REFERENCE

TCG TPM v2.0 Provisioning Guidance

Version 1.0
Revision 1.0
March 15, 2017

Contact: admin@trustedcomputinggroup.org

TCG **Published**

Disclaimers, Notices, and License Terms

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, DOCUMENT OR SAMPLE.

Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this document and to the implementation of this document, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this document or any information herein.

Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on document licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

Revision History

Revision	Date	Description
1.0	March 14, 2017	Initial release

Acknowledgements

Name	Company
Aigner, Ron	Microsoft
Anderson, Chris	US Government
Bell, Bob	Cisco Systems
Bennett, Daren	United States Government
Blum, Zachary	United States Government
Boucherak, Mehdi	STMicroelectronics
Boyle, Mike	United States Government
Cabre, Eduardo	Intel Corporation
Cathrow, Andrew	VeriSign, Inc.
Challener, David	Johns Hopkins University, Applied Physics Lab
Chin, Ga-Wai	Infineon Technologies North America Corp.
Collart, Olivier	STMicroelectronics
deCarle, Rob	WinMagic Inc
England, Paul	Microsoft
Ernst, Christoph	Swisscom
Fedorkow, Guy	Juniper Networks, Inc.
Fitzgerald-McKay, Jessica	United States Government
Fuchs, Andreas	Fraunhofer Institute for Secure Information Technology (SIT)
Goldman, Ken	IBM
Gratadour, Anne-Rose	Thales Communication
Hanna, Steve	Infineon Technologies North America Corp.
Hayoz, Thierry	Swisscom
Hoff, James	Lenovo (United States) INC
Hunt, Guerney	IBM
Kazmierczak, Greg	Wave Systems
Kelly, Scott	Hyperthought
Kuntze, Nicolai	Huawei Technologies Co., Ltd.
Laffey, Tom	Hewlett Packard Enterprise, HP Inc.
Latze, Carolin	Carolin Latze
Liu, Xin	Nationz Tech. Inc.
Lorenzin, Lisa	Pulse Secure, LLC

Name	Company
McDonald, Ira	High North Inc
McGill, Kathleen	Johns Hopkins University, Applied Physics Lab
Nelson, Amy	Dell, Inc.
Oliver, Doug	Lenovo (United States) INC
Potter, Stanley	United States Government
Pritikin, Max	Cisco Systems
Proudler, Graeme	Graeme Proudler
Ratliff, Emily	Advanced Micro Devices, Inc.
Rowe, Paul	MITRE
Rudy, Gregory	INTEGRITY Security Services, Inc.
Schiffman, Joshua	Advanced Micro Devices, Inc.
Schmidt, Charles	MITRE
Serrao, Gloria	United States Government
Stocco, Gabriel	Microsoft
Sulzen, Bill	Cisco Systems
Waller, Paul	CESG
Wiseman, Monty	Intel Corporation
Wooten, David	Microsoft
Wyant, Jeremy	General Dynamics C4 Systems
Zhang, Yi	Huawei

Table of Contents

1. Scope

This document outlines a process for provisioning and de-provisioning Trusted Platform Modules (TPMs) v2.0 for use in device identification, storage of encryption keys and credentials, and attestation of integrity measurements.

2. Terms and Definitions

Term	**Definition**
Advanced Encryption Standard	The specification for the encryption of electronic data published by NIST in FIPS PUB 197 and ISO/IEC 18033-3
AES	Advanced Encrypting System (aka RijnDael)
Attestation	The process of vouching for the accuracy of information. External entities can attest to shielded locations, protected capabilities, and Roots of Trust. A Platform can attest to its description of platform characteristics that affect the integrity (trustworthiness) of a Platform. Both forms of attestation require reliable evidence of the attesting entity.
Authentication	The process of verifying the claimed attributes, such as an identity, of an entity or User
Authorization	Granting access to a resource based on an authenticated identity
Authorization Value	A structure consisting of two bytes, which contains the size of the rest of the structure, and a set of bytes. Typically, the second set of bytes contains a password.
Basic Input Output System	Firmware that has first control of a system, and is responsible for basic configuration settings of a PC
BIOS	Basic Input Output System.
Centrally Managed	A class of Platforms that have HCIs that is managed/administered by a person or entity that centrally manages a large number of Platforms, ensuring they conform to common policies.
Certificate	A document that attests to the truth of contained statements. Typically a signed document that asserts that a key has some characteristics (such as the association of the private key with a User, device, or Platform.)
CIO	Chief Information Officer
Cryptographic Erase	Sanitization of data accomplished by sanitization of the encryption key used to encrypt the data. (See NIST SP 800-88 Guidelines for Media Sanitization [7])
CSO	Chief Security Officer
CTO	Chief Technology Officer
Device	A collection of resources that provide a service. This document calls end-User devices Platforms.
ECC	Elliptic Curve Cryptography
EK	Endorsement Key
EK Certificate	Endorsement Key Credential
EK Creator	The entity that creates the EK and its corresponding EK Certificate.
Empty Buffer	A byte array of length two set to 0x00 00.
Endorsement Key	An asymmetric key used to establish the provenance of a TPM.
Endorsement Key Credential	A credential containing the public key of an EK that asserts the TPM holding the private key of an EK conforms to the TCG's specifications for TPM.

Term	**Definition**
Enterprise Administrator	The entity responsible for administrating all aspects of one or more Platforms in an enterprise. This may include fulfilling the role of Platform Administrator for Platforms under its control.
EST	Enrolment over Secure Transport: a protocol for management of public key certificates
Extensible Firmware Interface	A modern replacement extension or replacement for BIOS
FIPS	Federal Information Processing Standard (created by NIST).
FIPS 140-2	A NIST standard for evaluation of cryptographic modules and software.
IETF	Internet Engineering Task Force
HCI	Human Computer Interface
Hierarchy	One of four sets of independent resources contained in a TPM. They are Storage (or Owner) Hierarchy, the Endorsement (or Privacy) Hierarchy, the Platform Hierarchy, and the Ephemeral Hierarchy.
Human Computer Interface	Interfaces on a Platform that typically include a keyboard, a display, a mouse, a camera, a microphone, and other peripherals which humans touch or with which they otherwise interact.
IDevID	Initial Device Identifier
IDevID Certificate	A credential holding the public key of an Initial Secure Device identifier signed by the device manufacturer.
IKE	Internet Key Exchange
Initial Device Identifier	(IEEE8021-DEVID-MIB). An asymmetric key provided by the manufacturer, typically used by a User in the creation of LDevIDs
Integrity Measurement	The process of obtaining metrics of platform characteristics that affect the integrity (trustworthiness) of a Platform and putting digests of those metric in shielded locations such as Platform Configuration Registers. More commonly, integrity measurement refers to the metric itself, or its digest.
Internet Protocol Security	A standard protocol for authenticating and encrypting each IP packet
IPsec	Internet Protocol Security
LDevID	Local Device Identifier
Locally Managed	A class of Platforms that have HCIs that is managed and/or administered by a person who is both the Owner and User of the Platform.
Local Device Identifier	An asymmetric key created by a User, and used to locally identify the device.
Lockout Administrator	A privileged TPM role used to manage the Lockout Mode Configuration Parameters, which in turn mitigates dictionary attacks against Authorization Values. It is the entity that either has knowledge of the Lockout Authorization Value (*lockoutAuth*) or can satisfy the Lockout Authorization Policy (*lockoutPolicy*).
NULL	A default password used in the TPM: a two-byte array of zeros.
NV Index	NV memory into which the OS or Platform may define a special structure in order to store persistent data values.
NV Memory	Non-volatile memory, which retains its values when power is removed (See NV Index and Persistent Memory)

Term	**Definition**
Owner	The entity that has administrative rights over certain aspects of a TPM installed in a Platform, such as the Storage Hierarchy. To disambiguate the Owner from the Platform Owner, this document sometimes called this entity the TPM Owner.
PC	Personal Computer
PCR	Platform Configuration Register:
Persistent Memory	NV Memory into which the OS or Platform may store keys on which it can perform operations.
PKINIT	Public Key Cryptography for Initial Authentication
Platform	A collection of resources that provide a service. In the context of this document, a Platform is typically a User device such as a desktop, laptop, smartphone, or tablet, and to a certain extent, network equipment such as routers and switches.
Platform Administrator	The entity responsible for administrating all aspects of a Platform. This typically includes the initial provisioning process once the Platform has been delivered to the owner. This entity is also the TPM owner.
Platform Configuration Register	Dynamic memory in a TPM typically used to store integrity measurements related to the software and configuration of a Platform on which the TPM resides
Platform Owner	The entity that owns the Platform with a TPM Installed. The Platform Owner is not necessarily the User of the Platform (e.g. in a corporation, the Platform Owner might be the IT Organization while the User is an employee). The Platform Owner also does not necessarily have administrative rights assigned to the Owner of the TPM.
Platform Manufacturer	Original Equipment Manufacturer of Platforms that either installs or instantiates TPMs in its equipment before provisioning them and delivering finished products to its customers.
PPI	Private Personal Information
Privacy Administrator	A privileged TPM role used to manage the Endorsement Hierarchy and the privacy policies associated with it. It is the entity that either has knowledge of the Endorsement Authorization Value (*endorsementAuth*) or can satisfy the Endorsement Authorization Policy (*endorsementPolicy*).
Protected Capability	The set of commands with exclusive permission to access shielded locations.
Protected Location	A location external to the TPM that stores TPM objects with encrypted sensitive fields that is protected from disclosure, but not protected from deletion.
Provision	Configure / Customize
Return Merchandise Authorization	Authorization to return equipment for repair or replacement by Platform Manufacturer.
Rivest-Shamir-Adleman Cryptosystem	A cryptosystem for public-key encryption whose name comes from the first initials of its creators
Root of Trust	A component that must always behave in the expected manner, because its misbehaviour cannot be detected.
Root of Trust for Reporting	A computing engine capable of reliably reporting information held by the Root of Trust for Storage.

