**Provisioning Boot**

*(differences with normal boot below are in red)*

1. The Hypervisor builds/starts the Domain Builder (from an image in ramdisk).
2. The Domain Builder builds/starts the vTPM Manager (from an image in ramdisk).
3. The vTPM Manager infers that this is a provisioning boot, presumably from the absence of
   1. encrypted vTPM Manager data,
   2. wrapped encryption key K,
   3. wrapped K2.

The vTPM Manager:

1. Creates its initial (empty) data in RAM.
2. Uses the TPM to generate K2, wrapped to the current PCRs 0-4.
3. Stores a hash of the wrapped K2 into TPM NVM.
4. The Domain Builder builds/starts the SVP Controller (from an image in ramdisk), with some information on how to access its Schema (in ramdisk).
5. The Domain Builder sends to the vTPM Manager the hash of the Controller + Schema and the domain ID of the Controller, which the vTPM Manager records internally.
6. The Controller asks the Domain Builder to build/start the Host Storage (from an image in ramdisk).
7. The Controller presumably sees from its Schema that this is a provisioning boot, so it chooses a LTN for the SVP vTPM -- how is this LTN chosen? generated? read from somewhere? is the user prompted at the terminal? is there a terminal running now?
8. The Controller asks the Domain Builder to build/start the SVP vTPM with the chosen LTN.
9. The Domain Builder builds/starts the SVP vTPM (from an image in Host Storage).
10. The Domain Builder sends to the vTPM Manager
    1. the LTN of the vTPM,
    2. the domain ID of the vTPM,
    3. the domain ID of the Controller,
    4. the hash of the vTPM image + LTN.

The vTPM Manager records this information. The vTPM Manager creates an entry in the table for the SVP vTPM: the entry is complete except for the sealed K3.

1. The vTPM, started in step 9, asks the vTPM Manager for the vTPM's data encryption key. Since the key is not there yet, the vTPM Manager will respond something like "no key yet", and from that the vTPM will infer that this is a provisioning boot. The vTPM:
   1. Initializes its own data in RAM.
   2. Generates a new symmetric encryption key K3.
   3. Encrypts its data.
   4. Saves its encrypted data to Host Storage.
   5. Sends K3 to the vTPM Manager.

This generation of K3, encryption and saving of the data, and sending of K3 to the vTPM Manager, is the normal process that happens whenever the (non-volatile) vTPM data changes.

1. The vTPM Manager:
   1. Receives K3 from the vTPM.
   2. Uses the domain ID of the vTPM to find the corresponding entry in the table.
   3. Uses the TPM to seal K3 to (i) the hash of vTPM image + LTN (read from the table) and (ii) the hash of Controller image + Schema (looked up using the domain ID of the Controller).
   4. Updates the table entry with the sealed K3.

This step is the normal process by which the vTPM Manager updates the K3 of a vTPM.

The provisioning boot for a UVP could be analogous to steps 4, 5, 7-12.

**Normal Boot**

*(differences with provisioning boot above are in red)*

1. The Hypervisor builds/starts the Domain Builder (from an image in ramdisk).
2. The Domain Builder builds/starts the vTPM Manager (from an image in ramdisk).
3. The vTPM Manager gets (from ramdisk)
   1. encrypted vTPM Manager data,
   2. wrapped encryption key K,
   3. wrapped K2.

The vTPM Manager:

1. Hashes the wrapped K2 and compares it with the hashed wrapped K2 that is stored in the TPM NVM.
2. Uses the TPM to unwrap K2.
3. Uses the TPM to unwrap K.
4. Decrypts vTPM Manager data using K.
5. The Domain Builder builds/starts the SVP Controller (from an image in ramdisk), with some information on how to access its Schema (in ramdisk).
6. The Domain Builder sends to the vTPM Manager the hash of the Controller + Schema and the domain ID of the Controller, which the vTPM Manager records internally.
7. The Controller asks the Domain Builder to build/start the Host Storage (from an image in ramdisk).
8. The Controller reads from its Schema the LTN of the SVP vTPM. When was the LTN added to the Schema?
9. The Controller asks the Domain Builder to build/start the SVP vTPM with the chosen LTN.
10. The Domain Builder builds/starts the SVP vTPM (from an image in Host Storage).
11. The Domain Builder sends to the vTPM Manager
    1. the LTN of the vTPM,
    2. the domain ID of the vTPM,
    3. the domain ID of the Controller,
    4. the hash of the vTPM image + LTN.

The vTPM Manager records this information. The vTPM Manager uses the LTN to look up the corresponding entry in the table, and it adds to the entry the domain ID of the vTPM, the domain ID of the Controller, and the hash of the vTPM image + LTN.

1. The vTPM, started in step 9, asks the vTPM Manager for the vTPM's data encryption key.
2. The vTPM Manager:
   1. Finds the correct table entry based on the domain ID of the requesting vTPM.
   2. Uses the TPM to unseal K3 using (i) the hash of vTPM image + LTN (read from the table) and (ii) the hash of Controller image + Schema (looked up using the domain ID of the Controller).
   3. Sends K3 to the vTPM.
3. The vTPM receives K3 and decrypts its own data.

The normal boot for UVP could be analogous to steps 4, 5, 7-13.

**vTPM Manager Data**

* non-volatile
  + set of vTPM entries, each consisting of
    - LTN of vTPM
    - sealed K3
* volatile
  + set of vTPM entries, each consisting of
    - LTN of vTPM (read from non-volatile)
    - sealed K3 (read from non-volatile)
    - domain ID of vTPM
    - domain ID of Controller
    - hash of vTPM image + LTN
  + set of Controller entries, each consisting of
    - domain ID of Controller
    - hash of Controller image + Schema

**Additional Questions**

* If some Controller Schema changes, how is the sealed K3 in the vTPM Manager table updated? (recall that Domain Builder and vTPMs are the only clients of the vTPM Manager)
* Confirm that “vTPM hash” includes not only image but also LTN.
* If hash of Controller image + Schema is not unique, could a VP start another VP’s vTPM? For example, could a UVP start the vTPM of another UVP?