

A Practical Guide to TPM 2.0

Using the Trusted Platform Module
in the New Age of Security



Will Arthur

David Challener

With Kenneth Goldman

Apress
open

A Practical Guide to TPM 2.0: Using the New Trusted Platform Module in the New Age of Security

Will Arthur & David Challener

Copyright © 2015 by Apress Media, LLC, all rights reserved

ApressOpen Rights: You have the right to copy, use and distribute this Work in its entirety, electronically without modification, for non-commercial purposes only. However, you have the additional right to use or alter any source code in this Work for any commercial or non-commercial purpose which must be accompanied by the licenses in (2) and (3) below to distribute the source code for instances of greater than 5 lines of code. Following this Apress rights section, you will find copyright notices for material used in this book by permission. If you wish to reuse this material, you must include the corresponding copyright language provided. For material used with permission from the Trusted Computing Group, you may have rights in addition to the rights granted by this ApressOpen license. Licenses (1), (2) and (3) below and the intervening text must be provided in any use of the text of the Work and it, along with additional copyright notices, fully describes the license granted herein to the Work.

(1) License for Distribution of the Work: This Work is copyrighted by Apress Media, LLC, all rights reserved. Use of this Work other than as provided for in this license is prohibited. By exercising any of the rights herein, you are accepting the terms of this license. You have the non-exclusive right to copy, use and distribute this English language Work in its entirety, electronically without modification except for those modifications necessary for formatting on specific devices, for all non-commercial purposes, in all media and formats known now or hereafter. While the advice and information in this Work are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

If your distribution is solely Apress source code or uses Apress source code intact, the following licenses (2) and (3) must accompany the source code. If your use is an adaptation of the source code provided by Apress in this Work, then you must use only license (3).

(2) License for Direct Reproduction of Apress Source Code: This source code, excepting the source code copyrighted by Intel as noted below, from *A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security*, ISBN 978-1-4302-6583-2 is copyrighted by Apress Media, LLC, all rights reserved. Any direct reproduction of this Apress source code is permitted but must contain this license. The following license must be provided for any use of the source code from this product of greater than 5 lines wherein the code is adapted or altered from its original Apress form. This Apress code is presented AS IS and Apress makes no claims to, representations or warranties as to the function, usability, accuracy or usefulness of this code.

(3) License for Distribution of Adaptation of Apress Source Code: Portions of the source code, excepting the source code copyrighted by Intel as noted below, provided are used or adapted from *A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security*, ISBN 978-1-4302-6583-2 copyright Apress Media LLC. Any use or reuse of this Apress source code must contain this License. This Apress code is made available at Apress.com/9781430265832 as is and Apress makes no claims to, representations or warranties as to the function, usability, accuracy or usefulness of this code.

Diagram from the section AMD Secure Technology in Chapter 22 Copyright © by Advanced Micro Devices, Inc., 2015.

Tables, commands, and diagrams reproduced with permission of Trusted Computing Group, © TCG 2014: Tables 5-1, 5-2, 5-3, 5-4, 5-5, 5-6, 5-7; code following Table 5-7; and figures 7-1, 13-3, 13-4, 13-5, 13-6, 13-7, 13-8, 13-9, 13-10, 13-11, 13-14, 13-15. See http://www.trustedcomputinggroup.org/legal_notices for current TCG license terms, conditions, and disclaimers. This document may provide you with additional rights to these items not granted in the ApressOpen rights above.

Publisher gratefully acknowledges the permission granted by Intel to use the following materials in this work. All rights and interest in that material belong to Intel: code in Chapter 7, SAPI section; code in Chapter 17; Figures 13-1, 13-2, 13-12, and 13-13; and Listings 13-1 and 13-2. Publisher grants that Intel can re-print and reuse these diagrams and source code and that these materials are being used in this book with Intel's permission.

Intel is a trademark of Intel Corporation in the U.S. and/or other countries.