Term	Definition
Root of Trust for Storage	A computing engine capable of maintaining an accurate summary of values in integrity digests and sequences of digests.
RSA	Rivest-Shamir-Adleman Cryptosystem
RSA Template	A template used by a TPM in the creation of RSA keys.
SHA	Secure Hash Algorithm. Comes in many varieties, including SHA-1, SHA256, SHA384, and SHA512
Shielded Locations	A place (memory, register, etc.) where it is safe to operate on sensitive data; data locations that can be accessed only be protected capabilities.
SRK	Storage Root Key
SRK Template	A template used by a TPM in the creation of SRKs.
Storage Root Key	The root key of a hierarchy that provides confidentiality for private keys and other secret values protected by the TPM.
Template	A structure used by a TPM in the creation of cryptographic keys. It contains information about the algorithm, key size, policy, types of authorization that may be used with the key, etc.
TLS	Transport Layer Security
Touch-Free Devices	Devices that may have access to relatively generous resources in terms of power, computing engines, and electronic storage. Provisioning these devices may require a means of identifying the device as owned by the provisioner. Note: The enterprise typically only has a list of device serial numbers to use for identifying those systems. Serial numbers are not sufficient for remote identification. The enrolment protocol may use IDevIDs as a means of securely identifying remote.
TPM	Trusted Platform Module
TPM Owner	The entity possessing Owner Authorization of a TPM (see Owner).
Transport Layer Security	A protocol designed to provide communication security over the internet
Trusted Platform Module	An implementation of functions defined in the TCG TPM specifications.
UEFI	Unified Extensible Firmware Interface: A modern replacement extension or replacement for BIOS
User	An entity that makes use of the Platform and the capabilities of the TPM installed within it. The User of the Platform is not necessarily the Platform Owner (e.g. in a corporation, the Platform Owner might be the IT department while the User is an employee). The Platform may have multiple Users.
Virtual Private Network:	An means of authenticating and encrypting communications shared over a network
VPN	Virtual Private Network
Touch-Free Provisioning	Provisioning of a device remotely, over a network. This requires a means by which a provisioner can identify a device. For example: An enterprise may only have a list of device serial numbers to use for identifying those systems. Serial numbers are not sufficient for remote identification. A Touch-Free Provisioning enrolment protocol may use IDevIDs as a means for provisioners to securely identify remote devices with known serial numbers.

3. Introduction

This document is not a TCG specification, but provides recommendations which customers may consider when making purchasing decisions. The target TPM for this document is a TPM version 2.0 revision 01.16 [24][25][26][27]. The target audience for this document includes TPM Manufacturers, Platform Manufacturers that incorporate TPMs into their products and Platform Administrators who want to take advantage of the TPMs in their Platforms. The requirements for provisioning TPMs in this document applies to broad categories of platforms, as evidenced in the use cases and overview sections that follow. Platform work groups may consider consulting this document when specifying required TPM features for their Platforms. If manufacturers and Platform Administrators follow the guidance provided by this document, or platform-specific guidance that is consistent with this guidance, then Application Developers and Users can count on finding a consistently provisioned TPM. This allows them to predict the security capabilities and behavior of the TPM and associated Platforms.

This document provides guidance for TPM Manufacturers, Platform Manufacturers and Platform Administrators that enables provisioning the TPM for common expected use cases. Those recommendations are found in Sections 9, 10, and 11.

4. Problem Statement

Enterprise Administrators of Platforms with TPM version 1.2 experienced difficulty in provisioning applications that use the TPM due to variations in vendor-specific provisioning implementations. Consistent with the flexibility of the TPM 1.2 specification, manufacturers deployed different methods for turning on, enabling, and activating the TPMs; some variations even existed between product lines from the same manufacturer. The TCG mitigated this issue with "PPI Specification 1.2" [22], which addressed these issues by defining BIOS-level variables that firmware and operating system developers and vendors may use to better predict the state of the TPM and control it. However, not all vendors built implementations to the most recent specification. Compounding the problem, in the first few years Platform Manufacturers sold TPM-enabled Platforms, several TPM vendors chose to provide no Endorsement Keys, and few vendors created Endorsement Key Certificates (EK Certs), both of which are necessary for TPM-based device identification and attestation. This lack of Endorsement Keys (EKs) and Certificates from the TPM Manufacturers compromised the ability of remote policy compliance and monitoring services to establish the authenticity of the TPM and prevented Users from utilizing the TPM as the root of trust for reporting. These problems, among others, negatively affected TPM adoption rate.

Although, in the last three years, many Platform Manufacturers have shipped TPM v1.2-enabled Platforms with EKs and EK Certs, the deployment of TPM v2.0 presents a new set of challenges, especially with the addition of new hierarchies, the complex role of authorization of hierarchies, and the separate authorization of objects within each hierarchy. This document provides recommendations for TPM manufacturers, Platform Manufacturers, and Platform Administrators (whether it is an enterprise administrator or an application acting as proxy for such an administrator). Operating system and application developers should have some confidence in the initial state of the TPM and focus on taking advantage of its features to provide a security capability without having to reconfigure each TPM before using it.

5. Use Cases

The following use cases provide examples for scenarios in which the TPM provides security to End Users and Platform Administrators. They also establish the assumptions and drive recommendations for the TPM state before and during all stages of provisioning. This document groups the use cases into classes of use cases. This convention helps the reader more easily identify which recommendations accommodate each class of use cases.

5.1 Identity

The TPM can provide a standardized, unspoofable, hardware-based identity – in contrast with IP and MAC addresses that are relatively trivial to spoof. Enterprises find this property useful if they need to identify Platforms on their network for many reasons, including, but not limited to, the following:

- **Asset Management -** Verifiable hardware identity can help Network Administrators seeking to prevent network attacks, monitor network health, and maintain compliance by allowing them to whitelist hardware to track the use of that hardware across their network.
- **Platform Supply Chain Integrity** - Platform Manufacturers may provide assurances of the authenticity and security of products they purchase. Cryptographically assertable standards-based identities, such as those provided by the TPM, can facilitate providing those guarantees to device Users and Platform Administrators.
- **Supply Chain Risk Management** – Platform Manufacturers can uniquely identify genuine components of their products. Users and Platform Administrators can examine evidence that their Platforms contain legitimate parts.
- **Compliance Reporting -** For Enterprises to reliably match compliance, health, and posture reports with Platforms, they require a durable unique identifier for each Platform. Such an identifier allows Network Administrators to locate, quarantine, or remediate Platforms that have fallen out of compliance with network policy.
- **Threat Detection -** Unique identification of Platforms allows Enterprises to aggregate Platform behavior patterns over time, which enables analysis for anomaly detection.
- **Platform Remediation –** If Platform Administrators need to update, reconfigure, or block a Platform, they first need to identify and locate that Platform on the Network.
- **Endpoint-to-Endpoint Authentication** - TPM-based Platform identifiers provide strong credentials to be used in standard authentication protocols such as Transport Layer Security (TLS), Internet Key Exchange (IKE) within IPsec, and PKINIT (Public Key Cryptography for Initial Authentication for Kerberos), and in Simple Authentication and Security Layer (SASL) mechanisms.

One could argue that device identity is not an end unto itself, but rather facilitates other more interesting use cases, some of which are rather complex. Nonetheless, many enterprises strongly desire the capability to electronically identify their own assets without physically visiting them. The examples above serve to provide a sampling of the possibilities, not all of which a single document such as this, one can enumerate. For this reason, we consider Identity as a class of use cases that motivates the recommendations in this document.

5.2 Storage

The TPM provides shielded locations for storing secret information as well as for protecting the integrity of information. Additionally, interfaces accompany these shielded locations that provide Users of the TPM the ability to control the access to and use of the contents of the shielded locations. These properties allow a class of use cases which this document collectively calls Storage and outlines a few of the use cases below.

- Disk Encryption Key Storage - The TPM can provide secure key storage to prevent unauthorized access to data on lost, misplaced or stolen Platforms. In addition, the TPM can protect data on these Platforms with hardware enforced dictionary attack mitigation. The TPM can enable cryptographic erase to deny unauthorized access to protected data.
- Public Key Certificate Protection - A number of applications rely on public key certificates for authentication and/or integrity verification. Trust anchor certificates (e.g. PKI root certificates) are one type of public key certificates that requires some form of protection against unauthorized modifications and replacements. Identity certificates are another type of public key certificate that requires protection from unauthorized deletion/replacement. The TPM provides ideal protection for keys attested by public key certificates.
- VPN Credential Storage - A VPN can provide remote access to sensitive internal network resources. Accordingly, enterprises might tightly control such access. They can use the TPM to store securely high-valued credentials used for VPN authentication.
  The TPM provides two mechanisms to bind credentials to a particular machine, preventing credential theft attacks. In the first, the mechanism uses the TPM to create and store the key internally. Subsequently, an enterprise can use key attestation to obtain a certificate from the server and use it for authentication. In the second, the mechanism imports a key into the TPM under the control of a trusted duplication authority and subsequently shields credentials from theft. In either case, enterprises can easily store the TPM protected authentication material on mass storage, but it is useless without access to the actual TPM device itself. (See ″TPM 2.0 Architecture Specification″ [24] Section 9.5.4 “Protected Location” and Section 10.3 “Protection of Shielded Locations” for more information).

