TPM Provisioning

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

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

to make commercial use of the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

What We'll Be Covering

- What do we mean by provisioning?
- Turning on the TPM
- The Endorsement Key: Theory vs. Reality
- Provisioning TPM Keys
- Certifying the EK (and other variations)

What is TPM Provisioning?

In short: getting the TPM ready to use.

Several components; not all required in all cases

- Turning the TPM on
- Making sure it has an Endorsement Key
- Making sure it has an Endorsement Credential
- Taking ownership
- Creating any initial keys
- Certifying these keys

Why Provisioning Matters

We can do all kinds of wonderful things by rooting trust in hardware. But:

How do we know we're actually communicating with the TPM?

Provisioning is how we establish trust in the TPM itself.

If provisioning is not done properly, an adversary can undetectably pretend to be our TPM.

Where Trust is Established

- We prove that we are talking to the TPM using keys.
- All of our trust is rooted in **the association of a key with the TPM**.
- The TPM has capabilities for certifying new keys, given a root key.
- **Security is essential when provisioning and certifying the root key**.
- Other steps in the provisioning process, like turning on the TPM, are less sensitive.
  - If we're wrong about having turned it on, just a DoS
  - No loss of trust in the TPM

What We'll Be Covering

- What do we mean by provisioning?
- Turning on the TPM
- The Endorsement Key: Theory vs. Reality
- Provisioning TPM Keys
- Certifying the EK (and other variations)

Turning on the TPM

- TPMs are turned on in the BIOS menu.
- Each BIOS varies in location
  - Common: “Security”, “TPM Security”, “Security Chip”
- TPM technically has two different versions of “on/off”:
  - Activated/Deactivated
  - Enabled/Disabled
- To use the TPM, it must be both Activated and Enabled.
  - Some BIOSes group these into a single option.
- A few systems require multiple reboots to turn the TPM on.

Aside: Clearing the TPM

- Some BIOSes also provide a **Clear** option for the TPM
- Clearing the TPM erases the Storage Root Key and owner
  - This makes all keys and encrypted data useless!
- Normally used before transferring machine to new owner
- Some BIOSes automatically turn the TPM off after clearing.

If your BIOS offers a Clear option, adding a BIOS password reduces the risk of an accidental DoS.

Demo: Turning on a TPM in the BIOS

What We'll Be Covering

- What do we mean by provisioning?
- Turning on the TPM
- The Endorsement Key: Theory vs. Reality
- Provisioning TPM Keys
- Certifying the EK (and other variations)

Review and Expand: Endorsement Key

- EK is root key for reporting
- TPM's unique identifier
- Only directly used to establish trust in TPM identities
  - ..but identities certify all other keys and sign TPM reports
- Source of all remote trust in TPM
  - "EK belongs to good TPM; key K in same TPM as EK; so trust K"
  - "K belongs to good TPM, so PCR quote is reliable"
  - All comes back to EK!
- Generally last lifetime of TPM
  - Revokable EKs are in spec, but not always implemented

The Endorsement Key, According to the TCG

According to the TPM spec:

- The TPM manufacturer creates the TPM's EK as part of manufacturing
  - Unique to each TPM and secret; process not specified
- Each TPM is shipped with an *Endorsement Credential (EC)* in NVRAM
  - Signed by manufacturer
  - Claim: "I created this TPM, and this is its root key"
  - Anyone can verify to trust TPM if they trust manufacturer
- TPMs can be trusted immediately, by tracing keys back to EK

The Endorsement Key, According to the TCG

According to the TPM spec:

- The TPM manufacturer creates the TPM's EK as part of manufacturing
  - Unique to each TPM and secret; process not specified
- Each TPM is shipped with an *Endorsement Credential (EC)* in NVRAM
  - Signed by manufacturer
  - Claim: "I created this TPM, and this is its root key"
  - Anyone can verify to trust TPM if they trust manufacturer
- TPMs can be trusted immediately, by tracing keys back to EK

Reality is somewhat different.

The Endorsement Key, Today

- Many TPM manufacturers do not include an EK at all.
- Most TPM manufacturers that do include an EK do not include EC
- If EC included, verification process unclear

For today's machines, we can't rely on the TCG process to establish EK trust.

The Endorsement Key, Today

- Many TPM manufacturers do not include an EK at all.
  - **EK must be created during provisioning.**
- Most TPM manufacturers that do include an EK do not include EC
  - **New or not, EK must be certified during provisioning**
- If EC included, verification process unclear

For today's machines, we can't rely on the TCG process to establish EK trust.

What We'll Be Covering

- What do we mean by provisioning?
- Turning on the TPM
- The Endorsement Key: Theory vs. Reality
- Provisioning TPM Keys
- Certifying the EK (and other variations)

The Provisioning Process

In brief:

1. Establish trusted environment
2. Create EK, if necessary
3. Record public EK for later certification
4. Take ownership of TPM
5. (Optional) Create additional TPM keys; save for later use
6. (Optional) Record public half of additional keys for later certification

Establishing a Trusted Environment (1/2)

- Threat to avoid: malware or adversary masquerading as TPM
  - Low risk, catastrophic damage– all future trust undermined
- Ideal: minimal, trusted, software; no network access
  - Boot CD with minimal Linux, no network drivers
  - Data transferred off via writeable CD
- Enterprises sometimes require compromises
  - Live CD requires technician running program on every machine: high trust, high cost, hard to scale
  - Script remotely run on windows: extremely low trust, but fast and scalable

Establishing a Trusted Environment (2/2)

Recommendation:

- Use trusted process when machines first acquired
  - IT department installing software anyway
  - Low additional overhead, trained personnel
- Use lower-security process for most machines in field
  - Average machine has lifespan of only a few years
  - Threat often not a big enough concern to warrant cost
- Use high-security on-site process for critical machines
  - Limited set; reduced scaling problems
  - Maintain maximal trust where it matters most
- **Certify with different keys, so low-trust machines can be phased out easily over time**

