INTERNATIONAL STANDARD

ISO/IEC 11889-1

Second edition
2015-12-15

Corrected version
2016-04-01

Information technology — Trusted Platform Module Library —

Part 1: Architecture

Technologies de l'information — Bibliothèque de module de plate-forme de confiance — Partie 1: Architecture

Reference number
ISO/IEC 11889-1:2015(E)

© ISO/IEC 2015



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

CONTENTS

Foreword xiv
Introduction xv
1 Scope 1
2 Normative references 2
3 Terms and definitions 3
4 Symbols and Abbreviated Terms 12
4.1 Symbols 12
4.2 Abbreviations 13
5 Conventions 15
5.1 Bit and Octet Numbering and Order 15
5.2 Sized Buffer References 15
5.3 Numbers 16
5.4 KDF Label Parameters 16
6 ISO/IEC 11889 Organization 17
7 Compliance 19
8 Changes from Previous Versions 20
9 Trusted Platforms 21
9.1 Trust 21
9.2 Trust Concepts 21
9.2.1 Trusted Building Block 21
9.2.2 Trusted Computing Base 21
9.2.3 Trust Boundaries 21
9.2.4 Transitive Trust 22
9.2.5 Trust Authority 22
9.3 Trusted Platform Module 23
9.4 Roots of Trust 23
9.4.1 Introduction 23
9.4.2 Root of Trust for Measurement (RTM) 24
9.4.3 Root of Trust for Storage (RTS) 24
9.4.4 Root of Trust for Reporting (RTR) 24
9.5 Basic Trusted Platform Features 25
9.5.1 Introduction 25
9.5.2 Certification 26
9.5.3 Attestation and Authentication 26
9.5.4 Protected Location 29
9.5.5 Integrity Measurement and Reporting 30
10 TPM Protections 31
10.1 Introduction 31
10.2 Protection of Protected Capabilities 31
10.3 Protection of Shielded Locations 31
10.4 Exceptions and Clarifications 31
11 TPM Architecture 33
11.1 Introduction 33
11.2 TPM Command Processing Overview 33
11.3 I/O Buffer 37
11.4 Cryptography Subsystem 37

 © ISO/IEC 2015 – All rights reserved

11.4.1 Introduction 37
11.4.2 Hash Functions 37
11.4.3 HMAC Algorithm 38
11.4.4 Asymmetric Operations 38
11.4.5 Signature Operations 39
11.4.6 Symmetric Encryption 41
11.4.7 Extend 43
11.4.8 Key Generation 43
11.4.9 Key Derivation Function 43
11.4.10 Random Number Generator (RNG) Module 47
11.4.11 Algorithms 49
11.5 Authorization Subsystem 50
11.6 Random Access Memory 51
11.6.1 Introduction 51
11.6.2 Platform Configuration Registers (PCR) 51
11.6.3 Object Store 52
11.6.4 Session Store 52
11.6.5 Size Requirements 52
11.7 Non-Volatile (NV) Memory 53
11.8 Power Detection Module 53
12 TPM Operational States 54
12.1 Introduction 54
12.2 Basic TPM Operational States 54
12.2.1 Power-off State 54
12.2.2 Initialization State 54
12.2.3 Startup State 55
12.2.4 Shutdown State 58
12.2.5 Startup Alternatives 58
12.3 Self-Test Modes 59
12.4 Failure Mode 60
12.5 Field Upgrade 61
12.5.1 Introduction 61
12.5.2 Field Upgrade Mode 61
12.5.3 Preserved TPM State 64
12.5.4 Field Upgrade Implementation Options 65
13 TPM Control Domains 66
13.1 Introduction 66
13.2 Controls 66
13.3 Platform Controls 67
13.4 Owner Controls 68
13.5 Privacy Administrator Controls 68
13.6 Primary Seed Authorizations 69
13.7 Lockout Control 69
13.8 TPM Ownership 70
13.8.1 Taking Ownership 70
13.8.2 Releasing Ownership 70
14 Primary Seeds 72

