TCG TPM2 Software Stack & Embedded Linux

Philip Tricca
philip.b.tricca@intel.com

Agenda

Background

- Security basics
- Terms

TPM basics

- What it is / what it does
- Why this matters / specific features

TPM Software Stack

- Architecture / Design
- Getting Started
- Getting Results

Level Set

There is no magic, there are no silver bullets

- “security” takes the whole village
- Architecture to implementation to maintenance
- There is no such thing as “a secure system”, only secure enough
- YOUR CUSTOMERS define “secure enough”

The Basics

Using the TPM does not a secure system make

- FTC case against ASUS: didn't take "reasonable steps" to secure its routers
  - Must maintain a comprehensive security program
- Mirai (nuf said)
- Basics == "reasonable steps"
  - Disable services / exclude tools / minimize exposure (aka attack surface)
  - Use writable storage only when you must
  - SIGNED UPDATES!
- Securing general purpose computers is a nightmare, embedded more tractable

Threat modeling

A process by which we identify, enumerate, prioritize & document

- Assets
- Threats to them

- IMHO the most important part of your security program
- Prioritize: decide where your efforts are best spent
  - Identify trade-offs
  - Accurately describe the properties of your system
    - What it protects against: threats mitigated
    - What it does not: threats accepted
    - And most importantly: why

If your team doesn't model threats …

Please do?

- Much of the body of knowledge was developed in Microsoft
- MSDN has lots of free content
  - https://msdn.microsoft.com/en-us/library/ff648644.aspx
- OWASP Application Threat Modeling
  - https://www.owasp.org/index.php/Application_Threat_Modeling
- Adam Shostack's book was my introduction (2014)
- Swiderski and Snyder book (2004)

Terms

Classic security concepts:

- Confidentiality
- Integrity
- Authentication
- Authorization (satisfy TPM2 policy)
- Non-repudiation

Use the TPM2 to build systems that implement these principles

TPM Protections

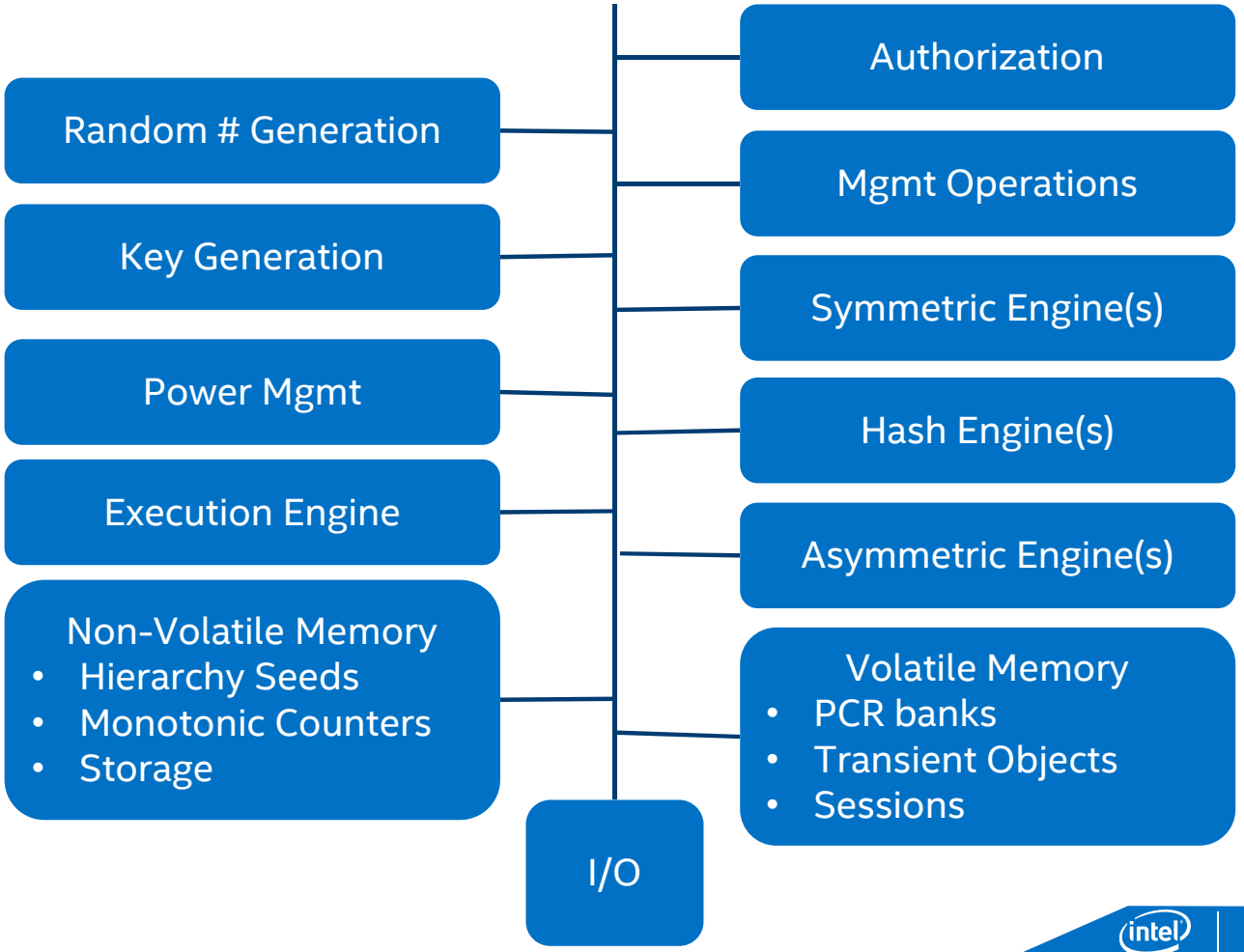
Documented in TPM Rev 2.0 Part-1: Architecture

