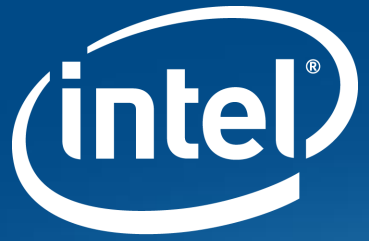


Software

TPM2 SOFTWARE STACK (TSS2)

Philip Tricca <philip.b.tricca@intel.com>

Agenda

TPM2 Sotware Stack (TSS2) in OSS

- Background
- Standardizing TSS2 in TCG
  - Components / Architecture
  - Intended use cases
- tpm2-software community
  - Purpose / goals
  - Participants, adoption & emerging use cases
  - Alternatives
- Musings
  - Interesting developments in TCG specs
  - TPM adoption in F/OSS community

TPM 1.2 vs TPM 2.0

TPM 2.0 resolves shortcomings of the 1.2 spec

- Use-case unchanged: keep keys out of main memory
- TPM 1.2 limited algorithm support
  - Require RSA 1k, 2k & SHA1, no larger key / hash sizes, AES optional
  - Single hierarchy, limited policy
- TPM 2.0 addresses shortcomings of 1.2
  - Flexible to support multiple algorithms & policy
  - PC Client spec requires RSA 2k, ECC P256, AES 128 & 256, SHA256 etc
  - Integrity protected and encrypted sessions
- Support for 1.2 is being phased out
  - Some OEMs dropping support for TPM 1.2 under Linux [1]

TPM2 SOFTWARE STACK (TSS2)

Design

API DESIGN / GOALS

Use-case driven design, integration with async programming

- Layered design, async throughout
  - Separate transport layer from APIs,
  - Async: suitable for integration with existing “event-driven” programming models
  - Details exposed if you need them, hidden behind “sane defaults” otherwise
- Lower layers of stack provide thin layer over TPM2 commands
  - Assume “expert” applications in thin environments, possibly w/o heap or filesystem
  - Minimal dependencies (libc)
- Upper layers provide convenience functions & abstractions
  - Adds dependencies on crypto libraries
  - Allocates memory on behalf of the caller

TCG TPM2 SOFTWARE STACK: DESIGN

System API (SYS)
- 1:1 mapping to TPM2 commands
- Async I/O
- No file I/O
- No crypto
- No heap

Enhanced SAPI (ESYS)
- Additional utility functions
- Provides Cryptographic functions for sessions
- No file I/O
- Requires heap / does memory allocations

Feature API (FAPI)
- Spec in draft form
- No implementation yet
- File IO
- Requires heap
- Must be able to do retries
- Context based state
- Must support static linking

TPM Command Transmission Interface (TCTI)
- Abstract command / response mechanism,
- Decouple APIs from command transport / IPC
- No crypto, heap, file I/O
- Dynamic loading / dlopen API

TPM Access Broker and Resource Manager (TAB/RM)
- Power management
- Potentially no file IO – depends on power mgmt.
- Abstract Limitations of TPM Storage
- No crypto

