Now that e.g. the first generation systems of WLANs have been superseded by much more sophisticated protection mechanisms, here too a requirement is arising for secure storage of key material and for storage of device certificates for identification. Here TPM offers the possibility of integrating security into the devices in a trusted manner, not only for the WLAN air interface but also for network accesses (RADIUS, DIAMETER).

- Equipment security and integrity
  For protecting high-value assets against attacks on their integrity or unauthorized modifications, a large number of applications present themselves. These range from protecting product features of cars such as engine control, or of value-related parameters such as mileages, through to protecting system controls e.g. in chemical plants. The discussions here are only in their infancy, but even here platform integrity is opening up new possibilities.

- Digital rights management
  DRM too is a technology which can be used both for management of corporate data (company documents or the secure organization of document and workflow management systems) and for so-called content (music, videos, games, etc.). With minimal expense, security areas can therefore be created on the terminals, enabling the distribution and management of usage rights to be checked in a fair and comprehensible manner.

9 Further reading

- The entire specification and concepts of the TCG can be openly found on the TCG web site [TCG01]. Typical but also necessary for specifications, however, is their considerable size (a total of around 500 pages) and the dry-as-dust presentation which can be quite off-putting [1].

- The basic objectives and principles are described in “TCPA Design Principles” [TCG04]. It is recommended that at least this overview document of the TCG be worked through directly.

- Security experts from HP who were involved in the Specification have written the book on TC [Pea01]: TCPA Design Philosophies and Concepts. This book is the best introduction to the TCG philosophy, is well explained and is best related to applications. Unfortunately the very latest revision of the Specification (V1.2) is currently not yet included.

- [IBM03] examines the misinformation, assumptions and incorrect statements in the TC discussion.

[1] Unfortunately there was therefore often a lack of well-founded information from the Specification in previous TC discussions.

References

[ADS01] Linux Kernel-Module for the Infineon Trusted Platform Module
http://www.prosec.rub.de/tpm/index.html

TrustedGRUB Bootloader
http://www.prosec.rub.de/trusted_grub.html

[AMI01] AMIBIOS8 und TC:
http://www.ami.com/support/doc/AMIBIOS8_TCPA_whitepaper.pdf

[IBM01] IBM-Linux Application Enforcer (Checking of programs during loading)
http://enforcer.sourceforge.net/

[IBM02] IBM: GPL-Sourcecode of the IBM TPM Driver
http://www.research.ibm.com/gsal/tcpa/tpm-1.1b.tar.gz

[IBM03] IBM: Watson Research - Global Security Analysis Lab: TCPA Misinformation Rebuttal
http://www.research.ibm.com/gsal/tcpa/tcpa_rebuttal.pdf

[INF01] Infineon: TPM Product Information
http://www.infineon.com/TPM

[INT01] Intel: La Grande Technology and Safer Computing Overview:
http://www.intel.com/idf/us/fall2003/presentations/F03USSCMS16_OS.pdf

[INT02] Intel: La Grande Technology Architecture
http://www.intel.com/idf/us/fall2003/presentations/F03USSCMS18_OS.pdf

[INT03] Intel: Trusted Mobile Controller Architecture
http://www.intel.com/idf/us/fall2003/presentations/F03USMOBS147_OS.pdf

[INT04] Intel: Software for LaGrande Platforms: Impact to Software Development Process
http://www.intel.com/idf/us/fall2003/presentations/F03USSCMS20_OS.pdf

[INT05] Intel: Privay method for Assuring Trust
http://www.intel.com/idf/us/fall2003/presentations/F03USscms19_OS.pdf

[INT06] Intel: Recovering from Computer failure if TPMs Go Bad
http://www.intel.com/idf/us/fall2003/presentations/F03USSCMS25_OS.pdf

[Lam01] Lambroux: Thesis Royal Holloway University of London (RHUL): TCPA enabled Open Source platforms
http://www.crazylinux.net/downloads/projects/TCPA/TCPA_thesis.html

[MS01]Microsoft: Microsoft's NextGeneration Secure Computing Base (NGSCB)
http://www.microsoft.com/NGSCB

[Pea01] Siani Pearson (ed.): Trusted Computing Platforms: TCPA Technology in Context, Prentice Hall PTR2003

[TCG01] Trusted Computing Group Website: http://www.trustedcomputinggroup.org

[TCG02] TCG: Writing trusted Applications and design principles:
https://www.trustedcomputinggroup.org/downloads/Writing_Trusted_Applications_TCG.pdf

https://www.trustedcomputinggroup.org/downloads/tpmwg-mainrev62_Part1_Design_Principles.pdf