vTPM Design Notes

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

May 30, 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.

Field	*Description*	*Initialized*	*Updated*
ID_{vtpm}	The persistent name of the vTPM	provisioning	never
ID_{dom}	Domain ID of vTPM VM	boot	boot
$\#D$	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

1 Introduction

2 vTPM Manager Data

3 Provisioning Sequence

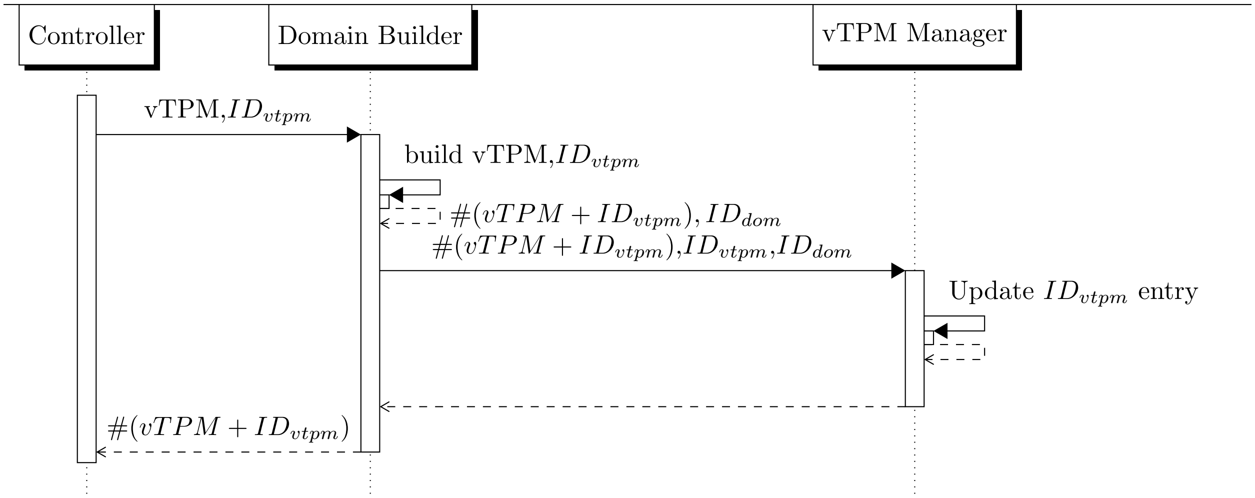
I'm not at all sure I have the provisioning sequence right. The vTPM can be used to generate an EK while the TPM_TakeOwnership command can be used to generate an SRK after startup. What else is initialized? Monotonic counter values and other stuff not specified here.

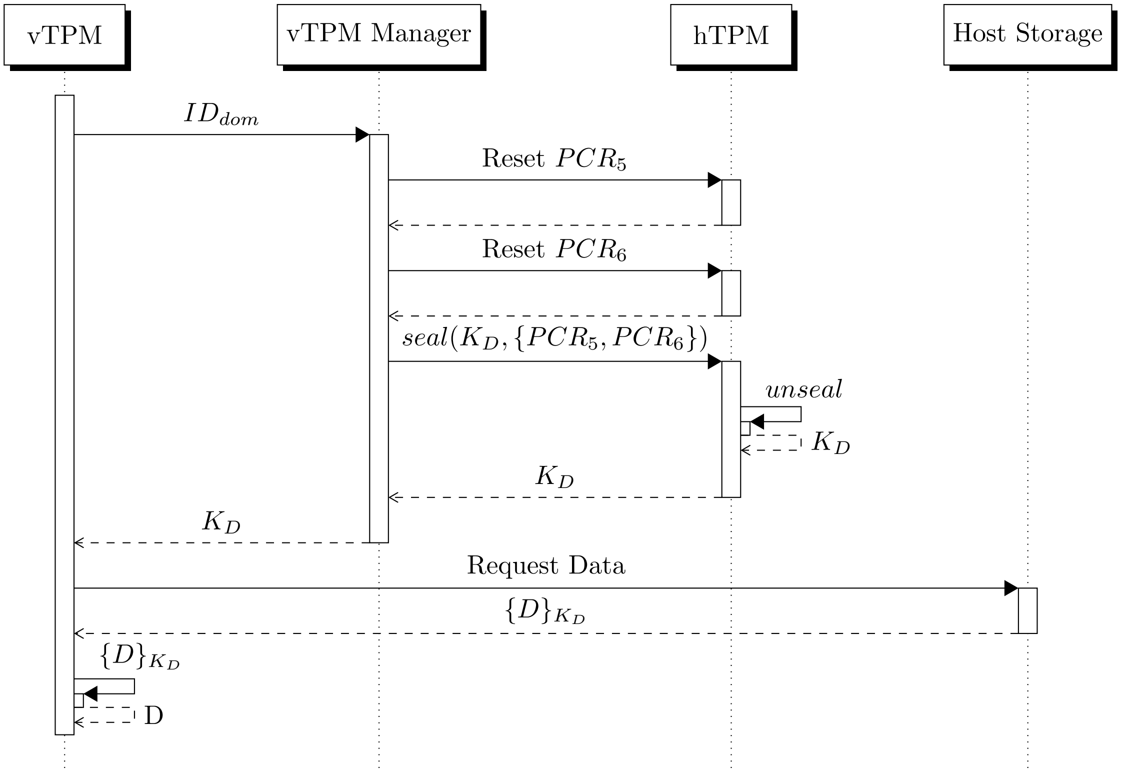
1. Platform booted through Controller – PCR_5 is now known by the vTPM Manager
2. A new Long Term Name is generated for the new vTPM
3. The new vTPM is built using the Domain Builder – PCR_6 is now known by the vTPM Manager
4. vTPM data is initialized – including fresh EK value.
5. vTPM is put to sleep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

4 Startup Sequence

4.1 vTPM Running

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

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.