5.3 Attestation of Firmware Integrity Measurements

When a Platform with a TPM boots, executable components may perform integrity measurements of other components and extend these in the TPM's Platform Configuration Registers (PCRs) before passing execution control to the newly measured components. Changes to the values in the PCRs would then indicate changes to the measured components. Local and remote Platform integrity monitoring applications can use PCRs to detect unauthorized firmware updates or modifications, or to determine whether an authorized update actually happened.

Users may seal keys within the TPM such that it releases them only if the Platform is in a pre-determined state – such as when the PCR values match the values present when the key was sealed. For example, in some environments it might be desirable to seal a hard-drive decryption key to the PCR values. If the PCR values change—indicating firmware or hardware modification—then the TPM does not release the key and the drive cannot be decrypted.

In a first embodiment of attestation, a remote evaluator wants to determine the integrity of a remote Platform as represented by the measurements stored in the PCRs of that Platform's TPM. In this scenario, the evaluator and Platform will use remote attestation to gate access to network resources. Providing remote attestation requires two things: proof of the state of the Platform as represented by the current PCR values, and proof that the Platform providing that proof is the one attesting to the evaluator. The TPM provides the TPM2_Quote() command to satisfy the first requirement by using a restricted signing key to sign the current PCR values. The TPM satisfies the second by allowing the creation of a restricted signing key fixed to the TPM, and uses a certificate trusted by the evaluator that attests that the key is locked to a particular Platform.

In a second embodiment of attestation, a local evaluator wants to determine the state of a local Platform as represented by the integrity measurements stored in the PCRs of the Platform's TPM. During a provisioning phase, the evaluator uses the local TPM to seal a value to a state of the local Platform represented by the integrity measurements in the PCRs. When the evaluator wants to expose some protected local resource (e.g. an encrypted drive, a network interface, a camera and microphone, etc.) to the Platform based on the state of the Platform, it will task the TPM to unseal the value. If the TPM presents the expected value this implies that the Platform is in the expected state based on integrity measurements in the PCRs. At that point, the evaluator grants access to the resource(s) on the local Platform. If the TPM fails to present the appropriate value because of mismatching PCRs, then the evaluator assumes the Platform is not in an acceptable state and denies access to the resource.

6. Overview of the Provisioning Process

A wide variety of Platforms contain TPMs, such as desktop computers, laptops, servers, tablets, and smartphones along with non-endpoint Platforms, such as printers, switches, routers, and wireless access points. Platforms with significantly constrained resources, such as feature phones, wearable devices, and automobile components, will likely contain embedded system TPMs that have a smaller set of functional capabilities that more closely match the expected use cases in constrained environments.

This document applies to TPMs implemented according to a Platform TPM profile, such as the PC Client, and Mobile TPM profiles (see “PC TPM 2.0 Profile” [21] and “TPM 2.0 Mobile Refererence Architecture” [16], respectively). It also may apply to other Platforms that contain TPMs, such as servers, printers, switches, routers, and wireless access points, even though a TPM profile for those devices does not yet exist.

Section 7 describes the foundational elements commonly provisioned for one or more Use Cases described in Section 5.

Section 8 provides an overview of functional requirements needed for provisioning the TPM for common Use Cases.

In the first provisioning stage in Section 9, the TPM manufacturer creates the TPM, and prepares it for installation into a Platform, including cryptographic evidence of provenance of the TPM.

Generally, the Platform Manufacturer will then install the TPM on a Platform and perform additional provisioning as described in Section 10.

After arriving at the customer, the Platform Administrator will finalize provisioning critical security features of the TPM for the use cases supported in their operating environment as described in section 11.

Finally, at end-of-life or when the Platform Owner transfers the Platform to a new User, the de-provisioning process securely eliminates critical security information stored in the TPM and prepares the TPM for transition to another User.

7. Foundational Elements of the TPM Provisioning Process

This section describes the basic objects involved in the provisioning process, such as Authorizations, Endorsement Hierarchies, Endorsement Keys (EKs), EK Templates, EK Certificates, Platform Hierarchies, IDevIDs, Storage Hierarchies, Storage Root Keys (SRKs), SRK Templates, etc. This section also provides recommendations on how to provision these objects to instantiate an operational TPM to support the use cases in section 5.

7.1 Authorizations

The TPM provides a rich set of access controls for hierarchies and objects. The sections that follow briefly describe the hierarchies and foundational elements for which the TPM provides access control, and the mechanisms it provides Users to grant and restrict access in these cases. Section 13 "TPM Control Domains" of "TPM 2.0 Architecture Specification" [24] provides an in-depth treatment of access control for the hierarchies, while Section 19 "Authorizations and Acknowledgements" of the same reference provides an exhaustive look at access control for individual objects.

7.1.1 Owner Authorizations

The TPM provides the Owner Authorization Value (*ownerAuth*) and Owner Authorization Policy (*ownerPolicy*) as access controls for a Platform Administrator of a Platform with the TPM to manage control of the availability of the Storage Hierarchies, creating primary keys in the Storage Hierarchy, NV Indexes, storage and eviction of persistence of keys, and changing of Owner Authorizations. They can also use *ownerPolicy* to allow additional means of authorizing owner privileges, such as the use of a smart card or biometric reader. "TPM 2.0 Architecture Specification" [24] indicates the default values of *ownerAuth* and *ownerPolicy* are the Empty Buffer.

7.1.2 Platform Authorizations

The TPM provides the Platform Manufacturer the opportunity to control features it offers to the Owner of the TPM. In particular, it provides Platform Hierarchy Authorization Value (*platformAuth*) and Platform Hierarchy Authorization Policy (*platformPolicy*) for the Platform Manufacturer to control the allocation of NV Indexes, PCR configuration, and availability of the Storage, Platform and Endorsement Hierarchies, changing of the primary seeds, resetting the *platformAuth* and *platformPolicy,* as well as other functions.

Upon every TPM2_Startup() command, "TPM 2.0 Architecture Specification" [24] and "TPM 2.0 Commands Specification" [26] require the TPM to enable the Platform Hierarchy by setting *phEnable* to 1. The TPM sets the *platformAuth* and *platformPolicy* to the Empty Buffer by default after each TPM Reset and TPM Restart per "TPM 2.0 Commands Specification" [26], Section 9.3.1 "General Description". Following a TPM Resume, the TPM sets *platformAuth* and *platformPolicy* to their previous states as saved by the *Shutdown(STATE)* command.

7.1.3 Privacy Authorizations

Platform Users may use the role of Privacy Administrator to establish and potentially maintain access control over the Endorsement Hierarchy objects. See the "EK Credential Profile 2.0" [19] for more information on the recommended use of the Privacy Administrator Authorizations and authorizations for Endorsement Hierarchy Primary Keys. According to section 13.5 "Privacy Administrator Control" of "TPM 2.0 Architecture Specification" [24], TPM2_Clear() and TPM2_ChangeEPS() set the initial values of *endorsementAuth* and *endorsementPolicy* to the Empty Buffer.

Privacy Administrators who set the *endorsementAuth* to 256 bits chosen randomly and the *endorsementPolicy* to either a *policyDigest* or the Empty Buffer will help prevent unauthorized control of the Endorsement Hierarchy and unauthorized creation of endorsement primary keys. If the Privacy Administrator (or his or her proxy, e.g. the Platform Administrator) sets *endorsementAuth* to 256 bits chosen randomly, then he or she can manage the creation of primary keys in the Endorsement Hierarchy. If the Privacy Administrator sets *endorsementAuth* to 256 bits chosen randomly without saving them or instituting some other mechanism for recalling them, then the Privacy Administrator effectively disables the Endorsement Hierarchy unless he or she has created an *endorsementPolicy* as an alternative.

7.1.4 Lockout Authorizations

A Lockout Administrator (which can be the Platform Owner or Platform Administrator) can implement Lockout Authorization either through a Lockout Authorization Value (*lockoutAuth*) or a Lockout Authorization Policy (*lockoutPolicy*). This authorization has two main purposes. First, Lockout Administrators can use it to manage the Dictionary Attack Parameters and Lockout. See section 7.7 "Dictionary Attack Parameter Defaults" for more information about Dictionary Attack Parameters. A Lockout occurs when a User attempts to authorize the TPM to use objects with incorrect Authorization Values in excess of the number of times allowed by the Dictionary Attack Parameters.

When the number of failed authorization tries reaches the limit, the Lockout Administrator can enter the *lockoutAuth* to reset the Dictionary Attack Counter. However, if he/she mistypes *lockoutAuth* then he/she must present to the TPM either a correct *lockoutPolicy* or wait for a TPM power cycle before it will allow a correct entry of the *lockoutAuth*. The time limit imposed on the *lockoutAuth* does not affect *lockoutPolicy.* Therefore, one could set the *lockoutAuth* to prevent unauthorized people from using it, and set the *lockoutPolicy* (for example, to reference a public key via TPM2_PolicySigned() to allow use of a private key for authorization) so that a mistyped authorization does not require a cold boot.