- Frames protections offered by TPM2 in section 10:
  - Shielded Location
  - Protected Capability
  - Protected Object
- TPM operations must be correct, sensitive data must be protected
- TPM severely memory constrained
  - offload storage to applications, encrypt all protected objects when not in shielded location
- Nature of physical security protections dictated by customer / requirements

What is a TPM?

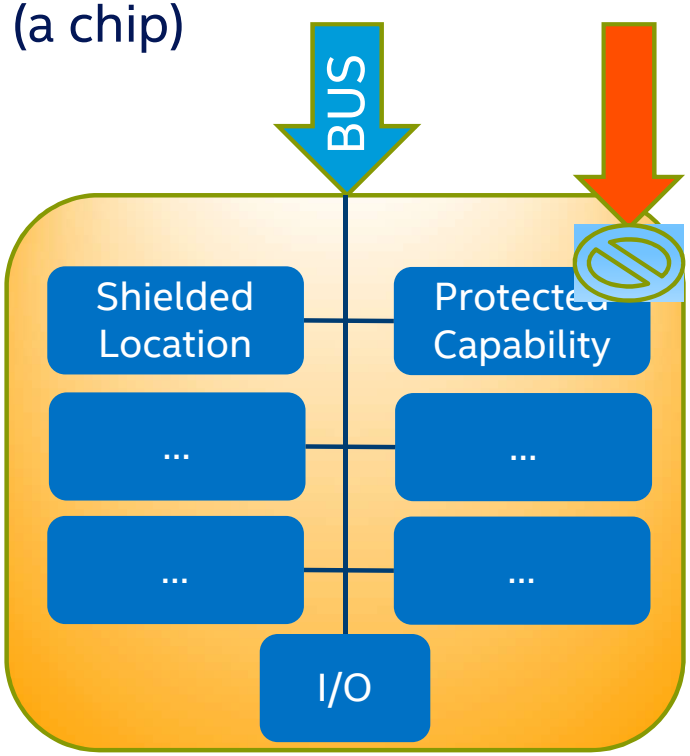
Small Crypto Engine

- Cryptographic functions
- Hashing functions
- Key generation & protection
- RNG
- Integrity measurement / reporting

