vTPM Modeling Notes

Perry Alexander and Brigid Halling
Information and Telecommunication Technology Center
The University of Kansas
{palexand,bhalling}@ku.edu

Allen Goldberg, Eric Smith, Alessandro Coglio
The Kestrel Institute
{addresses}@kestrel.edu

May 31, 2013

Contents

List of Figures

List of Tables

Abstract

This document is designed to document our work understanding the interaction among elements of the vTPM infrastructure and interaction between the vTPM and its operational environment.

1 Introduction

2 vTPM Manager Data

Data objects used by the vTPM subsystem are listed in figure 1 and figure 2. Figure 1 lists data that is stored by the vTPM Manager and includes PCR contents associated with vTPM and controller measurements, long- and short-term IDs, the sealed vTPM Data key and vTPM Data hash.

Field	*Description*	*Initialized*	*Updated*
ID_{vtpm}	The persistent name of the vTPM	provisioning	never
ID_{dom}	Domain ID of vTPM VM	build	build
$\#D$	vTPM data hash value	provisioning	vTPM sleep
PCR_6	Hash of the vTPM + Long Term Name	provisioning	boot
PCR_5	Hash of the platform controller	provisioning	boot
$seal(K_D, \{PCR_5, PCR_6\})$	Sealed symmetric key encrypting vTPM data	provisioning	vTPM sleep

Figure 1: vTPM Information Table contents

Figure 2 lists data that is involved in vTPM related communications and includes the encrypted vTPM data, the vTPM image, and hashes of the controller and booted vTPM. Note that the hash values are called out separately here as they may differ from what is stored in the vTPM Manager.

Data	*Description*	*Location*	*Initialized*	*Modified*
$\{D\}_{K_D}$	Encrypted vTPM data	Host Storage	provisioning	vTPM sleep
vtpm	*vTPM Image*	Host Storage	provisioning	never
$\#(vTPM + ID_{vtpm})$	Hash of vTPM named ID_{vtpm}	vTPM Manager	build	build
$\#Controller$	Hash of Controller and Schema	vTPM Manager	build	build

Figure 2: Other data used during vTPM activities

3 Provisioning Sequence

I'm not at all sure I have the provisioning sequence right. The vTPM can be used to generate an EK while the TPM_TakeOwnership command can be used to generate an SRK after startup. What else is initialized? Monotonic counter values and other stuff not specified here.

1. Platform booted through Controller – PCR_5 is now known by the vTPM Manager
2. A new Long Term Name is generated for the new vTPM
3. The new vTPM is built using the Domain Builder – PCR_6 is now known by the vTPM Manager
4. vTPM data is initialized – including fresh EK value.
5. vTPM is put to sleep using the standard sleep protocol

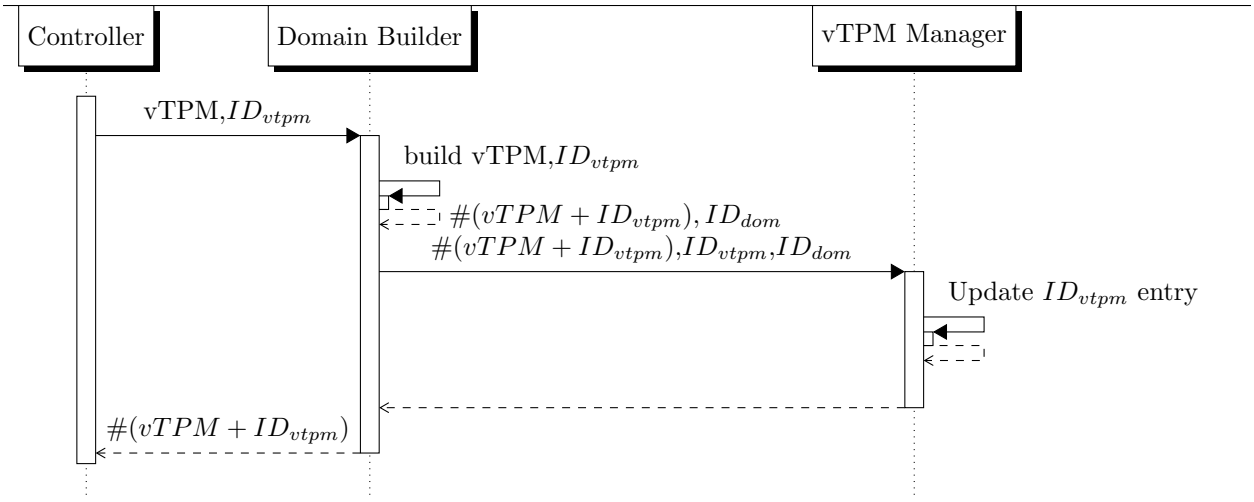

Figure 3: Command sequence to establish a running vTPM without its data.