Second, Lockout Administrators may use Lockout Authorization to change the Storage Primary Seed, which resets the Storage Hierarchy, wiping out all persistent, non-persistent, and permanent keys in that hierarchy. This also erases all NV Indexes associated with the hierarchy and removes their associated index information (see section 14.3.4, "Storage Primary Seed" of "TPM 2.0 Architecture

Specification" [24]). A change of the Storage Primary Seed also flushes all resident Storage and Endorsement Hierarchy keys, prevents the reloading of non-resident keys under the Endorsement Hierarchy, and erases all NV Indexes associated with the hierarchy, removing their associated index information (see section 14.3.4, "Storage Primary Seed" of "TPM 2.0 Architecture Specification" [24]). Platform Hierarchy keys will remain.

7.1.5 Authorization Policies

Authorization Policies of keys and other objects are immutable[1]. During the creation of the names of the objects, the TPM hashes over the authorization policy among other fields within its structure. Object owners are unable to change directly the authorization policies of their objects since doing so would damage the integrity of the object as reflected by the name, which renders these objects unusable. See Section 16 "Names" in "TPM 2.0 Architecture Specification" [24].

To accommodate those who want to select from among various Authorization Policies for their keys and objects, the TPM provides them the capability to pre-calculate Authorization Policies outside the TPM. This provides Platform and Enterprise Administrators with a method to create a library of acceptable policies created from which object owners can choose to remain compliant to organizational security policies. For more information, see "TPM 2.0 Architecture Specification" [24], Section 19.7 "Enhanced Authorization".

7.2 Primary Seeds

As stated in the ″TPM 2.0 Architecture Specification″ [24], Section 14.3.1, "Introduction", the size in bits of all primary seeds must be at least twice the size of the bit strength of the strongest algorithm supported. Seeds should be from an approved random source, such as the TPM's RNG. Although it is conceivable the TPM manufacturer may inject these bits into the TPM for the Endorsement Hierarchy, the security responsibility for this belongs to the TPM Manufacturer.

7.3 Platform Hierarchy

The TPM provides the ability to enable and disable the Platform Hierarchy through phEnable. Since the TPM initializes with phEnable set by default, Platform Manufacturers and Platform Administrators need take no further action to enable the Platform Hierarchy.

Platform manufacturers may use the Platform Hierarchy to protect the update mechanisms of the roots of trust of the Platform in order to comply with the NIST SP 800-147 series (see "NIST SP 800-147 BIOS Protection" [5] and "NIST SP 800-147B BIOS Protection for Servers" [11]). They may also anchor platform identity in the Platform Hierarchy, for example, for warrantee, but such identities

[1] However, TPM2_PolicyAuthorize() can be used to change the means of satisfying some policies.

should not be generally Platform Owner usable. To comply with internationally accepted norms for security strength, such as ISO, the Platform Manufacturer may provision a Platform Hierarchy with combinations of algorithms and key sizes commensurate with published standards. The Platform Manufacturer should choose an algorithm set that meets or exceeds the minimum security requirements for the data it stores and for the environment in which it anticipates the users will deploy it. This may help protect the integrity of the Platform Roots of Trust, and thus the integrity of the rest of the Platform with transitive trust chains that extend down to these roots of trust.

Platform specific working groups may specify the desired behaviors of platform services that may depend on the clearing and setting of *phEnable.* Platform manufacturers will determine how to control the Platform Hierarchy whilst providing properly working platform services. By default, the TPM enables the Platform Hierarchy. See section 7.1.2 “Platform Authorizations” of this document for more information about recommendations for enabled Platform Hierarchies. If the platform firmware disables the Platform Hierarchy by clearing *phEnable* to 0, then subsequent platform firmware, operating systems, and applications cannot use *platformAuth* or *platformPolicy* to authorize any TPM actions.

7.4 Endorsement Hierarchy

The TPM provides the ability to enable and disable the Endorsement Hierarchy through *ehEnable*. Since the TPM initializes with *ehEnable* set by default, manufacturers and administrators need take no further action to enable the Endorsement Hierarchy.

The TPM supports an Endorsement Hierarchy in order for the User/Platform Administrator to store and control objects deemed privacy sensitive. In particular, the Endorsement Hierarchy houses objects used in the certification of the authenticity of the TPM as well as the certification of the source of reports from the TPM. The Endorsement Hierarchy includes an Endorsement Primary Seed (EPS). Section 7.2 “Primary Seeds” discusses primary seeds. Section 7.1.3 “Privacy Authorizations” discusses the role of the Privacy Administrator who controls the creation of objects in Endorsement Hierarchy. The following subsections contain information for endorsement hierarchy foundational elements.

If Platform Manufacturers take no action then the TPM enables the Endorsement Hierarchy by default. Disabling the Endorsement Hierarchy restricts the ability of the User to verify the authenticity of the manufacturer of the TPM. Changing the EPS permanently destroys the ability of the end user to verify the authenticity of the manufacturer of the TPM.

7.4.1 Endorsement Primary Keys

The EK Creator is one of the TPM Manufacturer, the Platform Manufacturer, or the Platform Owner (User), depending on the type of TPM and the relationships that exist between the TPM Manufacturer and the Platform Manufacturer, and between the Platform Manufacturer and the Platform Owner. This document strongly encourages the EK Creator to follow the guidance in “EK Credential

Profile 2.0" [19], which recommends a default template for creating Endorsement Primary Keys, (also called Endorsement Keys or EKs for short).

TPM manufacturers could play the role of EK Creator to create both the EK and the EK Certificate. In cases where this is not possible (e.g. when the firmware or software is not available for use by the hardware until platform-manufacture time), the Platform Manufacturer may play the role of EK Creator and create both the EK and EK Certificate. In enterprises in which the Platform Owner plays the role of EK creator for both the EK and EK Certificate, only members of its ecosystem may value those certificates. In these cases, the certificates may have little or no value outside the Owner's immediate enterprise. This document assumes that the EK Creator will be either the TPM Manufacturer or the Platform Manufacturer.

The "Algorithm Registry" [18] offers a wide variety of cryptographic algorithms. However, "EK Credential Profile 2.0" [19] narrows the choices for EKs. Furthermore, platform profiles may mandate other algorithms in addition to the choices "EK Credential Profile 2.0" [19] makes.

7.4.1.1 Endorsement Primary Key Templates

"EK Credential Profile 2.0" [19]contains templates for each of the RSA Endorsement Keys (see section 2.1.5.1 "RSA Template") and the ECC Endorsement Keys (see section 2.1.5.2 "ECC Template"). Updates to that document will supersede the information in the informative paragraph below.

The EK Template features a field into which the EK creator may place information that causes the EK generation scheme internal to the TPM to generate statistically unique values. The TPM v2.0 specification calls this field *unique* and overloads it so that it contains one value when the application provides this structure as input and another value when the applications receives this structure as output. "EK Credential Profile 2.0" [19]uses a new term, EK Nonce, to denote that the EK Creator should use this field to input a random value to generate statistically unique primary keys. "EK Credential Profile 2.0" [19]recommends the Empty Buffer for the EK Nonce.

7.4.1.2 Endorsement Key Primary Objects and Handles

This document assumes that every TPM contains at least one EK when the Platform Owner receives it. TPM v2.0s can implement EKs with a variety of choices for algorithms.

"EK Credential Profile 2.0" [19] recommends EK Creators use the handles for preinstalled EKs that lie in a predetermined range of locations assigned by the Technical Committee in their "Registry of Reserved TPM 2.0 Handles and Localities" [17]. When an EK Creator supplies an EK Template to TPM2_CreatePrimary() to generate an EK, the TPM returns a transient object key handle. The Platform Manufacturer should provide software that enables the Platform Owner to store the EK persistently. Note that allocation of a handle does not automatically also allocate NV Index or Persistent Memory space.

7.4.1.3 EK Credentials and EK Credential Handles

"EK Credential Profile 2.0" [19] provides guidance on how to create Endorsement Key Credentials. "Registry of Reserved TPM 2.0 Handles and Localities" [17] reserves a range of NV Index handles to identify pre-provisioned EK certificates.

7.4.2 IDevID Keys

In addition to the EK Keys and Credentials, to which the preceding sections referred, for touch-free devices, the Platform Manufacturer may also install or cause to be created the IDevID Key pair and the associated IDevID Certificate. Note that in this section and subordinate sections (i.e. the IDevID sections) when it says "the Platform Manufacturer puts something in the TPM", it really means that either the Platform Manufacturer will actually provision the TPM directly before delivering the Platform to the User, or it will provide firmware or software that indirectly provisions the TPM when the User first boots the device or chooses to execute Platform Manufacturer-provided software that purposely provisions the TPM with IDevID Keys and Certificates. User access to IDevID Keys and Certificates is important for the Use Cases that depend on LDevIDs for device identity. However, the Platform Manufacturer may want to take the necessary precautions to prevent the inadvertent permanent removal of the IDevID from the device by the User. These mitigations are outside the scope of this document. Within scope is that the User access to the IDevID Keys and Certificates is consistent across Platforms, especially across a class of Platforms such as touch-free Platforms.