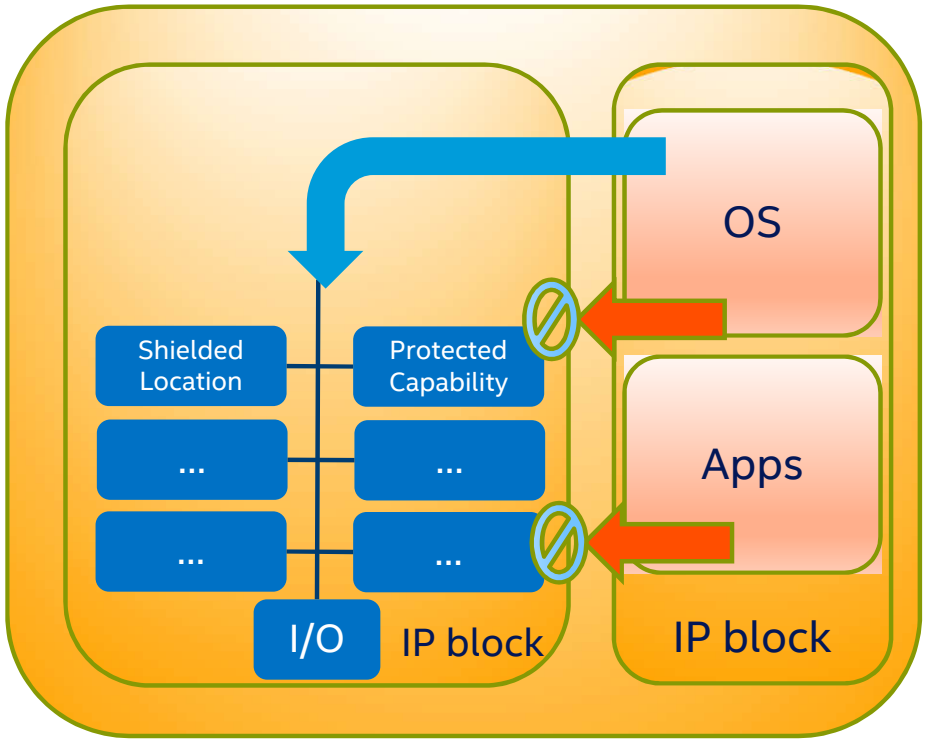


TPM2 Implementation: domain separation

Discrete IP Block
(a chip)



