Introduction to Trusted Computing: TPM 101

Ariel Segall
ariels@alum.mit.edu

Day 1

Approved for Public Release: 12-2749.
Distribution unlimited

License

All materials are licensed under a Creative Commons “Share Alike” license.

- http://creativecommons.org/licenses/by-sa/3.0

What We'll Be Covering

In this section:

- What is a TPM? What does it do?
- What's it good for?
- Some TPM myths (and the truths behind them)
- Why enterprises should care about TPMs

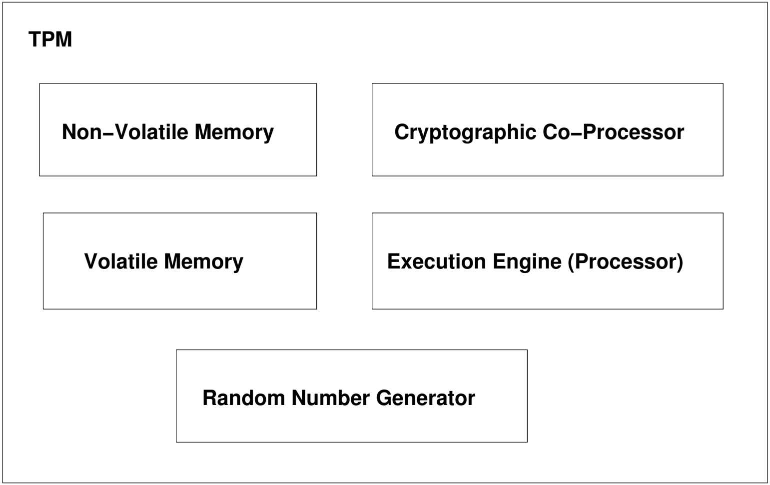
All at a high level– deep dive this afternoon.

What is a TPM?

- Trusted Platform Module
- Inexpensive (<$1, usually) chip on almost all motherboards today
  - Not in Macs
  - Only some servers have them– ask.
- Hardware basis for platform trust
  - In secrets
  - In platform state
    - Combined with a *Root of Trust for Measurement*[1]
  - In platform identity
- Current version is 1.2
  - Unless otherwise specified, we'll always refer to 1.2 TPMs
  - Previous version 1.1; next, 2.0.

[1] We'll get to these in a little while

What's In a TPM?



What TPMs Provide

- A *Root of Trust for Reporting*
- A *Root of Trust for Storage*
- Limited internal storage
  - *Platform Configuration Registers*
  - Key storage
  - Data storage
- Random number generation
- Highly constrained cryptographic functions
  - Feature, not bug (mostly)

Roots of Trust

We keep hitting this phrase: *Root of Trust*. What does it mean?

- The thing you base all other trust on
- Trusted inherently: no way to verify it directly
  - This is why standards are useful!
  - Out-of-band verification is your only option
  - Trust the chip because the manufacturer says it meets spec
  - Keep in mind the supply chain!
    - There are not currently any trusted foundries producing TPMs.
- No such thing as *generic* trust. Trust always has an associated verb!
  - I trust my electrician to repair wires, not update my bank account
  - I trust a TPM to protect my data, not to verify my antivirus

The Root of Trust for Reporting

Core question: “Is this system in a good state?”

Breaks down into two parts:

- What looked at the system state? *Root of Trust for Measurement*
- What told us the results? *Root of Trust for Reporting*

The TPM is a Root of Trust for Reporting (RTR);
it is **not** a Root of Trust for Measurement (RTM).

The Root of Trust for Storage

Core question: “Are my secrets kept secret?”

- The TPM is a Root of Trust for Storage (RTS)
- Does **not** store *all* secrets directly
- Stores one secret used to protect other secrets that may be outside
- Hence, *Root* of Trust.

What TPMs Provide

- A *Root of Trust for Reporting*
- A *Root of Trust for Storage*
- Limited internal storage
  - *Platform Configuration Registers*
  - Key storage
  - Data storage
- Random number generation
- Highly constrained cryptographic functions
  - Feature, not bug (mostly)

Platform Configuration Registers

The TPM has three kinds of internal storage. The one we'll talk about most are Platform Configuration Registers, or *PCRs*.