The class of touch-free Platforms requires provisioning in a touch-free environment. A device may not have embedded Human Computer Interfaces (HCI) or enterprises may not wish to send each Platform to a central location for provisioning before installation. In these cases, devices should have the ability to identify themselves remotely in such a way that allows the enterprise to know that the device is indeed the one purchased. The IDevID is an industry standard way to provide this assurance.

The "IEEE 801.1AR Secure Device Identity" [3] IDevID concept associates an asymmetric key pair and associated X.509 Certificate with the Platform. The Platform uses this certificate, issued by a Certificate Authority controlled by the Platform Manufacturer, to attest to the identity and authenticity of the Platform. In addition, applications may use it for other functions such as to act as authentication for enrollment protocols. The Platform Manufacturer may use the IDevID Certificate as a proof of authenticity for Return Merchandise Authorization (RMA) or licensing actions[2]. The Enterprise may also use it for asset management and similar functions. The Enterprise network management system may use the IDevID Certificate as the means to verify the eligibility of a Platform to join the Enterprise network.

[2] If the manufacturer intends enterprises to use IDevIDs, it may want to provide some protections against an end user inadvertently executing the TPM2_ChangeEPS() command and thus preventing usage of the IDevID.

7.4.2.1 IDevID Key Templates

The Platform Manufacturer may intend the Platform to create IDevID Keys after delivering the Platform to the Platform Owner (i.e. the Platform Manufacturer does not store the IDevID key in persistent memory). Storing the IDevID Template in a region of NV memory reserved for that purpose would greatly facilitate the creation of IDevIDs. IDevID Templates look almost identical to Endorsement Key Templates, except that they are restricted signing keys instead of restricted decryption keys. When applications load IDevID keys into the TPM without Platform Owner or User intervention, the template must have the *userWithAuth* bit CLEAR, as the Platform Manufacturer would have selected an Authorization Value when creating the key from the template, not under the control of the User. For applications that load IDevID keys into the TPM with User intervention, the template may have the *userWithAuth* bit SET, as the User will select an Authorization Value upon creation of the IDevID key from the template.

7.4.2.2 IDevID Key Object and Handle

Platform Manufacturers create an IDevID Key in the TPM using a Template or create an IDevID Key offline and import it into the TPM. Each method has advantages and disadvantages. For example, using Templates may cause the TPM to take an indeterminate amount of time to create the key, which causes uncertainty in the timing of the manufacturing process, but results in a very tight binding between the IDevID Key and the TPM. Importing IDevID Keys into TPMs takes the timing uncertainty out of the manufacturing process, but complicates the process of binding the key to the TPM and thus to the device. The destination hierarchy (e.g. Endorsement vs. Platform) depends in the projected use case for the IDevID. This guidance document allows the Platform Manufacturer to choose the method and hierarchy best suited to its products and markets.

In either case, the Platform Manufacturer may store the key in persistent memory or should allow the User the option of storing the key in persistent memory. The TCG in “Registry of Reserved TPM 2.0 Handles and Localities” [17] has set aside a range of persistent memory for Platform Manufacturers (or Users) to store IDevID keys. It is important for Platform and User applications to access the IDevID Key post manufacturing. It is also critical that the key have a known handle so that applications can reference it. The TCG set aside reserved handles for persistent keys; see Section 7.8 “NV Memory” for the recommended location for IDevID Keys. The Platform Manufacturer may install the IDevID Key using the default Privacy Administrator’s Authorization during its manufacturing process, during the initial boot of the Platform before the Platform Administrator sets the Privacy Administrator’s Authorization to something other than the default values, or later by the User with Platform Manufacturer-supplied software using Platform Administrator-set values for the Privacy Administrator. This document allows the Platform Manufacturer to choose from among these or some other method that suits its products and markets.

7.4.2.3 IDevID Credential and Credential Handle

The IDevID credential provides the identity for the Platform. This is crucial for Touch-Free Platforms and Enterprise environments that want to use standard protocols. The Registry of Reserved TPM 2.0 Handles and Localities has allocated a range of handles for the Platform Certificates. Section 7.8 “NV Memory” provides a range for the Platform Manufacturer to store the IDevID Certificate. Having a consistent place for storing IDevID Credentials facilitates interoperability and consistency for the creation of LDevIDs, which is a User function.

7.5 Storage Hierarchy

The TPM provides the ability to enable and disable the Storage Hierarchy by controlling the *shEnable value*. Since the TPM initializes with *shEnable* set by default, Platform Manufacturers and Platform Administrators need take no further action to enable it.

The Storage Hierarchy includes the Storage Primary Seed, the Owner Authorization Value, the Owner Authorization Policy, and Storage Hierarchy Primary Keys, better known by their legacy name as Storage Root Keys (SRKs). Section 7.2 “Primary Seeds” discusses Primary Seeds and Section 7.1.1 “Owner Authorizations” covers Owner Authorizations. Multiple key hierarchies rooted in distinct SRKs may co-exist within the same TPM. However, the Platform Administrator (i.e. TPM Owner) must authorize the creation of each SRK.

Legacy TPMs (i.e. version 1.2 and earlier) accommodated only one SRK and thus only one key hierarchy. By convention, Owners often set the authorization for the SRK to a well-known Authorization Value of all zeros so that multiple operating systems and applications could provision their own key hierarchies underneath it. For version 2.0, operating systems and applications may now create and manage their own SRK and the key hierarchy underneath it. The creation of a shared SRK facilitates backwards compatibility with the behavior of legacy operating systems and applications, as well as for operating systems and applications that want to use a shared SRK without the extra overhead of managing their own key hierarchies.

The creation of an SRK requires Owner Authorization through utilization of the Owner Authorization Value or Owner Authorization Policy. Since this document provides the Owner wide latitude in forming Owner Authorization, it cannot predict how the Owner will use policy to delegate SRK creation to instantiate non-persistent shared SRKs. However, use of a persistent SRK requires the authorization for that key and does not require Owner Authorization.

7.5.1 Storage Primary Key (SRK) Templates

This document uses “EK Credential Profile 2.0” [19] as a reference for building shared SRK Templates. For all templates for shared SRKs make the following changes to the EK Template: in the *objectAttributes* field, set the *userWithAuth* bit, clear the *adminWithPolicy* bit, and set the *noDA* bit. Also, set *authPolicy* to the Empty Buffer (see Table 1). Setting *noDA* to true disengages the dictionary attack mitigations because of the convention of assigning a well-known Authorization Value of all zeroes to shared SRKs. These authorization settings are appropriate for SRKs shared across multiple operating systems and applications.

For non-shared SRKs, this document allows the Platform Administrator (i.e. TPM Owner) discretion with respect to setting the authorization for them.

Parameter	**Type**	**Content**
objectAttributes->userWithAuth	TPMA_OBJECT->BIT	1
objectAttributes->adminWithPolicy	TPMA_OBJECT->BIT	0
objectAttributes->noDA	TPMA_OBJECT->BIT	1
authPolicy	TPM2B_DIGEST	
Size	UINT16	0x0000
Buffer	BYTE	NULL

Table 1: Authorization for Shared SRKs

7.5.2 Storage Primary Objects and Handles

The TCG provides guidance through the Technical Committee on the placement of persistent SRKs. The TCG provides no guidance on how to store non-persistent SRKs. Section 7.8 "NV Memory" contains a suggested location for the storage of a shared persistent SRK.

7.6 Golden Measurements

Platform Manufacturers should deliver a list of expected integrity measurements of the platform BIOS, firmware, and other binaries they provide "as shipped". They should use a standard format, such as the Reference Manifest specified in "Reference" [6]. Applications can use these measurements to validate the expected PCR values (see section 5.3).

The Platform Administrator may also want to configure the Platform, then use the Root of Trust for Reporting (i.e. the TPM with the LDevID) to create a new set of measurements representing the state of the Platform following configuration, and then store the Baseline PCRs in an appropriate place (locally or remotely) for use in accessing VPNs, protected content, etc.

This guidance allows Platform Manufacturers flexibility on the delivery method of the Golden Measurements and Baseline Measurements to the Platform and Enterprise Administrators for supporting Attestation Use Cases.

7.7 Dictionary Attack Parameter Defaults

Section 19.11 “Dictionary Attack Protection” of [24] and Section 25.3 “TPM2_DictionaryAttackParameters” of “TPM 2.0 Commands Specification” [26] provide information about dictionary attack parameters. Lockout Administrators may set three parameters: the number of failures before lockout occurs (*maxTries*), the time in seconds before an automatic decrement takes place (*recoveryTime*), and the time in seconds between unsuccessful attempts to use the authorization *lockoutAuth* (*lockoutRecovery*). “TPM 2.0 Supporting Routines” [27], Section 8.2.3.1, “DAPreInstall_Init()” sets the default values to 3, 1000, 1000, respectively. This document does not recommend values for these parameters.

If the Lockout Administrator sets *maxTries*, *recoveryTime*, and *lockoutRecovery* to the recommended settings for Windows 8 certification values of 32, 7200, and 86400 respectively, this allows for approximately 4400 tries per year without Lockout Administrator intervention (see “Microsoft TPM Fundamentals” [7]). If the Lockout Administrator uses *lockoutPolicy* instead of *lockoutAuth*, he is not restricted on when he can reset lockout if he has the authorization (i.e. the TPM ignores *lockoutRecovery*).