Integrated IP Block



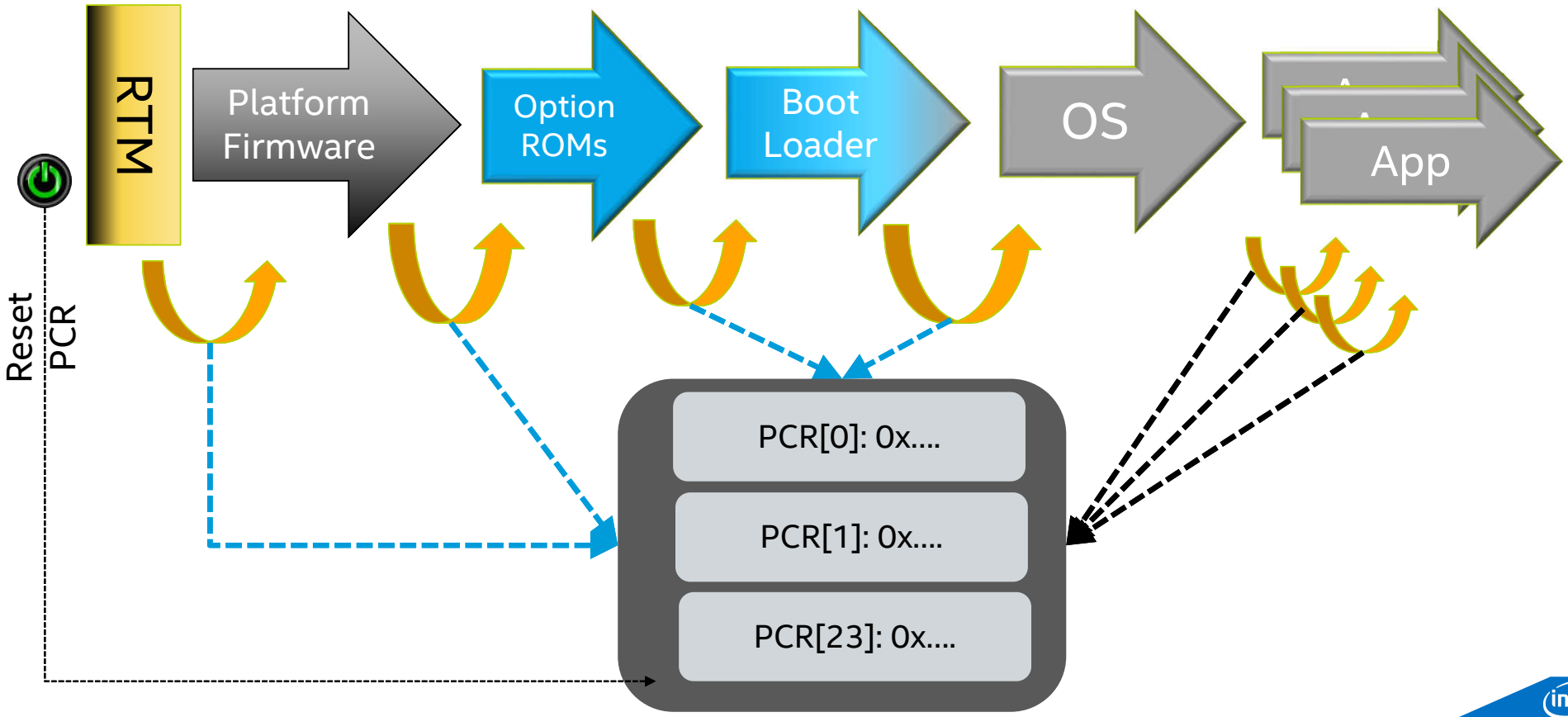
Integrity: Measured Boot

Platform Configuration Register (PCR) & the “Extend” operation

- Typically 24 PCRs in a TPM, addressed with index: PCR[0] – PCR[23]
- PCR is a Shielded Location, Extend operation is Protected Capability
- PCR usage (store hashes of which components) defined in TCG platform specs

- Software Measurement is synonymous with the hash produced
  - Extend hash of object (executable, config etc) into PCR
  - Extend: $PCR[0]_N = H(PCR[0]_{N-1} \mid X)$
  - PCR state becomes one way function depending on previous state
  - Computationally infeasible to forge, easy to verify

Integrity: Measured Boot



TCG TPM2 Software Stack: design goals

System API (SYS)

- 1:1 mapping to TPM2 commands
- No
  - file IO
  - crypto
  - heap
  - external library dependency

Enhanced SAPI (ESYS)

- 1:1 mapping to TPM2 Commands
- Additional commands for utility functions
- Provides Cryptographic functions for sessions
- No file IO
- Requires heap

Feature API (FAPI)

- File IO
- Requires heap
- Must be able to do retries
- Context based state
- Must support the possibility of reduced application code size by offering static libraries

TPM Command Transmission Interface (TCTI)

- Abstract command / response mechanism
- Decouple APIs driving TPM from command transport / IPC
- No crypto
- No heap, file I/O