ISBN-13 (pbk): 978-1-4302-6583-2

ISBN-13 (electronic): 978-1-4302-6584-9

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they aren't identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr
Associate Publisher: Jeffrey Pepper
Lead Editors: Steve Weiss (Apress); Patrick Hauke (Intel)
Coordinating Editor: Melissa Maldonado
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

About ApressOpen

What Is ApressOpen?

- ApressOpen is an open access book program that publishes high-quality technical and business information.
- ApressOpen eBooks are available for global, free, noncommercial use.
- ApressOpen eBooks are available in PDF, ePub, and Mobi formats.
- The user friendly ApressOpen free eBook license is presented on the copyright page of this book.

I dedicate my portions of this work to my wife Ruth, and sons Tim and Stephen — D. Challenger

To pastor Jon MacKinney and Intel managers Linda Zavaleta and Jody Pfotenhauer, who encouraged me to pursue an engineering degree at an age when many men start thinking about retirement. To John Pennington and Monty Wiseman: for support and mentoring. To my wife, Tammy, and daughters, Casey, Megan, and Rachel: for your patience and support as I've ridden this high-tech roller coaster for the past 30 years. Most of all to Jesus Christ, my ultimate source of security. — Will Arthur

Contents at a Glance

About the Authors	xxi
About the Technical Reviewers	xxiii
Acknowledgments	xxv
Introduction	xxvii
■ Chapter 1: History of the TPM	1
■ Chapter 2: Basic Security Concepts	7
■ Chapter 3: Quick Tutorial on TPM 2.0	23
■ Chapter 4: Existing Applications That Use TPMs	39
■ Chapter 5: Navigating the Specification	51
■ Chapter 6: Execution Environment	71
■ Chapter 7: TPM Software Stack	77
■ Chapter 8: TPM Entities	97
■ Chapter 9: Hierarchies	105
■ Chapter 10: Keys	119
■ Chapter 11: NV Indexes	137
■ Chapter 12: Platform Configuration Registers	151
■ Chapter 13: Authorizations and Sessions	163
■ Chapter 14: Extended Authorization (EA) Policies	217
■ Chapter 15: Key Management	249

■ Chapter 16: Auditing TPM Commands	263
■ Chapter 17: Decrypt/Encrypt Sessions	271
■ Chapter 18: Context Management	289
■ Chapter 19: Startup, Shutdown, and Provisioning	301
■ Chapter 20: Debugging	311
■ Chapter 21: Solving Bigger Problems with the TPM 2.0	323
■ Chapter 22: Platform Security Technologies That Use TPM 2.0	331
Index	349

Contents

- About the Authors..... xxi**
- About the Technical Reviewers xxiii**
- Acknowledgmentsxxv**
- Introductionxxvii**
- Chapter 1: History of the TPM 1**
 - Why a TPM?..... 1
 - History of Development of the TPM Specification from 1.1b to 1.2..... 2
 - How TPM 2.0 Developed from TPM 1.2 3
 - History of TPM 2.0 Specification Development 4
 - Summary..... 5
- Chapter 2: Basic Security Concepts 7**
 - Cryptographic Attacks 8
 - Brute Force 8
 - Attacks on the Algorithm Itself 10
 - Security Definitions 10
 - Cryptographic Families 12
 - Secure Hash (or Digest)..... 12
 - Hash Extend..... 13
 - HMAC: Message Authentication Code..... 14
 - KDF: Key Derivation Function 14
 - Authentication or Authorization Ticket..... 15