TPM Device Driver
- Device Interface (CRB / polling)
- Pre-boot log handoff

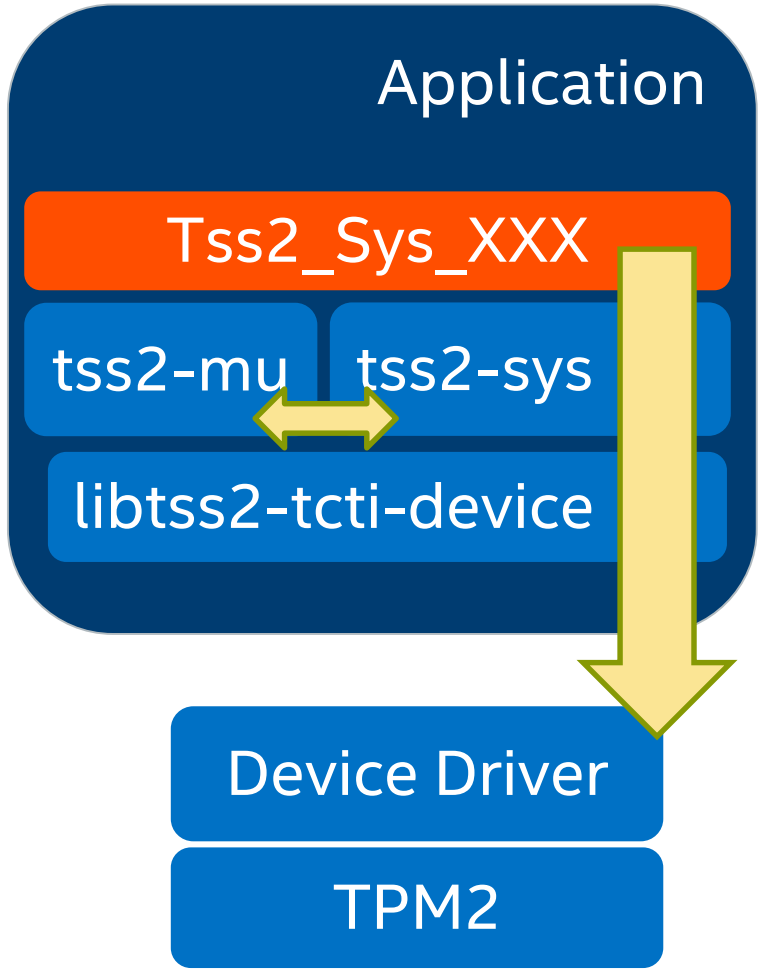
User

Kernel

TCG TPM2 SOFTWARE STACK

System API, Type Marshaling, & TCTI

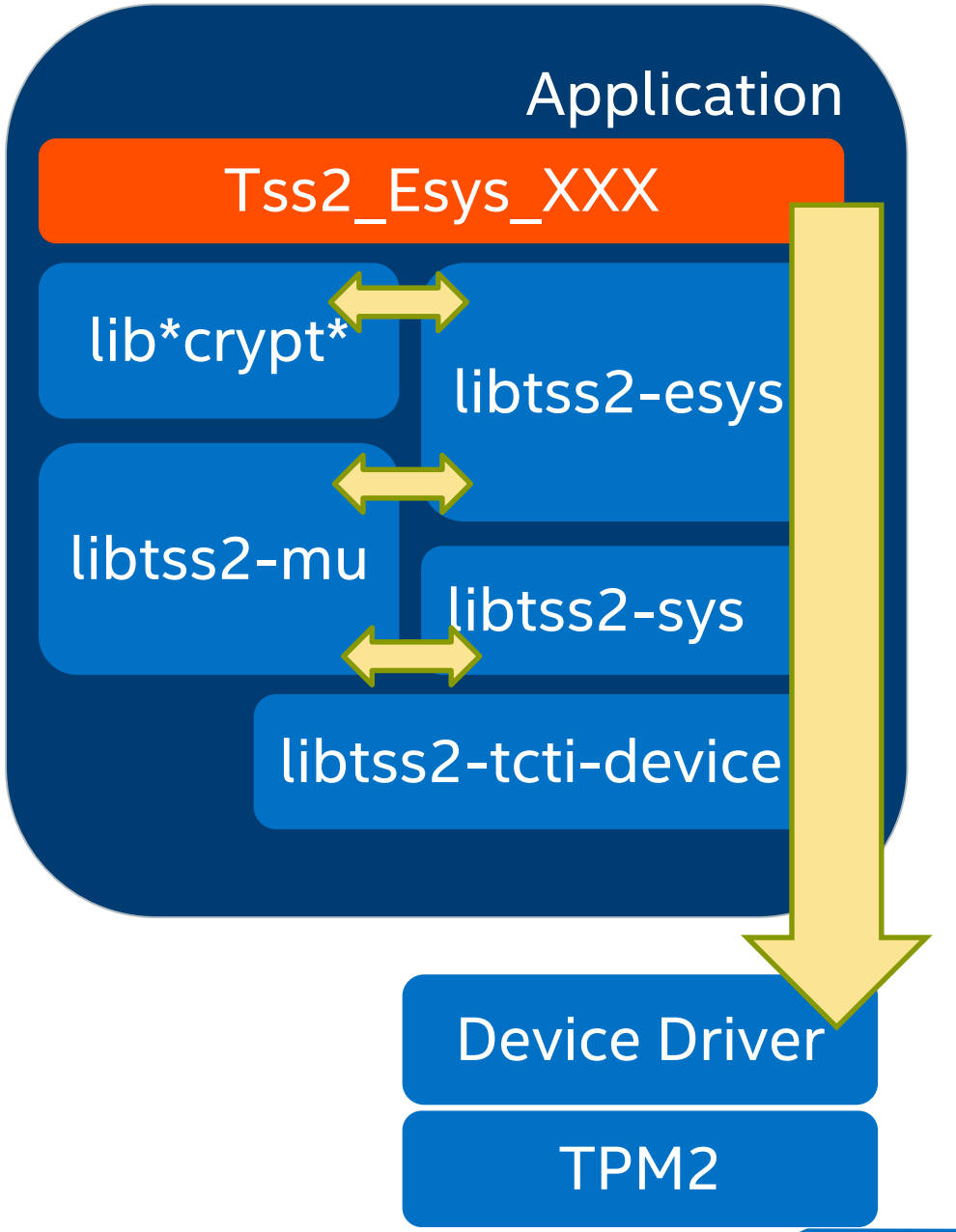
- System API: libtss2-sys
  - Transform C types to TPM command buffer
  - One-to-one mapping to TPM commands
  - Suitable for firmware / embedded applications
- Type Marshaling: libtss2-mu
  - Transform TPM types from C to wire format & back
- TPM2 Command Transmission Interface
  - Abstraction to hide details of IPC mechanism
  - libtss2-tcti-device + libtss2-tcti-mssim



