Trusted Computing and Open Source Software

Heiko Stamer

Universität Kassel
Department of Mathematics/Computer Science
Heinrich-Plett-Straße 40, D-34132 Kassel
stamer@theory.informatik.uni-kassel.de
76F7 3011 329D 27DB 8D7C 3F97 4F58 4EB8 FB2B E14F

Kryptowochenende 2006, Kloster Bronnbach
1 Introduction
2 Trusted Computing Platform
3 Trusted Platform Module
4 Open Source Software
5 Demonstration (TPM Emulator, TrouSerS, and jTSS Wrapper)
“Trusted Computing Group (TCG) is a not-for-profit organization formed to develop, define, and promote open standards for hardware-enabled trusted computing and security technologies, including hardware building blocks and software interfaces, across multiple platforms, peripherals, devices.”

- Successor of the former Trusted Computing Platform Alliance (TCPA)
- Open membership: (non-)for-profit corporations, other enterprises
- Different levels of membership ($50,000 – $1,000 p.a.) imply different benefits (pre-publication drafts, voting privileges, participate in Work Groups, product information on TCG web site)
- Reasonable and non-discriminatory (RAND) patent licensing
- Board, Committee, Work Group structure (supermajority voting)

Board of Directors: Mark Schiller (HP), Leendert Van Doorn (IBM), . . .

Advisory Council (2004): Rob Enderle, David J. Farber, Moira A. Gunn, Roger L. Kay, Gary S. Roboff, Rigo Wenning (W3C, FITUG e.V.)

Technical Work Groups: TPM, TSS, PC Client, Mobile Phone, Storage, Server Specific, Compliance, Conformance, Infrastructure, . . .

Promoters: AMD, HP, IBM, Infineon, Intel, Lenovo, Microsoft, Sun
Introduction
Published Specification Documents

- Infrastructure: Specification of the TC-Infrastructure
  - TCG Infrastructure Architecture, Version 1.0
  - TCG Infrastructure Credential Profiles, Version 1.0 (Rev. 0.981)
- Hardware: Specification of the Trusted Platform Module (TPM)
- Mainboard/Firmware/BIOS: Platform Specific Settings
  - PC Specific Implementation Specification, Version 1.1
  - PC Client TPM Interface Specification (TIS), Version 1.2
  - Implementation Specification for Conventional BIOS, Version 1.2
- Software: Specification of the TCG Software Stack (TSS)
  - Version 1.1 (September 2003), Version 1.2 (January 2006)
  - News: Auditing, delegation, monotonic counters, DAA, ...
Usage Scenarios (TCG Specification Architecture Overview):

- **Risk Management** (corporate process)
  - Minimize risk for corporate/personal information assets
  - Evaluated and certified implementation of security technologies

- **Asset Management**
  - Asset tracking and accounting
    (theft and unauthorized use of computing power)

- **E-Commerce**
  - Context for secure on-line transactions, payment, etc.

- **Security Monitoring and Emergency Response**
  - Trustworthy reporting of platform configuration
  - Restricted network access for untrusted devices
Use Case Scenarios (Mobile Phone WG): corresp. “user benefits”

- Platform Integrity (authorized operating system and hardware)
  - “...user can rely upon the trustworthiness of the device ...”
  - “...Device will be able to defend itself against malware ...”
  - “...key building block for other use cases ...”

- User and Device Authentication
  - “In some cases, the User will be better served by the Service Provider who trusts the User’s Device ...”

- Robust DRM Implementation
  - “enable valuable content to be distributed to mobile Devices ...”

- SIM-Lock / Device Personalisation
  - “...obtain device at a substantial discount ...”

- Secure Software Download and Software Use

- Secure Channel between Device and UICC (UMTS Circuit Card)

- Mobile Ticketing and Mobile Payment

- User Data Protection and Privacy
Use Cases (Storage WG): 

- Enrollment and Connection (dual mating, authorized access)
- Protected Storage (e.g. storage locations for system data)
- Locking and Encryption (end-of-life or repurposing)
- Logging (e.g. SMART, forensic logging)
- Cryptographic Services (compliance with well-known standards)
- Authorizing SD Feature Sets to Hosts
- Secure Download of Firmware
TCG Scope: Personal Computers (PC), Personal Digital Assistants (PDA), Cellular Phones, ...  