Symmetric-Encryption Key	15
Nonce	17
Asymmetric Keys	18
Public Key Certification	20
Summary	22
■ Chapter 3: Quick Tutorial on TPM 2.0	23
Scenarios for Using TPM 1.2	24
Identification	24
Encryption	26
Key Storage	26
Random Number Generator	27
NVRAM Storage	27
Platform Configuration Registers	28
Privacy Enablement	28
Scenarios for Using Additional TPM 2.0 Capabilities	29
Algorithm Agility (New in 2.0)	29
Enhanced Authorization (New in 2.0)	31
Quick Key Loading (new in 2.0)	34
Non-Brittle PCRs (New in 2.0)	34
Flexible Management (New in 2.0)	35
Identifying Resources by Name (New in 2.0)	36
Summary	37
■ Chapter 4: Existing Applications That Use TPMs	39
Application Interfaces Used to Talk to TPMs	39
TPM Administration and WMI	42
The Platform Crypto Provider	42
Virtual Smart Card	42

Applications That Use TPMs	42
Applications That Should Use the TPM but Don't	45
Building Applications for TPM 1.2.....	46
TSS.Net and TSS.C++.....	46
Wave Systems Embassy Suite	47
Rocks to Avoid When Developing TPM Applications	48
Microsoft BitLocker	48
IBM File and Folder Encryption.....	49
New Manageability Solutions in TPM 2.0	49
Summary	50
■ Chapter 5: Navigating the Specification	51
TPM 2.0 Library Specification: The Parts	52
Some Definitions	53
General Definitions	53
Definitions of the Major Fields of the Command Byte Stream.....	54
Definitions of the Major Fields of the Response Byte Stream	55
Getting Started in Part 3: the Commands.....	55
Data Details	60
Common Structure Constructs	61
Structure with Union.....	61
Canonicalization	62
Endianness	63
Part 2: Notation Syntax.....	63
Part 3: Table Decorations.....	64
Commonly Used Sections of the Specification.....	65
How to Find Information in the Specification	66

Strategies for Ramping Up on TPM 2.0	66
Will.....	66
Ken	68
Dave.....	68
Other TPM 2.0 Specifications	69
Summary.....	69
■ Chapter 6: Execution Environment	71
Setting Up the TPM.....	71
Microsoft Simulator	71
Building the Simulator from Source Code	72
Setting Up a Binary Version of the Simulator.....	72
Running the Simulator	72
Testing the Simulator.....	73
Setting Up the Software Stack	75
TSS 2.0	75
TSS.net	75
Summary.....	76
■ Chapter 7: TPM Software Stack.....	77
The Stack: a High-Level View	77
Feature API	79
System API	85
Command Context Allocation Functions	86
Command Preparation Functions	88
Command Execution Functions	89
Command Completion Functions.....	90
Simple Code Example.....	91
System API Test Code	93

TCTI	94
TPM Access Broker (TAB)	95
Resource Manager	95
Device Driver	96
Summary	96
■ Chapter 8: TPM Entities	97
Permanent Entities	97
Persistent Hierarchies	97
Ephemeral Hierarchy	98
Dictionary Attack Lockout Reset.....	98
Platform Configuration Registers (PCRs)	98
Reserved Handles.....	99
Password Authorization Session	99
Platform NV Enable.....	99
Nonvolatile Indexes	99
Objects	100
Nonpersistent Entities	100
Persistent Entities	101
Entity Names	102
Summary	104
■ Chapter 9: Hierarchies.....	105
Three Persistent Hierarchies	105
Platform Hierarchy.....	106
Storage Hierarchy	107
Endorsement Hierarchy	108

Privacy	108
Activating a Credential.....	109
Other Privacy Considerations.....	111
NULL Hierarchy	113
Cryptographic Primitives	113
Random Number Generator.....	114
Digest Primitives.....	114
HMAC Primitives.....	116
RSA Primitives.....	117
Symmetric Key Primitives.....	117
Summary	118
■ Chapter 10: Keys	119
Key Commands	119
Key Generator	120
Primary Keys and Seeds	120
Persistence of Keys	123
Key Cache	123
Key Authorization	124
Key Destruction	125
Key Hierarchy	125
Key Types and Attributes	125
Symmetric and Asymmetric Keys Attributes.....	126
Duplication Attributes.....	126
Restricted Signing Key.....	128
Restricted Decryption Key.....	129
Context Management vs. Loading	129
NULL Hierarchy	130