4 Startup Sequence

Establishing an operational vTPM involves getting the vTPM up and running, then installing its data. These are outlined in the following to subsections.

4.1 vTPM Running

Figure 3 describes the sequence of events necessary for a vTPM to start. The result of this sequence is a vTPM running and the vTPM Manager table reflecting vTPM information. The vTPM will not have its data yet. Following is a textual description of the process:

1. Controller requests a build by sending a manifest identifying the vTPM image and its Long Term Name to the Domain Builder
2. Domain Builder hashes the vTPM image and Long Term Name
3. Domain Builder starts the vTPM VM resulting in a Domain ID
4. Domain Builder sends the Long Term Name, Hash, and Domain ID to the vTPM Manager
5. vTPM Manager updates data associated with the Long Term Name in the vTPM information table

4.2 vTPM Data Initialization

Figure 4 describes the sequence of events necessary for a vTPM to load its data. The result of this sequence is a vTPM ready to provide TPM services to its associated virtual platform. Following is a textual description of the process:

1. vTPM requests its data key from the vTPM Manager
2. vTPM Manager uses the vTPM Domain ID to find table entry
3. vTPM Manager resets PCR_5 and PCR_6 with table entry values[1]
4. vTPM Manager attempts to unseal K_D

[1]This happens whenever the vTPM Manager communicates with a vTPM.

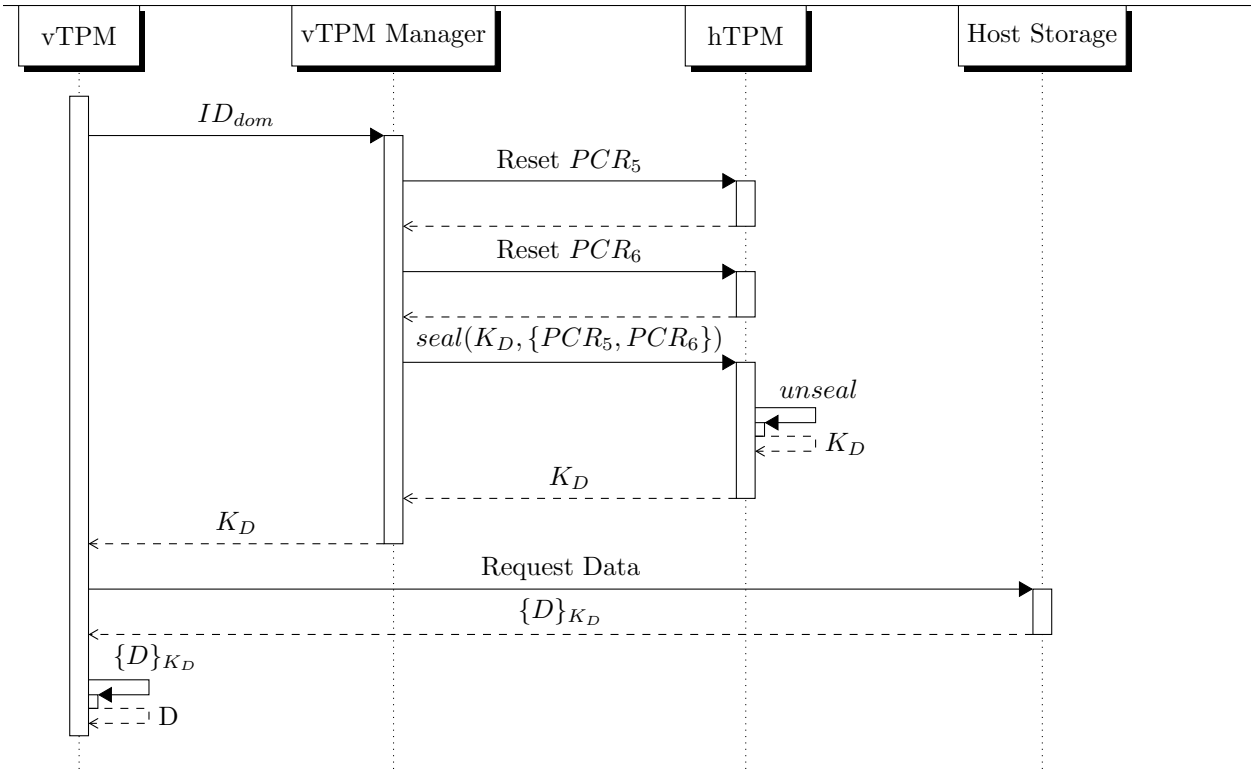

Figure 4: Command sequence to load vTPM data.

(a) On failure abort the request response
(b) On success return K_D and $\#D$ to vTPM

5. vTPM requests its encrypted data from Host Storage
6. vTPM decrypts its data with K_D and checks hash against $\#D$
7. vTPM installs data and begins providing services.