Fundamental Features (TCG Specification Architecture Overview):

- Protected Capabilities
  - Commands with exclusive permission to access shielded-locations (safe places to operate on sensitive data)

- Attestation
  - Process of vouching for the accuracy of information

- Integrity Measurement, Storage, and Reporting
  - Obtaining metrics of platform characteristics
  - Storing the integrity metrics and a digest of those metrics
  - Attesting the contents of integrity storage

Roots of Trust: components that must be trusted

1. Root of trust for measurement (RTM): e.g. CPU + CRTM
2. Root of trust for storage (RTS): e.g. TPM + Log
3. Root of trust for reporting (RTR): e.g. TPM

Trusted Building Block (TBB): e.g. connection of CRTM, TPM, and Physical Presence mechanism to a motherboard
- TDDLI: OS-independent interface, software-based TPM simulator
- TCS: Common set of TC-platform services (context management, credential and key management, event log of measurements, ...)
- TSP: C interface to the TPM (context management, supporting cryptographic functions, e.g. for DAA protocols)
Trusted Platform Module (TPM):

- I/O (connection to communication bus, e.g. LPC bus on PC)
- Cryptographic Co-Processor and Key Generation
  - Functionality: Asymmetric key generation, Asymmetric encryption/decryption, Hashing, Random number generation
  - MUST support the RSA algorithm (according to P1363 resp. PKCS #1), MUST support key sizes: 512, 768, 1024, 2048
  - MAY implement other algorithms (DSA, ECC) and key sizes
  - RSA public exponent MUST be $e = 2^{16} + 1$
  - If CRT is used, MUST provide protection and detection of failures
  - Symmetric encryption engine to encrypt authentication information, to provide confidentiality in transport sessions, and to provide internal encryption of blobs, e.g. TPM_SaveContext
  - Vernam “One-time-pad” (MGF1 from PKCS #1) with XOR
  - MAY support AES, 3DES, and other symmetric algorithms
Components of Trusted Platform Module (TPM):

- **HMAC Engine**
  - Functionality: Keyed-hashing as proof of knowledge of AuthData, Command authorization and integrity (OIAP, OSAP, DSAP)
  - RFC 2104: $\text{SHA1}((k \oplus \text{OPAD})||\text{SHA1}((k \oplus \text{IPAD})||m))$

- **Random Number Generator (RNG)**
  - Functionality: Provide random values for nonces, key generation, signatures, DAA, \texttt{TPM\_GetRandom}, ...
  - Entropy source and collector, State register, Mixing function
  - Allows implementation of a Pseudo RNG
  - State register MUST be non-volatile and updated by each mixing function call (two register solution for “burn-out problem”)

- **SHA-1 Engine (implemented according to FIPS-180-1)**
  - \texttt{TPM\_SHA1Start, TPM\_SHA1Update, TPM\_SHA1Complete}
  - \texttt{TPM\_Extend}: accumulate measured value $m$ in register $i$
    
    $$
    \text{PCR}[i] := \text{SHA1}(\text{PCR}[i]||m)
    $$

- **Opt-In** (activate/deactivate TPM, Physical Presence flags)

- **Execution Engine** (runs program code to execute commands)

- **Power Detection, Volatile and Non-Volatile Memory (NV)**
Selected Components of Non-Volatile Memory (shielded-locations):

- Endorsement Key (EK)
  - RSA, 2048 Bit, unique, non-migratable, decrypt-only
- Storage Root Key (SRK)
  - RSA, 2048 Bit, non-migratable, **TPM_TakeOwnership**
- Additional Key Slots (**TPM_LoadKey**, e.g. AIKs or legacy keys)
- Vendor Supplied Credentials (ASN.1 format)
  - Endorsement Credential (properly created and embedded EK)
  - Conformance Credentials (valid TPM, valid TBB, etc.)
  - Platform Credential (connects TPM and EK credential)
  - Validation Credentials (ref. for other measurable components)
  - Attestation Identity Credential (AIK; issued by Privacy CA)

- Owner Authorization Data, Persistent Flags, Free Space

Selected Components of Volatile Memory (shielded-locations):

- Platform Configuration Registers (PCR), MUST have 16 (24)
- Data Integrity Register (DIR), MUST have 1, deprecated
Classes of Protected Message Exchange:

- Binding (traditional PK encryption $\rightarrow$ confidentiality)
- Signing (traditional PK signing $\rightarrow$ integrity)
- Sealing (message bound to specific platform metrics)
- Sealed-Signing (signature assures specific platform metrics)

Encryption Schemes:

- Mandatory: RSA PKCS #1 v1.5, RSA OAEP SHA1 MGF1
- Optional: AES/3DES CNT mode or OFB mode

Signature Schemes:

- Mandatory: RSA PKCS #1 v1.5 SHA1/DER/INFO

Protected Storage by using RTS:

- Key hierarchy starting from SRK
- External Storage (storing encrypted BLOBs outside)
- Key Cache Management (resource management outside TPM)
Key Attributes: established at creation time and cannot be changed
- migratable / non-migratable
- isVolatile (must be unloaded on startup with `TPM_ST_CLEAR`)
- pcrIgnoredOnRead (PCRs are ignored when use public portion)
- migrateAuthority (key is under control of a migration authority)