© ISO/IEC 2015 – All rights reserved

14.1 Introduction ... 72
14.2 Rationale ... 72
14.3 Primary Seed Properties ... 73
14.3.1 Introduction ... 73
14.3.2 Endorsement Primary Seed (EPS) ... 73
14.3.3 Platform Primary Seed (PPS) ... 74
14.3.4 Storage Primary Seed (SPS) ... 74
14.3.5 The Null Seed ... 74
14.4 Hierarchy Proofs ... 74
15 TPM Handles ... 76
15.1 Introduction ... 76
15.2 PCR Handles (MSO=00_{16}) ... 76
15.3 NV Index Handles (MSO=01_{16}) ... 76
15.4 Session Handles (MSO=02_{16} and 03_{16}) ... 76
15.5 Permanent Resource Handles (MSO=40_{16}) ... 77
15.6 Transient Object Handles (MSO=80_{16}) ... 77
15.7 Persistent Object Handles (MSO=81_{16}) ... 77
16 Names ... 78
17 PCR Operations ... 80
17.1 Initializing PCR ... 80
17.2 Extend of a PCR ... 80
17.3 Using Extend with PCR Banks ... 80
17.4 Recording Events ... 81
17.5 Selecting Multiple PCR ... 81
17.6 Reporting on PCR ... 82
17.6.1 Reading PCR ... 82
17.6.2 Attesting to PCR ... 82
17.7 PCR Authorizations ... 83
17.7.1 Introduction ... 83
17.7.2 PCR Not in a Set ... 83
17.7.3 Authorization Set ... 83
17.7.4 Policy Set ... 84
17.7.5 Order of Checking ... 84
17.8 PCR Allocation ... 84
17.9 PCR Change Tracking ... 84
17.10 Other Uses for PCR ... 85
18 TPM Command/Response Structure ... 86
18.1 Introduction ... 86
18.2 Command/Response Header Fields ... 88
18.2.1 Introduction ... 88
18.2.2 *tag* ... 88
18.2.3 commandSize/responseSize ... 88
18.2.4 commandCode ... 88
18.2.5 responseCode ... 88
18.3 Handles ... 89
18.4 Parameters ... 89
18.5 *authorizationSize*/*parameterSize* ... 90

 © ISO/IEC 2015 – All rights reserved

18.6 Authorization Area 90
18.6.1 Introduction 90
18.6.2 Authorization Structure 92
18.6.3 Session Handles 93
18.6.4 Session Attributes (*sessionAttributes*) 93
18.7 Command Parameter Hash (*cpHash*) 95
18.8 Response Parameter Hash (*rpHash*) 95
18.9 Command Example 96
18.10 Response Example 97
19 Authorizations and Acknowledgments 99
19.1 Introduction 99
19.2 Authorization Roles 99
19.3 Physical Presence Authorization 100
19.4 Password Authorizations 101
19.5 Sessions 102
19.6 Session-Based Authorizations 102
19.6.1 Introduction 102
19.6.2 Authorization Session Formats 103
19.6.3 Session Nonces 103
19.6.4 Authorization Values 105
19.6.5 HMAC Computation 106
19.6.6 Note on Use of Nonces in HMAC Computations 107
19.6.7 Starting an Authorization Session 107
19.6.8 *sessionKey* Creation 108
19.6.9 Unbound and Unsalted Session Key Generation 109
19.6.10 Bound Session Key Generation 110
19.6.11 Salted Session Key Generation 112
19.6.12 Salted and Bound Session Key Generation 113
19.6.13 Encryption of *salt* 114
19.6.14 Caution on use of Unsalted Authorization Sessions 115
19.6.15 No HMAC Authorization 115
19.6.16 Authorization Selection Logic for Objects 116
19.6.17 Authorization Session Termination 116
19.7 Enhanced Authorization 117
19.7.1 Introduction 117
19.7.2 Policy Assertion 118
19.7.3 Policy AND 118
19.7.4 Policy OR 120
19.7.5 Order of Evaluation 122
19.7.6 Policy Assertions (Policy Commands) 122
19.7.7 Policy Session Context Values 125
19.7.8 Policy Example 126
19.7.9 Trial Policy 127
19.7.10 Modification of Policies 127
19.7.11 TPM2_PolicySigned(), TPM2_PolicySecret(), and TPM2_PolicyTicket() 128
19.8 Policy Session Creation 130
19.9 Use of TPM for *authPolicy* Computation 131

© ISO/IEC 2015 – All rights reserved

19.10 Trial Policy Session 131
19.11 Dictionary Attack Protection 132
19.11.1 Introduction 132
19.11.2 Lockout Mode Configuration Parameters 132
19.11.3 Lockout Mode 133
19.11.4 Recovering from Lockout Mode 133
19.11.5 Authorization Failures Involving lockoutAuth 134
19.11.6 Non-orderly Shutdown 134
19.11.7 Justification for Lockout Due to Session Binding 134
19.11.8 Sample Configurations for Lockout Parameters 135
20 Audit Session 136
20.1 Introduction 136
20.2 Exclusive Audit Sessions 137
20.3 Command Gating Based on Exclusivity 137
20.4 Audit Session Reporting 137
20.5 Audit Establishment Failures 138
21 Session-based encryption 139
21.1 Introduction 139
21.2 XOR Parameter Obfuscation 140
21.3 CFB Mode Parameter Encryption 140
22 Protected Storage 142
22.1 Introduction 142
22.2 Object Protections 142
22.3 Protection Values 142
22.4 Symmetric Encryption 143
22.5 Integrity 144
23 Protected Storage Hierarchy 146
23.1 Introduction 146
23.2 Hierarchical Relationship between Objects 146
23.3 Duplication 147
23.3.1 Definition 147
23.3.2 Protections 148
23.4 Duplication Group 153
23.5 Protection Group 155
23.6 Summary of Hierarchy Attributes 156
23.7 Primary Seed Hierarchies 156
23.8 Hierarchy Attributes Settings Matrix 156
24 Credential Protection 158
24.1 Introduction 158
24.2 Protocol 158
24.3 Protection of Credential 159
24.4 Symmetric Encrypt 159
24.5 HMAC 159
24.6 Summary of Protection Process 161
25 Object Attributes 162
25.1 Base Attributes 162
25.1.1 Introduction 162
25.1.2 *Restricted* Attribute 162
25.1.3 *Sign* Attribute 162