7.8 NV Memory

The TPM 2.0 specification provides space for persistent objects in NV Memory. During the TPM Manufacturer and Platform Manufacturer provisioning processes, they will store certain items in reserved locations of the NV Memory. The following table lists descriptions and the locations of NV Memory space reserved for provisioning objects not otherwise specified in the Registry. If the TCG updates the Registry, then the updates will supersede the affected items in the table. The value for EK Certificate that appears in the table below is consistent with the requirements for Windows 10 provided by Microsoft in “Microsoft TPM 2.0 System Fundamentals” [9].

The TCG does not compel TPM Manufacturers, Platform Manufacturers, and Users to allocate and use the physical space associated with the handles it reserves for the stated purposes. If the TPM Manufacturer, Platform Manufacturer, and the Users choose to store the EK, the EK Certificate, the SRK, the IDevID Key, and/or IDevID Certificate in NV Memory, then Table 2: Reserved Handles for TPM Provisioning Fundamental Elements contains the recommended handles within NV Memory for those values. Physical space in NV Memory is available on a first come first serve basis. Once the TPM allocates all available physical space in NV Memory, then no one else can store additional keys and certificates in NV Memory. This document provides no guidance on the amount of physical storage TPM manufacturers must build into their products. It is up to Platform Work Groups to recommend the appropriate amount of physical storage through their platform-specific TPM Profiles. The following is a copy of what is in the TCG’s published registry. It is here only for convenience, as that registry/catalogue is the normative reference.

Description	Reserved Handles
EK	0x81010001
EK Certificate	0x01C00002
SRK	0x81000001
IDevID Key	0x81020000
IDevID Certificate	0x01C90000

Table 2: Reserved Handles for TPM Provisioning Fundamental Elements

7.9 Security Certification

The TCG maintains a certification program for TPMs based on Common Criteria. However, consumers of products with TPMs may request certifications based on other criteria. Today, TPM vendors choose from among Common Criteria using "TCG Protection Profile PC Client Specific TPM 2.0 Specification L00 V1R1.16" [28] or "FIPS 140-2 Security Requirements for Cryptographic Modules" [10]. FIPS 140-2 ensures the TPM Manufacturer properly implemented the cryptographic features while Common Criteria ensures it properly implemented security features.

7.10 Defense against Advanced Persistent Threat (APT)

TCG specifications ensure that Platforms behave in a consistent manner. "NIST SP 800-147 BIOS Protection" [5]and "NIST SP 800-147B BIOS Protection for Servers" [11] ensures that procedures to update the boot firmware blocks malicious or unauthorized modifications to it. "Draft NIST SP 800-155 BIOS Integrity Measurements" [12][3] and "NIST SP 800-131A Transitioning Cryptographic Algorithms and Key Lengths" [2] provide guidance on how boot firmware should utilize integrity measurements with a hardware Root of Trust such as the TPM. Boot firmware should contain TPM drivers and follow both TCG and "Draft NIST SP 800-155 BIOS Integrity Measurements" [12] guidelines regarding the population of the TPM Platform Configuration Registers (PCRs).

[3] TCG recommends the reader refer to the current latest revisions of the Draft NIST documents

8. Utilities and Capabilities Provided to the Platform Administrator

This section recommends functional capabilities for the TPM provisioning process, including generating an EK and generating an SRK. The TPM restricts certain provisioning steps to specified parties depending on the Authorizations required to perform them. A Platform that supports these recommendations provides Users and Platform Administrators with a reliable and highly functional experience in provisioning and managing its security aspects. This guidance document assumes that Platform Administrators who replace the Platform Manufacturer-supplied operating system and applications will provide software with these TPM functional capabilities.

This document recommends that a Platform Manufacturer follow TCG Specifications, where available, regarding the installation of software that provides the following functionality in the pre-OS environment (e.g. Platform initialization firmware), labeled (PRE), the OS environment either as part of the OS or as an application, labeled (OS), or in both, labeled (PRE + OS). For those cases in which TCG guidance does not exist, or the TCG guidance makes no recommendation for a particular functionality listed below, this document makes the following recommendations.[4]

To support the usage classes of *Identity, Storage,* and/or *Attestation*, this document recommends that Platform Manufacturers provide the following properties:

- A pre-boot environment and an operating system with applications that supports interfaces that communicate with the TPM (PRE + OS);
- A pre-boot environment that meets the appropriate platform-class specifications for populating the PCRs (PRE);
- Functionality that provisions and manages the Authorization Values and Policies for the following administrative roles: Platform Administrator (i.e. TPM Owner), Privacy Administrator, and Lockout Administrator (PRE + OS);
- Functionality that sets the Dictionary Attack Parameters.

To support the usage classes of *Identity* and/or *Attestation*, this document recommends that Platform Manufacturers provide additionally the following three utilities:

- Create EKs, store EKs in persistent memory, and verify EK Certificates (PRE + OS);
- Create device identity (IDevID) keys, request and/or generate IDevID certificates, locate IDevID Certificates, and verify IDevID certificates against IDevID keys. (PRE + OS);

[4] In some Platforms the line between PRE and OS may be blurred

- Create a restricted signing key (a.k.a. Attestation Identity Key or AIK) and certify it for use to report (attest) PCR values. (PRE + OS).

To support the usage classes of *Storage* and/or *Attestation*, the document recommends that Platform Manufacturer provide the following two functionalities:

- Generate SRKs (OS).
- Clear, and thus cryptographically erase, the Storage Hierarchy, and NV Indexes (PRE + OS).

9. TPM Manufacturer Provisioning

This section presents a baseline of recommendations for the provisioning of TPMs by the TPM Manufacturer regardless of the type of Platform. The document further delineates these recommendations by the support they provide for the three classes of use cases as described in Section 5, namely *Identity*, *Storage*, and *Attestation*. Adherence to these recommendations promotes interoperability for enterprise management software that manages heterogeneous systems of Platforms and network devices, while leaving room for specialization of specific types of Platforms.

If the TCG publishes a platform-specific specification or guidance for TPMs, then the TPM Manufacturer is expected to follow those specifications and guidance for those Platforms. If platform-specific specification or guidance fails to address any of the recommendations in this section (i.e. that are silent with respect to one or more of the recommendations), this document recommends the TPM Manufacturer to follow the recommendations as listed below in order to enhance the interoperability from an enterprise management perspective. In those cases in which platform-specific specification does not exist, this document recommends the guidance in this section.

This document assumes at the beginning of this process that the TPM is in a clean state and contains no Endorsement Keys, no Endorsement Certificates, or any other objects prior to provisioning. This document provides no guidance on when or how TPM Manufacturers implement the following recommendations, only that they present a consistently provisioned TPM to the Platform Administrator at the end of this process.

The following common recommendations support the usage classes *of Identity, Storage,* and/or *Attestation.*

Cryptographic Algorithm Support – Ensure the TPM supports SHA256, AES-128 in CFB mode, and at least one of the following asymmetric algorithms: 1) RSA-2048, 2) ECC-P256.

Security Certification – Use an established scheme that satisfies target market requirements to certify the Trusted Platform Module.

Reserved Locations – Make available the reserved locations as indicated by Table 2: Reserved Handles for TPM Provisioning Fundamental Elements for the storage of persistent objects in the NV Index and in Persistent Memory.

The following additional recommendations support the usage classes of *Identity* and/or *Attestation*. Note that TPM Manufacturers can only perform these recommendations if the hardware and software are intact before the supply chain presents the TPM to the Platform Manufacturer. If the Platform Manufacturer must assemble a TPM with distinct hardware and firmware or software, then this guidance recommends deferring these steps to the Platform Manufacturer's provisioning process.

Endorsement Primary Seed (EPS) – Populate the endorsement primary seed (EPS) according to section 14.3.1 “Introduction” of “TPM 2.0 Architecture Specification” [24].

Endorsement Key – Create at least one Endorsement Key using an approved template from Section 7.4.1.1 "Endorsement Primary Key Templates" using default values.

[1] *Endorsement Key Certificate* – Create the Endorsement Key Certificate compliant to "EK Credential Profile 2.0Algorithm Registry

Trusted Computing Group, "TCG Algorithm Registry", Family "2.0", Level 00 Revision 01.22, Trusted Computing Group, Beaverton, OR, February 9, 2015

EK Credential Profile 2.0" [19] and store it an NV Index location as recommended by Section 7.8 "NV Memory".

10. Platform Manufacturer Provisioning

This section presents a baseline of recommendations for the provisioning of TPMs by the Platform Manufacturer regardless of the type of Platform. The document further delineates these recommendations by the support they provide for the three classes of use cases as described in Section 5, namely *Identity*, *Storage*, and *Attestation*. Adherence to these recommendations promotes interoperability for enterprise management software that manages heterogeneous systems of Platforms and network devices, while leaving room for specialization of specific types of Platforms.

If the TCG publishes a platform-specific specification or guidance, then the Platform Manufacturer is expected to follow those specifications and guidance for those Platforms. If platform-specific specification or guidance fails to address any of the recommendations in this section (i.e. that are silent with respect to one or more of the recommendations), this document recommends the Platform Manufacturer to follow the recommendations as listed below in order to enhance the interoperability from an enterprise management perspective. In those cases in which platform-specific specification does not exist this document recommends the guidance in this section.