Key Types: associated set of restrictions that limits the use
- Signing Keys: asymmetric general purpose keys for signing
- Storage Keys: asymmetric general purpose keys for encryption
- Identity Keys (AIK): non-migratable signing keys used to sign data originated by the TPM (PCR values, etc.)
- Legacy Keys: created outside the TPM, export/import
- Bind Keys: encryption of small data (`TPM_Bind`, `TPM_UnBind`)
- Authentication Keys: symmetric keys, e.g. transport sessions
- Endorsement Key: asymmetric special purpose key
Command Authorization: for security/privacy related commands

- Authorization Secrets: ownerAuth, usageAuth (keys, e.g. SRK)
- Authorization Sessions
  - HMAC protected messages between entity and TPM
  - HMAC calculation: command ordinal, input/output parameters, return code, authorization secrets, nonces, etc.
- Replay Attack: Avoided by “rolling-nonces” in session state
- Currently Supported Protocols:
  - Object-Independent Authorization Protocol (OIAP)
  - Object-Specific Authorization Protocol (OSAP)
  - Delegation-Specific Authorization Protocol (DSAP)
  - Authorization Data Insertion Protocol (ADIP)
  - Authorization Data Change Protocol (ADCP)
  - Asymmetric Authorization Change Protocol (AACP)
Open Source / Free Software:

- **TPMDD [Hal06]**
  - Character Driver (minor number 224) in `drivers/char/tpm`
  - Supported v1.1 TPMs: Atmel, NSC, Infineon
  - Generic TIS interface for v1.2 TPMs (since kernel version 2.6.17)

- **TrouSerS [Yod06]**
  - TSS Implementation (IBM)
  - Supported OS: i386 GNU/Linux (SuSE, Fedora, Gentoo)
  - License: CPL (Common Public License)
  - Plan: TSS 1.1 compliance while preparing for TSS 1.2 changes.

- **jTSS Wrapper [Win06]**
  - Java API for TSS (TU Graz)

- **Trusted GRUB [Stü04]**
  - Secure Boot Manager (RU Bochum)

- **TPM Emulator [St06]**
  - Software-based Emulator (ETH Zürich)
Mario Strasser: **Software-based TPM Emulator for Linux** [St06]

https://developer.berlios.de/projects/tpm-emulator/

**Goal:** Create a fully working software-based Trusted Platform Module emulator according to TCG Specifications, Version 1.2

**Application:** Explore TPMs for educational/experimental purposes

**Current State:** Release 0.4 (June, 2006), GNU GPL v2

- **Kernel module** `tpm_emulator.ko` (provides char. device `/dev/tpm`)
- Currently, 83 out of 120 TPM commands are implemented (admin startup, admin testing, admin opt-in, admin ownership, auditing, storage functions, cryptographic functions, endorsement key handling, identity creation and activation, integrity collection and reporting, authorization sessions, session management, eviction, timing ticks, transport sessions, monotonic counter, DAA, deprecated commands)
- Not yet/only partially implemented: capability, migration, maintenance, delegation, NV storage
- Packetized for Gentoo Linux (`$ emerge tpm-emulator`)

**Prerequisites:**
- Linux Kernel 2.6.x, GNU Compiler Collection, . . .
- GNU Multiple Precision Arithmetic Library (libGMP)
Roadmap/TODO:

1. Conformance with Revision 94 of the TPM Specification 1.2 (largely finished now)
2. Obtain better portability (kernel space vs. user space)
   1st Problem: Kernel stack size is very limited (architecture dependent, e.g. 4K resp. 8K on x86)
   2nd Problem: Persistent storage is needed to save the state

Possible Solution:
- Dummy “hardware interface” in the common TPM device driver
- TPM emulator serves only as a user space daemon (TDDLI)
3. Implementation of all mandatory commands (v1.2 rev 94)
4. Adding optional commands and algorithms (e.g. AES)

Contributions to our project [St06] are very welcome!
- Announce practical projects for university students (SOC)
Demonstration (TPM Emulator, TrouSerS, and jTSS Wrapper)

Demonstration

Heiko Stamer (Universität Kassel)  Trusted Computing, Open Source Software  Kryptowochenende 2006
Do you still know the “meaning” of the following abbreviations?

AACP, ADCP, ADIP, AIK, CRTM, DAA, DIR, DRM, DSAP, EK, NV, OIAP, OSAP, PCR, RAND, RTM, RTR, RTS, SRK, TBB, TCG, TCPA, TCS, TCSI, TDDL, TDDLI, TIS, TNC, TPM, TSP, TSPI, TSS, UICC

Thank You! Questions?

https://www.trustedcomputinggroup.org/


https://developer.berlios.de/projects/tpm-emulator/


http://tpmdd.sourceforge.net/


http://trousers.sourceforge.net/


http://trustedjava.sourceforge.net/


http://www.prosec.rub.de/trusted_grub_details.html