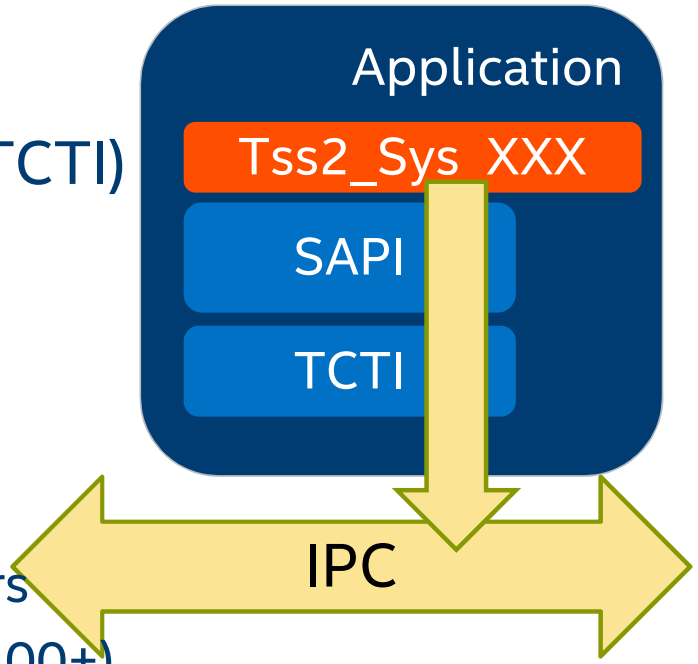
TPM Access Broker and Resource Manager (TABRM)

- Power management
- Potentially no file IO – depends on power mgmt.
- Abstract Limitations of TPM Storage
- No crypto