- Series of 20-byte registers (length of a SHA-1 hash)
- Most modern TPMs have 24; older ones have 16
- Used to store system measurements
    - Although they can be more flexible than that!
- Highly constrained behavior
    - Always reset to a known value at boot
    - Only store data using an *Extend* operation
    - **Extend: hash new data with current contents**
    - Permissions based on *locality*; similar to OS rings
        - Can never be freely overwritten
        - Verifier can determine every value extended in
        - Easy to check; computationally infeasible to forge

The Roots of Trust for Measurement

Core question the TPM can't meet: "What is the state of the system?"

- TPM has no visibility outside itself!
- RTM must be capable of inspecting system.
- Two current RTM options:
  - BIOS (technically, BIOS boot block)
    - Also known as the *Static Root of Trust for Measurement* or SRTM
  - Special CPU code operating in trusted mode
    - *Dynamic Root of Trust for Measurement*, or DRTM
    - Intel: Trusted Execution Technology (TXT)
    - AMD: Secure Virtual Machine (SVM)
- Place initial measurements into PCRs before handing off control

Reporting PCR Measurements

	0	1	2	...	24
PCRs	936AA11BA2F7...	44852FBD5D44...	A89FB8F88CAA...		000000000000...

We have system measurements in our PCRs; how do we use them?

- Can be read directly, but not trustworthy!
  - If unsigned, just report from software about software
- Instead, request a **Quote**:
  - Signed report from TPM
  - Contains hash of current PCR values
  - Uses nonce (created by requestor) to prove freshness
- Quotes can be provided to other parties for PCR verification
  - Trustworthy, remote state reporting!

Other Uses of PCRs

We can also use the TPM's PCRs in other ways.

- Encrypted data can be *sealed* or *bound* to a set of PCR values
  - Decryptable only when current values match target
- Keys can be constrained to a set of PCR values
  - Key only usable when values match
- Non-measurement data can be stored in PCRs
  - We'll get to use cases for this later.

What TPMs Provide

- A *Root of Trust for Reporting*
- A *Root of Trust for Storage*
- Limited internal storage
  - *Platform Configuration Registers*
  - Key storage
  - Data storage
- Random number generation
- Highly constrained cryptographic functions
  - Feature, not bug (mostly)

TPM Root Keys

There are only two keys that never leave the TPM:

- **Endorsement Key (EK)**: The key that the TPM uses in its role as Root of Trust for Reporting.
    - Only used directly to certify Identity Keys (AIKs), which we'll get to soon.
    - Critical: trust in all keys in the system come down to trust in EK
- **Storage Root Key (SRK)**: The key that the TPM uses in its role as Root of Trust for Storage.
    - Used to protect other keys and data via encryption
    - Can protect other storage keys: heirarchy of protection

All other keys created by the TPM have their private halves encrypted by the SRK (or another storage key), and are stored outside the TPM.

Non-Root Keys (1/2)

All TPM keys are RSA keys, but have specialized roles:

- Encryption/Decryption: Storage, Binding
- Signing/Reporting: Identity, Signing
  - Identity keys better known as *Attestation Identity Keys*, or *AIKs*
- Legacy keys can be used for either, but are not created by the TPM
  - TPMs can import keys; less secure, but sometimes useful

We'll cover the details of when to use which later today.

Non-Root Keys (2/2)

- Keys are stored in "blobs"[2] on disk (outside TPM)
- Private key encrypted; integrity protection on other data
- Only decryptable by TPM that created it, unless explicitly created otherwise
  - Local-only keys are *non-migratable*
  - Keys that can be exported off of the machine are *migratable*
- Loaded back into the TPM for use
- Remain in the TPM while space allows, or until reboot
  - TPMs have limited amount of internal space for keys!
  - Owner[3] can set a particular key to remain in the TPM

[2] Yes, that's the technical term

[3] We'll get to owners shortly

Data Storage

- TPMs have a limited amount of non-volatile storage (*NVRAM*)
  - Non-volatile because (unlike most TPM data) remains between boots
- Access can be controlled (read and write separately)
  - Owner
  - PCR values
  - Authorization value (password)
- Part of NVRAM set aside for certificate storage
  - Manufacturer may supply credentials for TPM
    - ...but they probably didn't.