TCG TPM2 SOFTWARE STACK

Enhanced System API: libtss2-esys

- Suitable for general C applications
- Builds on top of lower-level tss2-* libs
- Expose all TPM2 functions + utilty functions
  - HMAC calculations for HMAC session
  - Encryption / Decryption for encrypted session
  - Maintain state for authorizations
- Adds dependency on crypto library
  - Current implementation use libgcrypt
  - Additional crypto modules under development



TPM2 RESOURCE MANAGEMENT

TPMs are resource constrained: small & inexpensive

- RAM on the order of “a few kilobytes”
- Scarce resources must be shared
  - TPM commands specific to object and session management:
    - ContextLoad, ContextSave & FlushContnext
  - Resource Management: Saving & Loading “contexts”
- Isolation through Resource Management
  - Associate objects (keys, session) with connection
  - Prevent access by other connections (with exceptions)
- Components of resource mgmt. tasks moving into kernel driver
  - /dev/tpmrm0: performs simple object / session isolation & load / save
  - Currently aligning user-space daemon w/ in-kernel resource mgmt.

TSS2 IMPLEMENTATION & NEW USE CASES

Bootstrapping & expanding community

FROM PROTOTYPE TO OSS PROJECT

Stability & Reliability

- Align development model with community norms: know your audience
- Semantic versioning scheme: https://semver.org
- Model a healthy OSS project
  - Align to a coding standard (always an uphill battle)
  - Testing: unit & integration, make adding new tests easy
  - Continuous Integration (CI): travis-ci, coveralls, coverity / static analysis
- Eliminate high priority technical debt
  - Make it debuggable
  - Complete re-write of resource mgmt. daemon

TPM2-SOFTWARE PROJECT GITHUB

Community dedicated to development and use of TCG TSS2 APIs

- TPM2 Software Github Org: https://github.com/tpm2-software
  - Core libraries: https://github.com/tpm2-software/tpm2-tss
  - Resource Mgmt: https://github.com/tpm2-software/tpm2-abrmd
  - Command line tools: https://github.com/tpm2-software/tpm2-tools
  - Mailing list: https://lists.01.org/mailman/listinfo/tpm2
  - #tpm2.0-tss on Freenode
- Community
  - Maintainers from: Intel, Fraunhofer SIT (tss2-esys), RedHat
  - Patches from: Facebook, Alibaba, Suse, RedHat, Debian
- Impending Projects
  - PKCS#11 module, OpenSSL Engine

COMMAND-LINE TOOLS: TPM2-TOOLS

Automate Common Tasks

- Often times a user's first experience with the TSS2
- Started as a clone of the IBM command line tools from TSS for TPM 1.2
- Has evolved into a near 1:1 mapping to TPM2 commands
- Individual tool execs can be strung together to achieve a higher level task
  - Create policy assertion
  - Create object bound by policy
  - Save object to disk
- Adding new tools on demand

DOWNSTREAM ADOPTION

Support and Usage in OpenEmbedded, RHEL & Suse

- OpenEmbedded upstreaming efforts underway
  - Maintained as part of meta-measured
  - Planning effort to upstream into OE proper: reduce duplication
    - Integrate config fragments with yocto-kernel-cache via MACHINE_FEATURES
    - Integrate kernel module packages into packagegroup-base
    - TSS recipces in openembedded-core, included by DISTRO_FEATURES?
- RHEL 7.4 - 1.x releases of TSS2 libraries as “official” package
  - Working to align 2.0 TSS2 release to RHEL 8
  - Clevis supporting TPM2 module [6]
- Suse adding support in next major SLES release