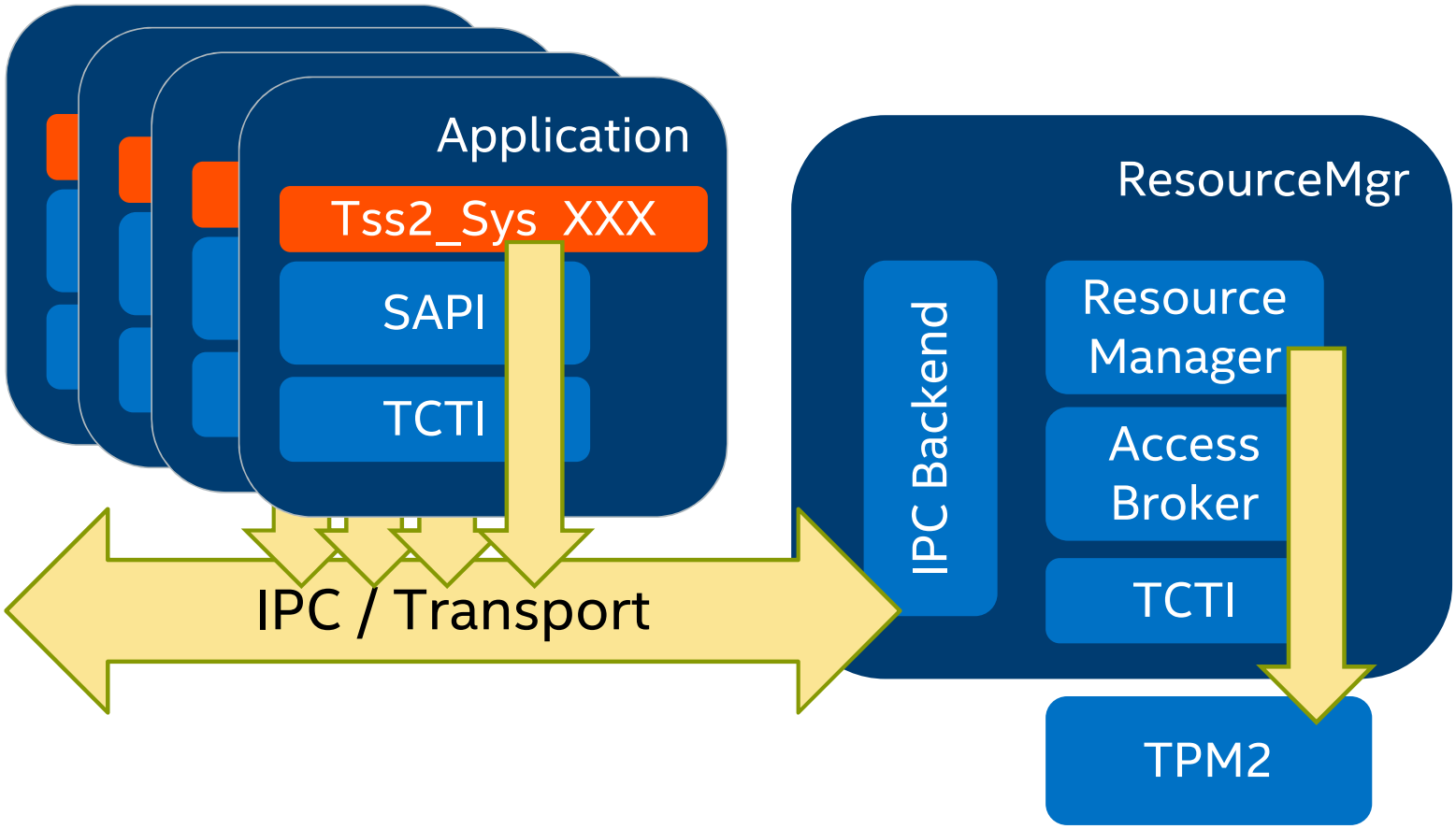
TPM2 software stack

System API & TCTI specification

- TPM2 Command Transmission Interface (TCTI)
  - Abstraction to hide details of IPC mechanism
  - libtcti-device & libtcti-socket
  - Adds flexibility missing from 1.2 TSS
- System API (SAPI)
  - Serialize C structures to TPM command buffers
  - One-to-one mapping to TPM commands (all 100+)
  - Minimal external dependencies: libc
  - Suitable for highly embedded applications / UEFI



TPM2 TSS Components: w/ resourcemgr



Use case: RNG

TPM requires RNG for key creation, nonce generation.

- an entropy source and collector
- state register
- mixing function (typically, an approved hash function)
- Differentiation between TPMs w/ certification (NIST SP800-90 A)
- TPM RNG integrated with Linux kernel RNG
  - If you need an entropy source DO NOT use TPM RNG alone
  - Load the ‘tpm_rng’ kernel driver & setup rng-tools
  - Use /dev/(u)?random

Use case: Sealed Storage aka Local Attestation

Use TPM2 policy authentication as access control on TPM protected object

- Microsoft Bitlocker uses this mechanism for disk crypto keys
- OpenXT virtualization system uses similar mechanism
- Assumes measured boot records TCB in PCRs: software identity
  - Create TPM object holding auth data for disk crypto
  - Bind object to PCR policy: select PCRs based on TCB & requirements
  - On successful boot w/ PCRs in expected state, load object
  - Can be used to hold secrets for LUKS volumes

Use case: Attestation (1)

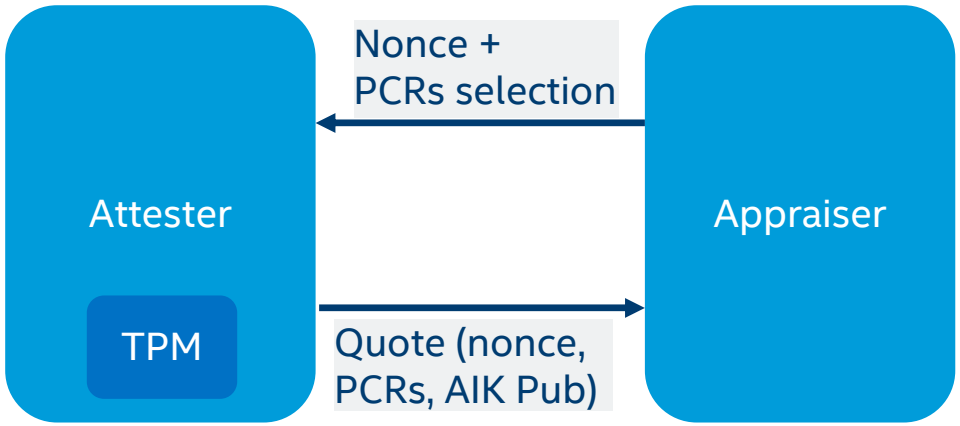
The presentation of verifiable evidence of software state to a remote party