Certification..... 130

Keys Unraveled..... 132

Summary..... 135

■ **Chapter 11: NV Indexes 137**

 NV Ordinary Index..... 138

 NV Counter Index 141

 NV Bit Field Index 141

 NV Extend Index..... 142

 Hybrid Index..... 143

 NV Access Controls..... 144

 NV Written..... 145

 NV Index Handle Values 146

 NV Names 147

 NV Password..... 149

 Separate Commands 149

 Summary 150

■ **Chapter 12: Platform Configuration Registers..... 151**

 PCR Value 151

 Number of PCRs 153

 PCR Commands 153

 PCRs for Authorization 154

 PCRs for Attestation 156

 PCR Quote in Detail..... 158

 PCR Attributes..... 159

 PCR Authorization and Policy..... 160

 PCR Algorithms..... 160

 Summary..... 161

■ **Chapter 13: Authorizations and Sessions..... 163**

- Session-Related Definitions 164
- Password, HMAC, and Policy Sessions: What Are They? 165
- Session and Authorization: Compared and Contrasted 167
- Authorization Roles 170
- Command and Response Authorization Area Details 172
 - Command Authorization Area 172
 - Command Authorization Structures..... 174
 - Response Authorization Structures 175
- Password Authorization: The Simplest Authorization 176
 - Password Authorization Lifecycle..... 176
 - Creating a Password Authorized Entity..... 177
 - Changing a Password Authorization for an Already Created Entity 177
 - Using a Password Authorization 178
 - Code Example: Password Session..... 178
- Starting HMAC and Policy Sessions 182
 - TPM2_StartAuthSession Command..... 183
 - Session Key and HMAC Key Details..... 185
 - Guidelines for TPM2_StartAuthSession Handles and Parameters..... 187
 - Session Variations 187
- HMAC and Policy Sessions: Differences..... 189
- HMAC Authorization..... 190
 - HMAC Authorization Lifecycle..... 190
 - HMAC and Policy Session Code Example 193
 - Using an HMAC Session to Send Multiple Commands (Rolling Nonces) 203
 - HMAC Session Security 205
 - HMAC Session Data Structure 206

Policy Authorization.....	207
How Does EA Work?	207
Policy Authorization Time Intervals.....	209
Policy Authorization Lifecycle	210
Combined Authorization Lifecycle	215
Summary.....	216
■ Chapter 14: Extended Authorization (EA) Policies.....	217
Policies and Passwords.....	218
Why Extended Authorization?.....	218
Multiple Varieties of Authentication	219
Multifactor Authentication	219
How Extended Authorization Works.....	220
Creating Policies	222
Simple Assertion Policies	222
Command-Based Assertions	233
Multifactor Authentication.....	234
Example 1: Smart card and Password.....	234
Compound Policies: Using Logical OR in a Policy.....	237
Making a Compound Policy	240
Example: A Policy for Work or Home Computers	240
Considerations in Creating Policies.....	241
End User Role	241
Administrator Role.....	242
Understudy Role	242
Office Role	242
Home Role	242

Using a Policy to Authorize a Command.....	242
Starting the Policy	243
Satisfying a Policy	243
If the Policy Is Compound	244
If the Policy Is Flexible (Uses a Wild Card)	246
Certified Policies	247
Summary.....	248
■ Chapter 15: Key Management	249
Key Generation	249
Templates	252
Key Trees: Keeping Keys in a Tree with the Same Algorithm Set	252
Duplication	253
Key Distribution	255
Key Activation.....	256
Key Destruction	257
Putting It All Together	258
Example 1: Simple Key Management	258
Example 2: An Enterprise IT Organization with Windows TPM 2.0 Enabled Systems	259
Summary.....	261
■ Chapter 16: Auditing TPM Commands.....	263
Why Audit	263
Audit Commands	265
Audit Types	265
Command Audit	265
Session Audit.....	266
Audit Log	267