This document assumes at the beginning of this process that the TPM is in a state consistent with the recommendations listed in the TPM Manufacturer Provisioning section appropriate for the use cases it supports. In some cases, the Platform Manufacturer assembles TPM hardware and software. In those cases, the Platform Manufacturer then performs steps from the previous section the TPM Manufacturer could not have completed before proceeding with the steps in this section. This document provides no guidance on when or how Platform Manufacturers implement the following recommendations, only that they present a consistently provisioned TPM to the Platform Administrator at the end of this process.

The following common recommendations support any of the usage classes of *Identity*, *Storage*, and/or *Attestation*.

Boot Firmware Protection[5] - Install compliant boot firmware following security guidance published by the TCG and NIST for designing and installing boot firmware.

Functional Requirements - Install pre-boot and OS drivers and software for functionality as required in section 8 “Utilities and Capabilities Provided to the Platform Administrator”. Follow the OS manufacturer’s recommended guidelines for configuring boot loaders and OS loaders, including a TPM device driver and software to configure it.

Reserved Locations - Make available the reserved locations as indicated by Table 2: Reserved Handles for TPM Provisioning Fundamental Elements for the storage of persistent objects in the NV Index and in Persistent Memory.

[5] This recommendation primarily fulfills security requirements as discussed in previous sections of this document.

The following additional recommendations support the *Identity* usage class.

IDevID Key[6] - Create and store an IDevID Key Pair using an approved template from section 7.4.2 "IDevID Keys ". Encrypt the key pair with the EK public key and store the resulting blob in a known external location so that Platform software can import it into the TPM later.

IDevID Key Certificate[6] - Create the IDevID Certificate using an approved template from section 7.4.2.3 "IDevID Credential and Credential Handle" to generate an IDevID Certificate. Store the IDevID Credential in a safe location that prevents unauthorized modification and deletion so that Platform software can load it into the TPM later.

Platform Authorization[7] – The platform firmware should set *platformAuth* and *platformPolicy* to one of the first three combinations illustrated in the "Hierarchy Control Setting Combination" table in section 13.2 "Controls" of "TPM 2.0 Architecture Specification" [24] as specified in specifications from platform-specific work groups. Since the TPM sets *platformAuth* and *platformPolicy* to the Empty Buffer by default after each TPM Reset and Restart, the platform firmware should reinstate non-Empty Buffer values of *platformAuth* and *platformPolicy* following each TPM Reset and Restart.

Platform Hierarchy[7] –The platform firmware may disable the Platform Hierarchy by clearing *phEnable* to 0, which means that subsequent platform firmware, operating systems, and applications cannot use *platformAuth* and *platformPolicy* to authorize any TPM action until the Platform reboots.

The following additional recommendation supports the *Storage* and *Attestation* usage classes.

Golden Integrity Measurements – Generate Golden Integrity Measurements using formats specified in "Reference Manifest Schema" [6] and "Platform Trust Services" [14] to represent the expected default values of the integrity measurements which the boot firmware and subsequent code generates and extends into TPM PCRs. Make the Golden Integrity Measurements available for administrative purposes to Platform Administrators.

[6] These recommendations are primarily for the touch-free class of devices. However, any class of device can use them.

[7] These recommendations primarily fulfill security requirements as discussed in previous sections of this document.

11. *Platform Administrator Provisioning*

This section presents a baseline of recommendations for the provisioning and deprovisioning of TPMs by the Platform Administrator regardless of the type of Platform. The document further delineates these recommendations by the support they provide for the three classes of use cases as described in Section 5, namely *Identity*, *Storage*, and *Attestation*. Adherence to these recommendations promotes interoperability for enterprise management software that manages heterogeneous systems of Platforms and network devices, while leaving room for specialization of specific types of Platforms.

This document assumes at the beginning of the provisioning process that the TPM is in a state consistent with the recommendations listed in the TPM Manufacturer Provisioning and Platform Manufacturers Provisioning sections appropriate for the use cases it supports. This document provides no guidance on when or how a Platform Administrator implements the following recommendations, only that it presents a consistently provisioned TPM to the User and the user's applications following this process.

The customer purchasing the Platform may be a direct User of the Platform. In this case he or she will likely rely, either directly or indirectly, on the Platform Manufacturer to make reasonable choices on security. If the customer is an IT administrator, an acquisition manager, a CTO, a CIO, a CSO, or some other Platform or network security subject matter expert, he or she will likely base his or her decision on security options offered by the Platform Manufacturer. They may also base their decision on compliance with certain security policy requirements either they set, or set by federal or state regulations, or set by corporate policy. In any of these cases, the Platform Manufacturers have provisioned their Platforms and the TPMs contained in them according to the provisions set forth in previous sections. Note that these provisioning steps contain choices in algorithms. The market dynamics between the Platform Manufacturers offering products with certain choices and the customers choosing products that comply with their policies will ultimately determine which of the choices will become de facto standards.

For the purposes of this section and the subsections contained therein, the Platform Administrator is either a real person physically present on a HCI-enabled Platform or a software proxy standing in for the Platform Administrator.

11.1 Provisioning the TPM

This section presents a baseline of recommendations for provisioning TPMs by a Platform Administrator. Platforms that contain software that automates these recommendations with as little User (Platform Administrator) intervention as possible enhances the likelihood that the end state of the TPM will support interoperability for remote enterprise management software.

The following common recommendations support *Identity, Storage,* and *Attestation* usage classes.

Clear the TPM[8] - Clear the Storage and Endorsement Hierarchies**.** Clearing the TPM will invalidate and cryptographically erase all keying material left in the Storage Hierarchy by previous Owners. It will also invalidate the binding of primary keys to the Endorsement Hierarchy. However, Platform Administrators can reconstitute primary keys and rebind to the Endorsement Hierarchy by using the templates.

The following additional recommendations support the *Identity* and *Attestation* usage classes.

EK Create - Create the EK as described in section 7.4 "Endorsement Hierarchy".

EK Verify - Obtain and verify the authenticity and integrity of the Endorsement Certificate issued by the TPM Manufacturer by verifying its signature. Verify that the public portion of the Endorsement Key created in the previous step matches the public key in the certificate.

EK Persist - Store the EK in persistent memory at the location recommended in Section 7.8 "NV Memory".

IDevID Verify[9] - Obtain and verify the contents, authenticity, and integrity of the IDevID Certificate issued by the Platform Manufacturer if it exists by verifying its signature. Verify the public portion of the IDevID key matches the public key in the certificate.

IDevID Persist[9] - Store the IDevID key pair in persistent memory and the IDevID Certificate in NV Index at the location recommended in Section 7.8 "NV Memory".

Local Device Identity - Create a restricted signing key and get it certified for use as a device identity (for touch-free provisioning, per "IEEE 801.1AR Secure Device Identity" [3]). This certified restricted signing key is the Platform Administrator's identity for this Platform. The Platform Administrator or User can utilize the restricted signing key as an Attestation Identity Key (AIK). If an IDevID key exists then use "IEEE 801.1AR Secure Device Identity" [3].

Endorsement Authorization (Privacy Administrator) - Set *endorsementAuth* to 256 bits chosen randomly and the *endorsementPolicy* to either a *policyDigest* or the Empty Buffer as the proxy for the Privacy Administrator.

The following additional recommendations support the *Storage* and *Attestation* usage classes.

Owner Authorization - Set *ownerAuth* to 256 bits chosen randomly and *ownerPolicy* either to a *policyDigest* or to the Empty Buffer.

SRK Create and Persist - Create a shared SRK using a template described in section 7.5.1 "Storage Primary Key (SRK) Templates" and store the default shared SRK in persistent memory at 0x81000000. Use the template in section 2.1.5.1 "RSA Template" of "EK Credential Profile 2.0" [19] for shared RSA SRKs and section 2.1.5.2 "ECC Template" of "EK Credential Profile 2.0" [19]for shared ECC SRKs with the changes to authorizations as outlined above and summarized in Table 1: Authorization for Shared SRKs.

[8] This recommendation primarily fulfills security requirements as discussed in previous sections of this document.

[9] These recommendations are primarily for the touch-free class of devices. However, any class of device can use them.

Lockout Administrator Authorization - Set *lockoutAuth* to 256 bits chosen randomly and set *lockoutPolicy* to a *policyDigest*.

Golden Integrity Measurements - Compare the Golden Integrity Measurements provided by the manufacturer against the current PCR values.

Baseline PCRs - Establish Baseline PCRs by configuring and rebooting the Platform.

11.2 De-Provisioning or End-of-Life Provisioning for the TPM

This section describes the Platform Administrator's responsibilities and steps for decommissioning a TPM that contains sensitive values and materials. The following recommendation supports the *Identity*, *Storage*, and *Attestation* usage classes.

Clear the TPM[10] - Clear the Storage Hierarchy and empty the Endorsement Hierarchy.

The Platform Administrator can use the Dictionary Attack reset authorization or policy to do this. The OS can use "PPI Specification 1.2" [22] to do this. The Platform Administrator may use the BIOS directly to accomplish this. This will not clear the Endorsement Primary Seed (EPS).

[10] This recommendation primarily fulfills security requirements as discussed in previous sections of this document.

12. Appendix: Bibliography

[1] NIST SP 800-90A DRBG

Barker, Elaine, and John Kelsey, "NIST Special Publication 800-90A Revision 1: Recommendation for Random Number Generaton Using Deterministic Random Bit Generators," National Institute for Standards and Technology, Gaitherburg, MD, June 2015.