What They Provide

- A *Root of Trust for Reporting*
- A *Root of Trust for Storage*
- Limited internal storage
  - *Platform Configuration Registers*
  - Key storage
  - Data storage
- Random number generation
- Highly constrained cryptographic functions
  - Feature, not bug (mostly)

Random Number Generator

- TPMs required to have internal random number generator
- Spec strongly encourages but does not require hardware entropy source
  - Quality of entropy not defined in spec!
  - Suitable for most day-to-day purposes, but may not meet high security requirements
  - Externally generated entropy can be added into the TPM RNG
- RNG used to generate all TPM keys and nonces
- Can also provide random bits directly to user on request

Cryptographic Functions

The TPM provides several specialized cryptographic functions:

- Encryption/Decryption
  - Seal/Unseal: Encrypt/decrypt data for local TPM
  - Unbind: Decrypt data from anywhere (no TPM required to encrypt)
- Sign
  - Constrained data formats: SHA-1, DER, custom TPM structure
  - NOTE: attack exists on custom TPM structure; do not use.
- Key certification
  - TPM can certify any key it creates
  - Custom format; includes all key properties
- SHA-1 hash generation

Why so specialized? Two reasons:

- Prevent attacks resulting from key misuse
- Make it possible to verify constraints

Other TPM Functions

- Monotonic Counter
  - Always increases; good for rollback prevention
- Tick Counter
  - Not quite a clock, but useful for timing
- Direct Anonymous Attestation (DAA)
  - Zero-knowledge proof of identity
  - Extremely complicated!
  - Added to address privacy concerns

These will not be covered in detail in this class.

TPM Functionality: Quick Review

- A *Root of Trust for Reporting*
- A *Root of Trust for Storage*
- Limited internal storage
  - *Platform Configuration Registers*
  - Key storage
  - Data storage
- Random number generation
- Highly constrained cryptographic functions
  - State reporting
  - Data protection
  - Cryptographic utilities (e.g., signing)

Ownership: Whose TPM Is It Anyway?

- The TPM has a single *owner*.
    - Usually the machine owner (IT dept in corporate setting)
- Someone must take ownership for the TPM to be used!
- Anyone with physical presence can take ownership
- SRK is created when ownership is taken; if replaced, old key erased
- Owner has admin privileges; e.g. can change TPM configuration settings
- Owner has exclusive right to create TPM identities
    - Users can freely create other keys unless SRK requires authorization
- Owner does **NOT** automatically get access to resources
    - TPM ownership is not like root or administrator access in OS

What We'll Be Covering

In this section:

- What is a TPM? What does it do?
- What's it good for?
- Some TPM myths (and the truths behind them)
- Why enterprises should care about TPMs

All at a high level– deep dive this afternoon.

What's it good for?

The TPM's big benefits:

- Machine Authentication
- Attestation
- Data Protection

Machine Authentication

- We can use TPM keys to reliably identify a machine!
  - TPM soldered to motherboard
  - Keys cryptographically bound to a particular TPM
- Signing-based authentication
  - This data passed through machine X
  - (Note: Can't prove origination with just a signature)
- Decryption-based authentication
  - Only machine X can read this data
- One of the simplest TPM applications

Attestation

Attestation: *the presentation of verifiable evidence about machine state to a remote party*

- Quotes are all about attestation!
    - Signed report of current PCR contents
    - Many PCR constraints (e.g. keys) can be used for attestation also
- Remote verifier can check boot state of machine
- Potentially very powerful!
    - Is this machine running the right image?
    - Is the software trustworthy?
- Easier said than done:
    - Interpreting PCR values is **hard**
    - Work to regularize them is ongoing
    - Values are very fragile and hard to predict!

Data Protection

- TPM is **not** suitable for bulk data encryption
  - Too slow! Public key encryption only, cheap processor
  - No fast symmetric ciphers due to export regulations
- Use to encrypt small, high-value data; for example:
  - Software-held private keys (e.g. user identities)
  - Symmetric keys usable for bulk encryption
  - Password stores
- Can be used for hard drive encryption if supported
  - TPM-sealed symmetric key encrypts drive
  - Bitlocker option!
- Provide hardware protection, tamper resistance to sensitive data

What We'll Be Covering

In this section:

- What is a TPM? What does it do?
- What's it good for?
- Some TPM myths (and the truths behind them)
- Why enterprises should care about TPMs

All at a high level– deep dive this afternoon.