Audit Data.....	268
Exclusive Audit	268
Summary.....	269
■ Chapter 17: Decrypt/Encrypt Sessions.....	271
What Do Encrypt/Decrypt Sessions Do?.....	271
Practical Use Cases.....	271
Decrypt/Encrypt Limitations.....	272
Decrypt/Encrypt Setup	273
Pseudocode Flow	273
Sample Code	275
Summary.....	287
■ Chapter 18: Context Management	289
TAB and the Resource Manager: A High-Level Description.....	289
TAB	290
Resource Manager	291
Resource Manager Operations	291
Management of Objects, Sessions, and Sequences.....	294
TPM Context-Management Features	294
Special Rules Related to Power and Shutdown Events	296
State Diagrams	297
Summary.....	299
■ Chapter 19: Startup, Shutdown, and Provisioning.....	301
Startup and Shutdown	301
Startup Initialization	303
Provisioning.....	304
TPM Manufacturer Provisioning	305
Platform OEM Provisioning	306

End User Provisioning.....	307
Deprovisioning.....	308
Summary.....	309
■ Chapter 20: Debugging	311
Low-Level Application Debugging	311
The Problem	312
Analyze the Error Code	312
Debug Trace Analysis.....	313
More Complex Errors.....	315
Last Resort	315
Common Bugs	317
Debugging High-level Applications	317
Debug Process.....	318
Typical Bugs	318
Summary.....	321
■ Chapter 21: Solving Bigger Problems with the TPM 2.0	323
Remote Provisioning of PCs with IDevIDs Using the EK	323
Technique 1	324
Technique 2	325
Technique 3	327
Data Backups	327
Separation of Privilege	328
Securing a Server’s Logon	329
Locking Firmware in an Embedded System, but Allowing for Upgrades.....	330
Summary.....	330

■ **Chapter 22: Platform Security Technologies That Use TPM 2.0** **331**

The Three Technologies..... **331**

 Some Terms..... 332

Intel® Trusted Execution Technology (Intel® TXT) **333**

 High-Level Description 333

 How TPM 2.0 Devices Are Used..... 339

ARM® TrustZone® **341**

 High-Level Description 341

 Implementation of TrustZone 343

AMD Secure Technology™..... **346**

 Hardware Validated Boot 347

 TPM on an AMD Platform..... 348

 SKINIT 348

Summary..... **348**

Index..... **349**

About the Authors



Will Arthur is a senior staff firmware engineer in the Datacenter Engineering Group for Intel Corporation. He leads the development of authenticated code modules (ACMs) for the server version of Intel Trusted Execution Technology (TXT). As an active participant in the Trusted Computing Group's TPM and TSS working groups, he wrote the TCG TPM 2.0 System API and TPM 2.0 TAB and Resource Manager specifications, developed the TCG versions of the code that implements those specifications, and reviewed and edited the TPM 2.0 specification for readability and accuracy. Will has over 30 years of experience in low-level embedded firmware and software, the last 19 of those years with Intel. Will earned a BSCS in computer science from Arizona State University.



David Challener has been working on Trusted Computing since it started over a dozen years ago. He is currently co-chair of the TPM Working Group, and in the past has been chair of the TSS workgroup and on the TCG technical committee and Board of Directors. He has contributed to a number of other TCG specifications as well. He has a PhD in applied mathematics from the University of Illinois and currently works at The Johns Hopkins University Applied Physics Laboratory.

About the Technical Reviewers



Justin D. "Ozzie" Osborn is the chief scientist of the Commercial Device Operations Group at The Johns Hopkins University Applied Physics Laboratory. He has almost a decade of experience in software reverse engineering and embedded software development. He has worked on several projects that involved developing TPM software and performing vulnerability analyses of TPM solutions.