5 Sleep Sequence

Figure 5 describes the sequence of events necessary for putting a vTPM to sleep. The result of this sequence is the vTPM's data saved to Host Storage encrypted with a fresh session key. The vTPM Manager's table will be updated to reflect the new key and data hash. Following is a textual description of the process:

1. vTPM generates a fresh session key, K_D
2. vTPM hashes its data, $\#D$ and encrypts with K_D
3. vTPM stores $\{D\}_{K_D}$ in Host Storage
4. vTPM sends sleep request to vTPM manager with K_D and $\#D$
5. vTPM Manager uses vTPM Domain ID to find table entry
6. vTPM Manager resets and loads PCR_5 and PCR_6 from table entry
7. vTPM Manager seals K_D to TPM state
8. vTPM Manager updates its table with $seal(K_D, \{PCR_5, PCR_6\})$ and $\#K_D$

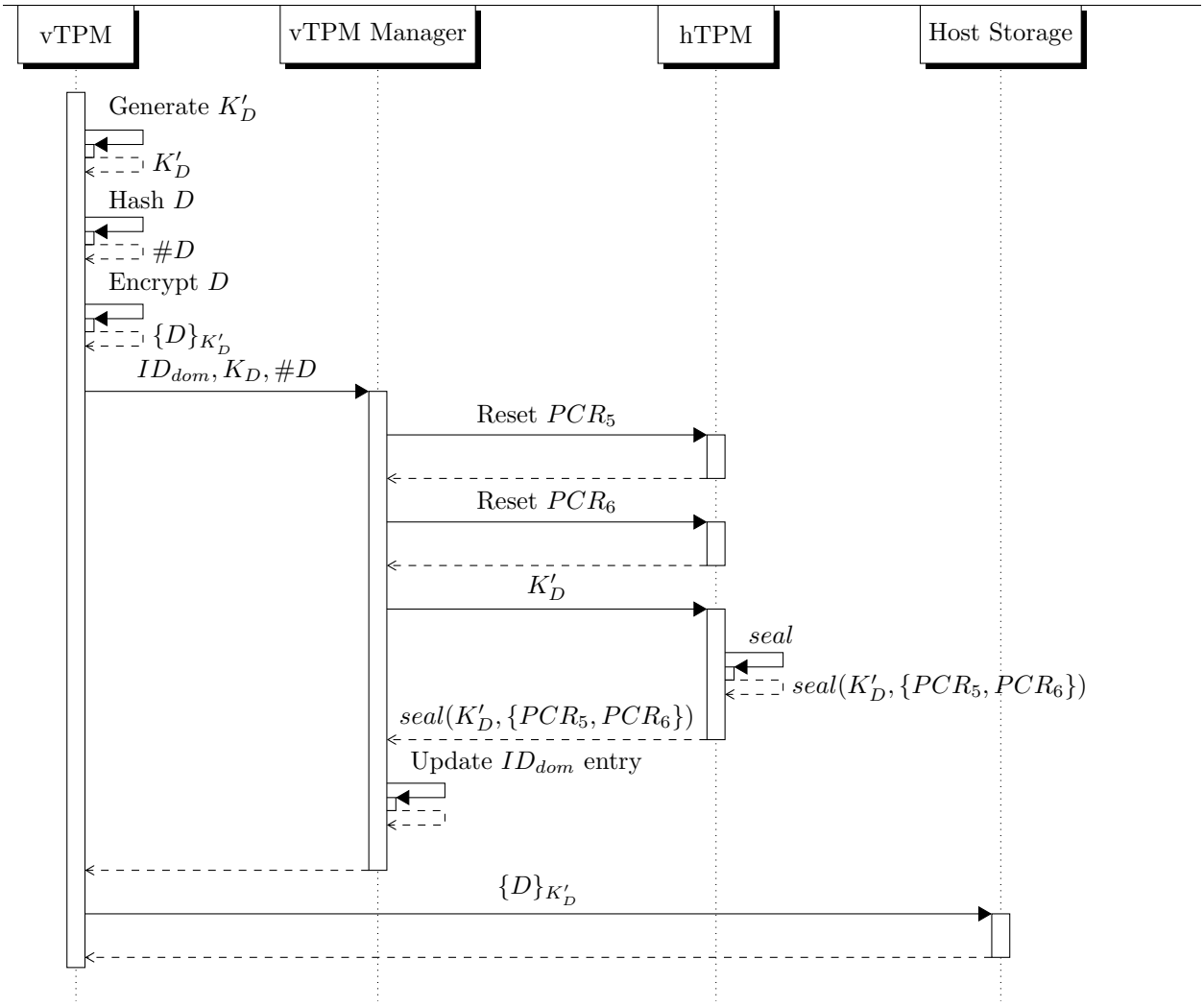

Figure 5: Command sequence to put a vTPM to sleep.

9. vTPM Manager clears Domain ID[2]

References

[2]I'm not certain of this, but it seems reasonable