- Software identity stored in PCRs: depends on correct measured boot!
- TPM Quote command produces signed report of PCR state
  - Can include arbitrary user data in quote (don't mix in Nonce!)
  - Signed using purpose specific key: attestation identity key
- Verifier challenges attester
  - Provides nonce (freshness)
  - Combined with hash of requested / negotiated PCRs in signed quote

Use case: Attestation (2)

Attestations are simple cryptographic operations over data (sign)

- "the Devil is in the details"
- Association between AIK & EK links AIK to platform
  - "privacy CA" as trusted 3rd party to protect anonymity of AIK
  - Enhanced Privacy ID (EPID)
- Deriving meaning from PCR state
  - Must reconstruct hash from event log
  - Map hash values to known software
  - No authoritative source for mapping



Implementation & Community

Intel implementing TCG TSS as Open Source

- Project hosted under '01.org' on Github
  - https://github.com/01org/tpm2.0-tss
  - https://github.com/01org/tpm2.0-tools
- 3-clause BSD == maximum flexibility
- Development on GitHub "in the open"
  - I don't always have the answer, someone else may though
  - Main development on 'master', tagged releases
  - Packages working their way into distros
- Lots of churn in the next few months

Embedded Builds

My personal OSS work

- meta-measured: https://github.com/flihp/meta-measured
  - TPM1.2 & 2.0 packages
  - Reference 'live' images & initrds
  - Grub2 patches extend measured launch (soon obsoleted by upstream!)
  - + BSP for Minnowboard Max to add TPM2 support as MACHINE_FEATURE
- Working on ARM reference platform + Infineon SPI TPM
  - Coreboot TPM2 support for chromebooks good starting place?
  - Still some work in TSS code to support big-endian systems (facepalm)

Shout-Outs!

Many thanks for contributions to materials:

- Monty Wiseman @ General Electric
- Lee Willson @ Security Innovation
- Andreas Fuchs @ Fraunhofer SIT

& Everyone who's contributed code / answered questions on GitHub!

- Bill Roberts @ Intel OTC
- Imran Desai @ Intel IOTG

THANKS!

Resources

Threat Modeling: Designing for Security – Adam Shostack

- http://www.wiley.com/WileyCDA/WileyTitle/productCd-1118809998.html

Trusted Platforms UEFI, PI and TCG-based firmware

- https://people.eecs.berkeley.edu/~kubitron/cs194-24/hand-outs/SF09_EFIS001_UEFI_PI_TCG_White_Paper.pdf

Open Security Training Trusted Computing Module:

- http://opensecuritytraining.info/IntroToTrustedComputing

Davide Guerri TPM2.0 talk @ FOSDEM

- https://fosdem.org/2017/schedule/event/tpm2/

TPM RNG linux howto:

- https://scotte.org/2015/07/TPM-for-better-random-entropy

Physical security & implications

- Tamper Resistant
  - Cast it in Epoxy
- Tamper Evident
  - Wrap it in “tamper tape”
- Tamper Responsive
  - Tamper detection mechanisms destroy secrets
- Physical security is $$$
- TPM designed to be cheap to promote adoption

Physical attacks against TPM

Several documented over last ~10 years

- LPC bus intercept / reset attack
  - Dartmouth College Computer Science Technical Report TR2007-597
  - http://www.cs.dartmouth.edu/~pkilab/sparks/
- Bus snooping largely addressed by new encrypted / HMAC sessions
- Chris Tarnovsky - Attacking TPM @ Defcon20
  - $200k in equipment + 6 months
  - https://www.youtube.com/watch?v=h-hohCfo4LA
  - https://www.defcon.org/html/links/dc-archives/dc-20-archive.html