Monty Wiseman is a security architect in Intel's Data Center Group (DCG). His current projects include architecture for TCG, Intel's TXT technologies, Boot Guard, and other security initiatives. Monty has participated in and chaired the TCG PC Client working group and Security Evaluation working group for TPM 1.2. He participates in the TPM and other TCG workgroups and is Intel's representative on the TCG Technical Committee. Monty has 20 years of experience in desktop, network, and mainframe environments and has held security-related and other engineering positions at Novell, Fujitsu, and Control Data. He has been developing hardware and software for computers ranging from mainframes to microcomputers since 1975.

Acknowledgments

The authors gratefully acknowledge the contributions, edits, and suggestions from our external and internal reviewers:

- Ken Goldman wrote many of the chapters and ruthlessly reviewed the text for technical errors.
- Emily Ratliff and Jon Geater contributed their expertise and knowledge to the ARM and AMD sections of Chapter 22. Bill Futrall also contributed text to Chapter 22.
- Paul England, David Wooten, and Ari Singer helped us understand the specification.
- Paul England helped us understand Microsoft interfaces to the TPM.
- Monty Wiseman, Justin Osborn, Alex Eydelberg, Bill Futral, Jim Greene, and Lisa Raykowski did technical reviews.
- Patrick Hauke of Intel provided moral support and guidance throughout this process.
- We would also like to recognize the many direct and indirect contributions of the TSS and TPM WG members.

Introduction

“Seminal!”

“Riveting! I couldn’t put it down until the last page.”

“I’m exhausted from reading this book! It kept me up three nights in a row. Where’s my Ambien when I need it?”

“The suspense was killing me. I just *had* to read it straight through!”

Although these responses to our book would be gratifying, it’s doubtful that any book on digital security will ever garner this type of reaction. Digital security is the computer equivalent of disaster insurance. Few people care very much about it or give it much thought, and everyone hates paying for it ... until a catastrophe hits. Then we are either really glad we had it or really sad that we didn’t have enough of it or didn’t have it at all.

We may sound like Chicken Little crying the “the sky is falling, the sky is falling,” but mark our words: a digital security catastrophe is headed your way. We could quote a plethora of statistics about the rising occurrence of digital security threats, but you’ve probably heard them, and, quite frankly, you don’t care, or at least you don’t care enough. It’s questionable whether any preaching on our part will make you care enough until you’re personally impacted by such a calamity, but we’ll try anyway.

When your reputation is tarnished, your finances are impacted, your identity is stolen, your physical well-being is threatened, your company’s reputation and finances are harmed, and, quite possibly, your country is overthrown, then you’ll wake up to the need for cyber security. But it might be too late then. Like people living in a flood zone, the question isn’t whether the flood is coming, but rather when the disaster will hit and whether you’ll be prepared for it. The time to buy digital-security flood insurance is now! Don’t wait until the flood hits.

A Practical Guide to TPM 2.0 can be part of your digital-security insurance policy. The TPM was designed as one of the core building blocks for digital security solutions. The November 2013 “Report to the President: Immediate Opportunities for Strengthening the Nation’s Cybersecurity” recommends “the universal adoption of the Trusted Platform Module (TPM), an industry-standard microchip designed to provide basic security-related functions, primarily involving encryption keys, including for phones and tablets. Computers and devices that incorporate a TPM are able to create cryptographic keys and encrypt them so they can be decrypted only by the TPM. A TPM provides this limited but fundamental set of capabilities that higher layers of cybersecurity can then leverage. Today, TPMs are present in many laptop and desktop personal computers. They’re used by enterprises for tasks like secure disk encryption, but they have yet to be incorporated to any significant extent in smartphones, game consoles, televisions, in-car computer systems, and other computerized devices and industrial control systems. This needs to happen for such devices to be trustworthy constituents of the increasingly interconnected device ecosystem.”