NEW USE-CASES: AC_* COMMANDS

Attached Components (AC)

- TPM2 is useful for key protection, not so great for usage
  - Create, use keys in shielded location
  - Rich policy governing key use
  - TPM often referred to as a “crypto decelerator”
- AC Commands
  - Allow for TPM to release unencrypted keys to attached components [2]
  - Extend policy to include AC
  - “attached” is loosely defined: not over command channel
  - Use-case driven: Encrypted SSDs, crypto accelerators

ALTERNATIVES

Healthy OSS always has options

- Google / ChromeOS: Chaps [8]
  - "Chaps" daemon brokers access to TPM device
  - PKCS#11 module communicates with Chaps over dbus
  - OSS, but not a "Google project"
- IBM [9]
  - Non-standard API
    - Effectively TCTI, SYS & ESYS rolled together
    - No async operation
  - 3 functions
    - Create & Delete context object
    - Execute TPM2 command: command code passed as parameter, varargs
  - Stated goal is simplicity, maybe too simple

MUSINGS

Other Interesting Stuff

DICE & EMERGING SPECS / WORKING GROUPS

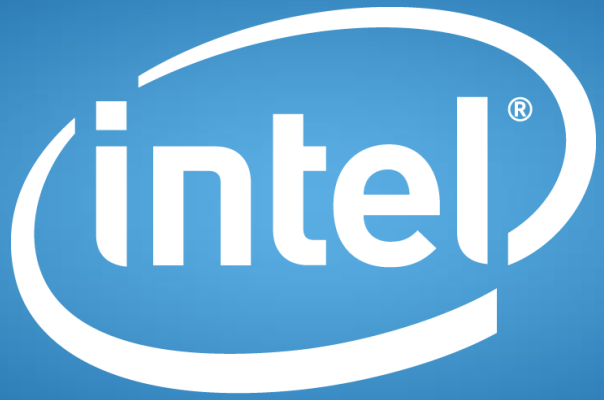
Device identity composition engine (DICE)

- First TCG spec I've read in one sitting – 12 pages
- Intended for small devices, too small for TPM2
  - TPM2 identity schemes establish identity through possession of key
  - DICE combines unique secret with hash of "first mutable code" -> CDI
- DICE itself is root of trust, updates permitted but not apparent in CDI
- CDI susceptible to replay
  - First mutable code to prevent access after launch
  - Fixing bugs in first mutable code -> new CDI
- Trend: shift from platform identity to identity of individual parts

TPM2 IN FOSS

FOSS TPM adoption & ingredients to build your own

- Purism: security and privacy focused laptops
  - Offered optional TPM: Reported 98% of orders opted for TPM at additional $99 [3]
  - Purism Librem laptops shipping with TPM by default (1.2 ☹)
- Microsoft released reference code from spec as OSS [4]
  - See CONTRIBUTING.md – changes to some parts -> changes to TCG spec
- Stefan Berger libtpms & swtpm
  - Integration with QEMU in 2.11 [7]
  - Subsequent integration / backport to QEMU 2.10 for openmbedded & meta-iot-refkit
- Google's ChromeOS / platform: OSS, but not a "google project"
- Pick your TPM2 code, add TEE / hardware (Cortex M?)



Software

REFERENCES

[1] Dell TPM support: http://en.community.dell.com/techcenter/enterprise-client/w/wiki/11849.tpm-1-2-vs-2-0-features

[2] Intel QAT & KPT: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/key-protection-technology-paper.pdf

[3] Purism, TPM by default: https://puri.sm/posts/tpm-by-default-and-free-international-shipping/

[4] Microsoft TPM2 reference code: https://github.com/Microsoft/ms-tpm-20-ref

[5] Microsoft Technical Report on ARM TZ + TPM2: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/msr-tr-2015-84.pdf

[6] Clevis TPM2: https://blog.dowhile0.org/2017/10/18/automatic-luks-volumes-unlocking-using-a-tpm2-chip/

[7] QEMU 2.11 ChangeLog w/ TPM: https://wiki.qemu.org/ChangeLog/2.11#TPM

[8] ChromeOS Chaps: https://www.chromium.org/developers/design-documents/chaps-technical-design

[9] IBM TSS: https://sourceforge.net/projects/ibmtpm20tss/