   (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

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

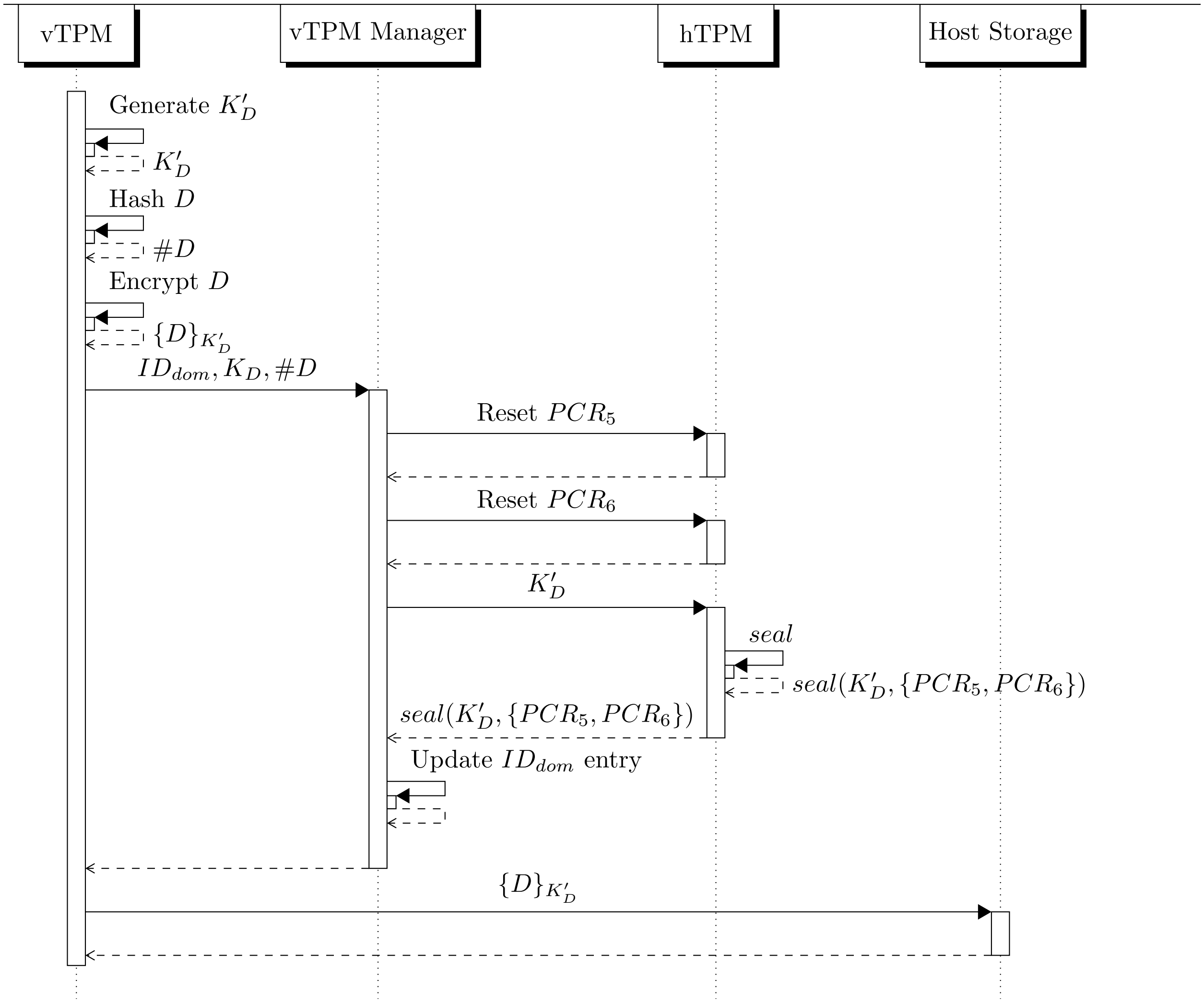
6. vTPM Manager resets and loads PCR_5 and PCR_6 from table entry

7. vTPM Manager seals K_D to TPM state

8. vTPM Manager updates its table with $seal(K_D, \{PCR_5, PCR_6\})$ and $\#K_D$

9. vTPM Manager clears Domain ID[2]

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


vTPM
vTPM Manager
hTPM
Host Storage
Generate K'_D
K'_D
Hash D
$\#D$
Encrypt D
$\{D\}_{K'_D}$
$ID_{dom}, K_D, \#D$
Reset PCR_5
Reset PCR_6
K'_D
$seal$
$seal(K'_D, \{PCR_5, PCR_6\})$
$seal(K'_D, \{PCR_5, PCR_6\})$
Update ID_{dom} entry
$\{D\}_{K'_D}$

References