Our passion in writing this book is to empower and excite a rising generation of IT managers, security architects, systems programmers, application developers, and average users to use the TPM as the bedrock of increasingly sophisticated security solutions that will stem the rising tide of threats that are being aimed at us, our employers, and our civil institutions. Furthermore, the TPM is just plain cool. How many engineers, as children, played with simple cryptography for fun? The ability to send an encrypted message to a friend appeals to the secretive part of our human nature—the same part that enjoyed playing spy games when we were young. And besides being fun, there’s something inherently, morally right about protecting people’s assets from being stolen.

The TPM 2.0 technology can accomplish this. We believe in this technology and hope to make believers of you, our readers, as well. Our hope is that you’ll get as excited about this technology as we are and “go out and do wonderful things” with it, to paraphrase Robert Noyce, one of Intel’s founders.

Why a Book?

Technical specifications are typically poor user manuals, and TPM 2.0 is no exception. One reader of the specification claimed it was “security through incomprehensibility.” Although the specification attempts to describe the functionality as clearly as possible, its prime objective is to describe how a TPM should work, not how it should be used. It’s written for implementers of TPMs, not for application writers using TPMs.

Also, for better or for worse, the detailed operations of the TPM commands are specified in C source code. The structures are defined with various keywords and decorations that permit the Word document to be parsed into a C header file. Microsoft agreed with TCG that the source code in the specification would have an open source license and could be used to implement a TPM. However, although C can describe actions very precisely, even the best code isn’t as readable as text. One of the major purposes of this book is to interpret the specification into language that is more understandable to average software developers, especially those who need to understand the low-level details of the specification.

Many readers don’t need to understand the detailed operation of the TPM and just want to know how to use the various functions. These readers expect TSS (the TCG software stack) middleware to handle the low-level details. They’re interested in how to use the new TPM features to accomplish innovative security functions. Thus, this book is just as concerned with describing how the TPM can be used as it is with explaining how it works. Throughout the book, as features are described, use cases for those features are interwoven. The use cases aren’t complete—they describe what the TPM 2.0 specification writers were thinking about when those features were designed, but the specification is so rich that it should be possible to implement many things beyond these use cases.

Audience

In writing this book, we’re trying to reach a broad audience of readers: low-level embedded system developers, driver developers, application developers, security architects, engineering managers, and even non-technical users of security applications. We hope to encourage the broadest possible adoption and use of TPMs.

Non-technical readers will want to focus on the introductory material, including the history of the TPM (Chapter 1), basic security concepts (Chapter 2), and existing applications that use TPMs (Chapter 4). Visionaries who know what they want to accomplish but aren't themselves programmers will also benefit from reading these chapters, because knowing the basic ways in which TPMs can be used may provide inspiration for new use cases.

Engineering managers, depending on their needs and technical expertise, can go as deep as they need to or want to. We hope that executives will read the book, see the possibilities provided by TPMs, and subsequently fund TPM-related projects. When they realize, for example, that it's possible for an IT organization to cryptographically identify all of its machines before allowing them onto a network, that true random number generators are available to help seed OSs' "get random number" functions, and that weaker passwords can be made stronger using the anti-dictionary-attack protections inherent in the TPM design, they may decide (and we hope they will) to make these features easily available to everyday people.

Security architects definitely need to understand the functions provided by TPM 2.0 and, depending on the applications being developed, dive deep into how the TPM works in order to understand the security guarantees provided. Linking disparate machines or different functions to provide trusted software and networks should be possible using TPM functionality as security architects get creative. Commercial availability of this capability is long overdue.

Application developers, both architects and implementers, are a significant focus of this book. These readers need to understand the TPM from a high-level viewpoint and will be especially interested in the use cases. TPM 2.0 is feature rich, and the use cases we describe will hopefully inspire creativity in developing and inventing security applications. Developers have to know the basics of symmetric and asymmetric keys and hashes in developing their applications—not the bit-by-bit computations, which are done in the TPM or support software—but rather the types of guarantees that can be obtained by using the TPM correctly.