TPM Myths

There are many common **misconceptions** about the TPM.

- Some are misunderstandings based on early marketing materials
- Most are the result of simplified summaries of a very complicated topic

We'll debunk a few of the most common, and talk about the truths behind the myths.

Myth: The TPM Controls Boot

It can stop your machine from booting if bad software is running.

- The TPM has no control over the rest of your machine; it's a purely passive device.
- Nor does it have any awareness of what's happening on the system beyond what measurement software tells it.
- The TPM *can*, in **highly controlled situations**, limit data access to only good software; but this is fragile.
- High-security, predictable systems designed with this in mind can use the TPM to limit bad boots.
  - Bitlocker
  - TPM-enabled device encryption
- Note: Does *not* stop machine from booting; just protects data.

Myth: The TPM is Tamper-Proof

- TPMs are tamper-*resistant*. . . for consumer products.
- Tremendously good for their cost!
  - Cost $<$ \$1
  - Breaking cost researcher $>$\$100,000; destroyed several in the process
- **Not** designed with government tamper-resistance standards in mind.

Myth: The TPM Works for Disney/Microsoft/etc.

- Grew out of early TPM publicity
  - Originally pitched for digital rights management
  - Not actually the best use
- TPM belongs to the machine owner!
  - In enterprise setting, usually IT department
  - Owner can turn on/off
  - Owner can control identities TPM uses
- This myth is one reason TPM has so many privacy features.

Myth: You Can Delegate All Crypto To The TPM

- Many people want the TPM to be a general cryptographic coprocessor, but:
- It's highly constrained
  - Generally a good thing– prevents many attacks
  - Can't be dropped in for every application
- It's *very* slow
  - Priority is cost, not performance
  - High-speed applications like packet signing: right out

What We'll Be Covering

In this section:

- What is a TPM? What does it do?
- What's it good for?
- Some TPM myths (and the truths behind them)
- Why enterprises should care about TPMs

All at a high level– deep dive this afternoon.

Why Should Enterprises Care?

- TPMs are everywhere
  - Already in almost all enterprise machines
- No additional cost to acquire
  - Although integration isn't free– we'll talk about that more
- Very good return on investment for security
  - Software trust $\rightarrow$ hardware
  - Some tamper resistance better than nothing!

Reminder: The TPM's Benefits

Earlier, we talked about the TPM's big benefits:

- Machine Authentication
- Attestation
- Data Protection

Each of these are directly applicable to enterprises.

Enterprise Machine Authentication

Enterprises often want to know the identity of machines on their network.

- Network Access Control: should a machine be allowed to connect?
- Audit trails: Which machine did this data come from?
- Authorization: Is this request coming from an expected machine?
  - Particularly useful for sensitive data
- Smartcard replacement: machine instead of user ID

Enterprise Attestation

Today's enterprise security approach in a nutshell:
ask the software if the software should be trusted

- TPM-rooted attestation gives us noticably more assurance
- Software cannot fake "good" measurement or use old one
- RTMs below the level a rootkit can interfere with
  - We'll talk about the details and other threats shortly
- Machine authentication use cases + state
  - Not just which machine, but what software
- Particularly valuable when combined with sw reporting tools
  - Check if antivirus is good before believing its report

Note: Full promise of these capabilities not yet available

Enterprise Data Protection

- Generally, TPM not providing *new* capability
- Better assurance over existing solutions
- TPMs more tamper-resistant than most smartcards
- TPMs far more tamper-resistant than software encryption solutions
- Hardware-tied keys mean adversary cannot steal
  - Noticable improvement over purely software keys and certs
  - Note: adversary with machine access can use, but difficulty raised

Review: What We Covered

- What is a TPM? What does it do?
- What's it good for?
- Some TPM myths (and the truths behind them)
- Why enterprises should care about TPMs

Review: TPM Functionality

- A *Root of Trust for Reporting*
- A *Root of Trust for Storage*
- Limited internal storage
  - *Platform Configuration Registers*
  - Key storage
  - Data storage
- Random number generation
- Highly constrained cryptographic functions
  - State reporting
  - Data protection
  - Cryptographic utilities (e.g., signing)

Review: The TPM's Benefits

- Machine Authentication
- Attestation
- Data Protection

Questions?

Next up: Other key trusted computing technologies