[2] NIST SP 800-131A Transitioning Cryptographic Algorithms and Key Lengths

Barker, Elaine and Allen Roginsky, "NIST Special Publication 800-131A Revision 1: Transistions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths", NIST, November 2015

[3] IEEE 801.1AR Secure Device Identity

Borza, Mike (ed.) and Max Pritikin (ed.), "802.1AR-2009 – IEEE Standard for Local and Metropolitan Area Networks - Secure Device Identity", IEEE Computer Society, New York, New York, December 10, 2009

[4] NIST SP 800-164 Hardware-Rooted Security in Mobile Devices

Chen, Lily, Joshua Franklin, and Andrew Regenscheid, "NIST Special Publication 800-164: Guidelines on Hardware Rooted Secuirty in Mobile Devices (Draft)", NIST, October 2012

[5] NIST SP 800-147 BIOS Protection

Cooper, David, William Polk, Andrew Regenscheid, and Murugiah Souppaya, "NIST Special Publication 800-147: BIOS Protection Guidelines", NIST, April 2011

[6] Reference Manifest Schema

Hardjono, Thomas (ed.), "TCG Infrastructure Working Group Reference Manifest (RM) Schema Specification", version 1.0, revision 1.0; Trusted Computing Group; Beaverton, OR; November 17, 2006

[7] NIST SP 800-88 Guidelines for Media Sanitization

Kissel, Richard, Andrew Regenscheid, Matthew Scholl and Kevin Stine, "NIST Special Publication 800-88: Guidelines for Media Sanitization", Revision 1, NIST, Gaithersburg, MD, December, 2014

[8] Microsoft TPM Fundamentals

Microsoft, "TPM Fundamentals", https://technet.microsoft.com/en-us/library/jj889441.aspx#BKMK_HowTPMmitigates, visited April 2, 2015

[9] Microsoft TPM 2.0 System Fundamentals

Microsoft, "System.Fundamentals.TPM20", https://msdn.microsoft.com/en-us/library/windows/hardware/dn932828%28v=vs.85%29.aspx, visited April 29, 2015

[10]FIPS 140-2 Security Requirements for Cryptographic Modules

NIST, "Security Requirements for Cryptgraphic Modules", NIST, May 25, 2001

[11]NIST SP 800-147B BIOS Protection for Servers

Regenscheid, Andrew, “NIST Special Publication 800-147B: BIOS Protection Guidelines for Servers”, NIST, August, 2014

[12]Draft NIST SP 800-155 BIOS Integrity Measurements

Regenscheid, Andrew, Karen Scarfone, “NIST Special Publication 800-155: BIOS Integrity Measurement Guidelines (Draft)”, NIST, December 2011

[13]Critical Security Controls

SANS, “Critical Security Controls”, http://www.sans.org/critical-security-controls, visited September 26, 2014

[14]Platform Trust Services

Smith, Ned (co-editor) and Greg Kazmierczak (co-editor); “TCG Infrastructure Working Group Platform Trust Services Interface Specification (IF-PTS); Version 1.0, Revision 1.0, Trusted Computing Group, Beaverton, OR; November 17, 2006.

[15]Trusted Computing Best Practices

TCG Best Practices Committee, “Design, Implementation, and Usage Principles”,Trusted Computing Group, Beaverton, OR, February 2011

[16]TPM 2.0 Mobile Refererence Architecture

Trusted Computing Group, “Committee Specification TPM 2.0 Mobile Reference Architecture”, Family “2.0”, Level 00, Revision 138, Trusted Computing Group, Beaverton, OR, April 4, 2014

[17]Registry of Reserved TPM 2.0 Handles and Localities

Trusted Computing Group, "Registry of Reserved TPM 2.0 Handles and Localities," Trusted Computing Group, Beaverton, OR, October 11, 2013.

[18]Algorithm Registry

Trusted Computing Group, “TCG Algorithm Registry”, Family “2.0”, Level 00 Revision 01.22, Trusted Computing Group, Beaverton, OR, February 9, 2015

[19]EK Credential Profile 2.0

Trusted Computing Group, "TCG EK Credential Profile for TPM Family 2.0, revision 14," Trusted Computing Group, Beaverton, OR, 2014.

[20]EFI Protocal Specification 2.0

Trusted Computing Group, Draft “TCG PC Client Platform EFI Protocol Specification”, Family “2.0”, Level 00 Revision 00.04, Trusted Computing Group, Beaverton, OR March 13, 2015

[21]PC TPM 2.0 Profile

Trusted Computing Group, "TCG PC Client Platform TPM Profile (PTP) Specification", Family “2.0”, Level 00 Revision 00.35; Trusted Computing Group, Beaverton, OR, March 13, 2014.

[22]PPI Specification 1.2

Trusted Computing Group, “TCG Physical Presence interface Specification”, Specification Version 1.2, Revision 1.00, Trusted Computing Group, Beaverton, OR, February 10, 2011

[23]TPM 1.2 Keys for Platform Identity

Trusted Computing Group, "TPM Keys for Platform Identity for TPM 1.2," Trusted Computing Group, Beaverton, OR, May 22, 2014.

[24]TPM 2.0 Architecture Specification

Trusted Computing Group, "Trusted Platform Module Library Part 1: Architecture," Family “2.0” Level 00 Revision 01.16, Trusted Computing Group, Beaverton, OR, October 30, 2014.

[25]TPM 2.0 Structures Specification

Trusted Computing Group, "Trusted Platform Module Library Part 2: Structures," Family “2.0” Level 00 Revision 01.16, Trusted Computing Group, Beaverton, OR, October 30, 2014.

[26]TPM 2.0 Commands Specification

Trusted Computing Group, "Trusted Platform Module Library Part 3: Commands," Family “2.0” Level 00 Revision 01.16, Trusted Computing Group, Beaverton, OR, October 30, 2014.

[27]TPM 2.0 Supporting Routines

Trusted Computing Group, "Trusted Platform Module Library Part 3: Supporting Routines," Family “2.0” Level 00 Revision 01.16, Trusted Computing Group, Beaverton, OR, October 30, 2014.

[28]TPM 2.0 Protection Profile

Trusted Computing Group, "TPM Keys for Platform Identity for TPM 1.2," Trusted Computing Group, Beaverton, OR, May 22, 2014.

13. Appendix: Users

Defining the "user" of the TPM helps to clarify the requirements for the TPM's provisioning state. Usage scenarios provide a high-level, real-life example of how Users use the products, or in this case, the standard that the product implements. This appendix presents the usage scenarios from the User's perspective. The use cases in section 5 provide more of a workflow level of detail.

User Role:	Product Designer
Primary Goal:	Develop a product that provides endpoint or network security in an interoperable way.
Background:	Technical- and business-minded. Has a product (or an idea for a product) that requires interoperability with other security products. Does not want to reinvent the wheel for basic network or endpoint security functions. Requires roots of trust as a basis for security functionality.
Typical Usage:	Wants to make use of the TPM as a root of trust, and build upon that trust to create a product that meets a complex endpoint or network security use case.
Motivations and Expectations:	Often profit-driven. Needs to know what state the TPM—regardless of Platform Manufacturer, operating system or Platform-- will be in to ensure that their product will work when it goes to market.

User Role	Solution Architect
Primary Goal:	Design a network that is resistant to outsider attacks and insider threats.
Background:	Primary consideration is making sure the network works for the User. Security is a large consideration. Wants robust security, but has difficulty making many network security products work together. Cannot allow security to hinder the work of the network.
Typical Usage:	Looking for standards-based products that are interoperable and can be expected to solve one or more clearly defined problems. Needs these solutions to fit in to existing network.
Motivations and Expectations:	Interoperability, ease-of-use, and security. Needs the big picture, not detailed product requirements.

User Role:	Solution Implementer
Primary Goal	Set up network equipment efficiently and easily
Background	Detailed technical knowledge of how the network works (and what changes would make it break). Provides valuable input during acquisition and architecture design.
Typical Usage	Configures network equipment to accomplish network security goals. Leverages knowledge of standards and equipment functionality to configure them to support security use cases.
Motivations and Expectations:	Wants to know what features Platforms support and how to configure them. Desires a common language for communicating between devices on the network to simplify job. Needs extensive understanding of the details of the protocols and schema each equipment uses.

User Role:	Platform Administrator (Operations and Maintenance)
Primary Goal:	Keep network resources available to authorized Users and secure from unauthorized Users
Background:	Often under-resourced. Uses many non-interoperable tools to manage network oversight. Balances security needs with User needs. Has to prove compliance to regulations, but often lacks the ability to gather necessary data
Typical Usage:	24/7 response to threats, vulnerabilities, access requests, etc.
Motivations and Expectations:	Wants security solutions that work together and that do not require a lot of time to manage. Wants to be able to prevent attacks while not denying access to authorized Users. Wants to be able to “check the boxes” for regulation without expending a lot of energy.

User Role:	Enterprise Endpoint User
Primary Goal:	Access endpoint and network resources
Background:	Has a job to perform and needs to access endpoint and/or network resources to do it. Willing to work around security as needed to get job done. Does not see the security of the network as their primary mission.
Typical Usage:	Day-to-day access to resources, both on enterprise owned and personally owned Platforms.
Motivations and Expectations:	Wants endpoints and network access to “just work”. Does not want to spend any time at all on security. Sees security measures as a hindrance to productivity.