We also want the book to be useful to embedded system developers, middle ware developers, and programmers integrating TCG technology into operating systems and boot code. The TPM now exposes more general-purpose cryptographic functions, which are useful when a crypto library isn't available due to either resource constraints or licensing issues. We hope that low-level developers will find that this book goes as deep as they need it to and that it serves as a critical tool in interpreting the specification. Toward this end, diagrams and working code examples are used to help clarify many concepts. We expect that embedded systems will increasingly use TPMs as the cost of the technology is reduced (making cryptographic computations cheap to integrate into embedded software) and as attacks on embedded software become more active.

Roadmap

If you're new to security or need a refresher, Chapter 2 gives an overview of the security concepts required to understand the book. This chapter provides high-level knowledge of cryptography: we explain symmetric and asymmetric keys, secure hash algorithms, and how a message authentication code (MAC) can be used as a symmetric key digital

signature. This chapter doesn't delve into the underlying math used to implement cryptographic algorithms; this isn't intended as a general-purpose security or cryptography textbook, because there is no need for most TPM 2.0 developers to possess that depth of knowledge.

Chapter 3 presents a high-level tutorial on TPM 2.0 and the design rationale behind it. It begins with applications and use cases enabled by TPM 1.2, all of which are also available in TPM 2.0, and then continues by describing the new capabilities that are available with the TPM 2.0 specification. This chapter should help you understand why people are excited about the technology and want to use it in their applications and environments.

Chapter 4 describes existing applications that use TPMs (currently, mostly 1.2). We assume that many of these applications will be ported to TPM 2.0. Some are open source, some are demonstration code written by academics to demonstrate what the TPM can do, some are applications that have been around a long time and that can be linked to use TPM features, and some are generally available applications written specifically to take advantage of the TPM's capabilities.

Chapter 5 provides a high-level orientation to the TPM 2.0 specification, offers pointers to critical parts of the specification, and explores some best practices for using the specification.

Chapter 6 describes the setup and use of the execution environments available for running TPM 2.0 code examples.

Chapter 7 discusses the trusted software stack (TSS). This is presented early in the book because succeeding code examples use various layers of the TSS.

Chapter 8 begins the deep dive into TPM 2.0 functionality with a description of TPM 2.0 entities: keys, data blobs, and NV indices.

Chapter 9 discusses hierarchies.

Chapter 10 covers keys.

Chapter 11 discusses NV indexes.

Chapter 12 explores PCRs and attestation.

Chapter 13 is one of the most in-depth chapters and is crucial if you're developing low-level code or architecting systems that make extensive use of sessions and authorizations.

Chapter 14 discusses enhanced authorization.

Chapter 15 explains key management.

Chapter 16 describes the TPM's auditing capabilities.

Chapter 17 examines decryption and encryption sessions and how to set them up.

Chapter 18 describes object, sequence, and session context management and the basic functionality of a resource manager.

Chapter 19 discusses TPM startup, initialization, and provisioning. In typical usage, these occur before keys and sessions are used, but knowledge of TPM entities and sessions is a prerequisite to understanding TPM initialization and provisioning. This is why we include this chapter after the previous three chapters.

Chapter 20 presents best practices for debugging TPM 2.0 applications.

Chapter 21 examines high-level applications that could use TPM 2.0 functionality.

Chapter 22 discusses platform-level security technologies that incorporate TPM 2.0 devices into their security solutions.

Assumptions

Although this is a technology book, we have tried to assume as little about our readers as possible. Code examples use C, and a working knowledge of C is useful. However, most of the concepts stand alone, and much of the book should be comprehensible to non-programmers. Security concepts are explained at a high level, and every attempt is made to make them understandable.

Some knowledge of the TPM 1.2 and 2.0 specifications is definitely beneficial but not required. We encourage you to download the TPM 2.0 specifications from www.trustedcomputinggroup.org so that you can refer to them as you read the book.