The Provisioning Process

In brief:

1. Establish trusted environment ✓
2. Create EK, if necessary
3. Record public EK for later certification
4. Take ownership of TPM
5. (Optional) Create additional TPM keys; save for later use
6. (Optional) Record public half of additional keys for later certification

Creating the Endorsement Key

- `TPM_CreateEndorsementKeyPair`
  - Some platforms provide more user-friendly tools
    - `tpm_createek` command line tool in linux tpm-tools package
  - Creates EK permanent for life of TPM
  - **Usually used**
- `TPM_CreateRevocableEK`
  - Optional command; some TPMs may not support it
  - No convenient preexisting utilities
  - Creates EK which can be revoked using authorization set at creation
  - Tradeoffs: more control over TPM history, but opens DoS avenue
- Either command will produce an error if an EK already exists.

The Provisioning Process

In brief:

1. Establish trusted environment ✓
2. Create EK, if necessary ✓
3. Record public EK for later certification
4. Take ownership of TPM
5. (Optional) Create additional TPM keys; save for later use
6. (Optional) Record public half of additional keys for later certification

Retrieving the Public EK

- `TPM_ReadPubek`
  - Retrieves public portion of Endorsement key
    - No way to retrieve private portion!
  - Must be executed before ownership is taken!
    - The public EK can be read later, but much more complicated
- Public EK must be saved for transfer to CA
- **Recommend human write down key fingerprint for verification**
  - Or other out-of-band mechanism to make sure saved key is the one certified
- Note: Public EK is (as name suggests) not secret: we only care about integrity

The Provisioning Process

In brief:

1. Establish trusted environment ✓
2. Create EK, if necessary ✓
3. Record public EK for later certification ✓
4. Take ownership of TPM
5. (Optional) Create additional TPM keys; save for later use
6. (Optional) Record public half of additional keys for later certification

Taking Ownership (1/2)

When taking ownership, two authorization values (passwords) are set:

- An owner authorization value
  - Used to change TPM configuration, create identities
  - Less security critical than, e.g., root password
  - Enterprises may wish to use standard values to simplify management
  - Owner privileges can be individually delegated if access needed
- A SRK authorization value
  - Called for whenever the SRK is used. . . which is often!
  - Unless you're doing something unusual, *every time you load another TPM key*.
  - **Strongly recommend using the well-known secret**
    - Effectively, no password.
  - If you want to protect data with a password, create another key later.

Taking Ownership (2/2)

- `TPM_TakeOwnership`
- Linux utility tpm_takeownership exists in tpm-tools package
- Windows 7 has a utility that will enable the TPM and take ownership, but:
  - **There are reports that taking ownership with the Windows utility will result in a TPM unable to be used by anything except Bitlocker.**
- Taking ownership creates the SRK and set the owner authorization.
- Owner remains until TPM is cleared, although auth can be changed.

The Provisioning Process

In brief:

1. Establish trusted environment ✓
2. Create EK, if necessary ✓
3. Record public EK for later certification ✓
4. Take ownership of TPM ✓
5. (Optional) Create additional TPM keys; save for later use
6. (Optional) Record public half of additional keys for later certification

Creating Additional TPM Keys

How of key creation later. For now: Why?

- Creating keys during provisioning can be practical
  - Identity keys require owner approval; easiest now if owner is IT
  - Can create and put in standard locations for apps
  - Can make sure all users have standard utility set
- Can shortcut key certification
  - *Not recommended long-term approach!* Not scalable.
  - Key certification complicated– more on that later
  - Already doing direct certification of EK; can certify other keys as well with no decrease in trust
  - *In this case*, public halves of keys should be recorded for certification
  - As with the EK, a fingerprint or other verification mechanism recommended.
- No keys other than EK, SRK required!

The Provisioning Process

In brief:

1. Establish trusted environment ✓
2. Create EK, if necessary ✓
3. Record public EK for later certification ✓
4. Take ownership of TPM ✓
5. (Optional) Create additional TPM keys; save for later use ✓
6. (Optional) Record public half of additional keys for later certification ✓

What We'll Be Covering

- What do we mean by provisioning?
- Turning on the TPM
- The Endorsement Key: Theory vs. Reality
- Provisioning TPM Keys
- Certifying the EK (and other variations)

Certifying the EK

We currently have:

- A TPM with an uncertified EK
- A public EK with verification mechanism

We want to create an Endorsement Credential:

- Certificate claiming EK is Endorsement Key of legit TPM
- Signed by enterprise CA
- Containing relevant info about TPM
  - e.g., machine identifier, TPM manufacturer, version...

Challenges of EK Certification

- Most commercial CAs expect x.509 certificate signing requests
  - Self-signed request
  - EK not capable of signing request! (Nor are most other TPM keys.)
  - Need to update CA to accept request based on good process
    - No actual security loss- self-signing adds nothing to trust
- Certification, like provisioning, needs to guarantee association
  - Is the public EK being certified the same was created?
  - If sent over network, rewritable media: verify integrity

Otherwise, certification is pretty standard.

Certifying Non-EK Keys

- Can establish trust in other provisioned keys in same way as EK
- Same challenges apply!
- Certificates should clearly establish key type
  - Should not be possible to mistake for EK!
- If multiple certification mechanisms used (more on this soon), distinguish provisioning-certified keys from cryptographically certified keys
  - Just good practice! Allows better revocation if needed.

Provisioning Review

- Process to establish initial trust in TPM
- Performed in trusted environment for security
- Create and certify Endorsement Key
- Take ownership, creating SRK
- Optionally create other TPM keys

Questions?