 © ISO/IEC 2015 – All rights reserved

25.1.4 *Decrypt* Attribute 163
25.1.5 Uses 163
25.2 Other Attributes 165
25.2.1 fixedTPM and fixedParent 165
25.2.2 stClear 165
25.2.3 sensitiveDataOrigin 165
25.2.4 userWithAuth 165
25.2.5 adminWithPolicy 165
25.2.6 noDA 166
25.2.7 encryptedDuplication 166
26 Object Structure Elements 167
26.1 Introduction 167
26.2 Public Area 167
26.3 Sensitive Area 168
26.4 Private Area 168
26.5 Qualified Name 169
26.6 Sensitive Area Encryption 169
26.7 Sensitive Area Integrity 170
27 Object Creation 171
27.1 Introduction 171
27.2 Public Area Template 171
27.2.1 Introduction 171
27.2.2 type 171
27.2.3 nameAlg 172
27.2.4 objectAttributes 172
27.2.5 authPolicy 172
27.2.6 parameters 172
27.2.7 unique 172
27.3 Sensitive Values 172
27.3.1 Overview 172
27.3.2 userAuth 173
27.3.3 data 173
27.4 Creation PCR 173
27.5 Public Area Creation 173
27.5.1 Introduction 173
27.5.2 type, nameAlg, objectAttributes, authPolicy, and parameters 173
27.5.3 unique 174
27.6 Sensitive Area Creation 175
27.6.1 Introduction 175
27.6.2 type 175
27.6.3 authValue 175
27.6.4 *seedValue* 175
27.6.5 sensitive 176
27.7 Creation Data and Ticket 177
27.8 Creation Resources 178
28 Object Loading 179

© ISO/IEC 2015 – All rights reserved

28.1 Introduction 179
28.2 Load of an Ordinary Object 179
28.3 Public-only Load 179
28.4 External Object Load 180
29 Object Creation in Reference Implementation 181
30 Context Management 182
30.1 Introduction 182
30.2 Context Data 183
30.2.1 Introduction 183
30.2.2 Sequence Number 183
30.2.3 Handle 184
30.2.4 Hierarchy 185
30.3 Context Protections 185
30.3.1 Context Confidentiality Protection 185
30.3.2 Context Integrity Protection 186
30.4 Object Context Management 187
30.5 Session Context Management 187
30.6 Eviction 188
30.7 Incidental Use of Object Slots 189
31 Attestation 190
31.1 Introduction 190
31.2 Standard Attestation Structure 190
31.3 Privacy 191
31.4 Qualifying Data 191
31.5 Anonymous Signing 191
32 Cryptographic Support Functions 192
32.1 Introduction 192
32.2 Hash 192
32.3 HMAC 192
32.4 Hash, HMAC, and Event Sequences 193
32.4.1 Introduction 193
32.4.2 Hash Sequence 193
32.4.3 Event Sequence 193
32.4.4 HMAC Sequence 194
32.4.5 Sequence Contexts 194
32.5 Symmetric Encryption 194
32.6 Asymmetric Encryption and Signature Operations 194
33 Locality 195
34 Hardware Core Root of Trust Measurement (H-CRTM) Event Sequence 196
34.1 Introduction 196
34.2 Dynamic Root of Trust Measurement 196
34.3 H-CRTM before TPM2_Startup() 197
35 Command Audit 198
36 Timing Components 200
36.1 Introduction 200
36.2 Clock 201
36.2.1 Introduction 201
36.2.2 *Clock* Implementation 201

 © ISO/IEC 2015 – All rights reserved

36.2.3 Orderly Shutdown of *Clock* 202
36.2.4 *Clock* Initialization at TPM2_Startup() 202
36.2.5 Setting *Clock* 203
36.2.6 *Clock* Periodicity 203

36.3 Time 204
36.4 resetCount 204
36.5 restartCount 204
36.6 Note on the Accuracy and Reliability of *Clock* 205
36.7 Privacy Aspects of Clock 206
37 NV Memory 207
37.1 Introduction 207
37.2 NV Indices 207

37.2.1 Definition 207
37.2.2 NV Index Allocation 208
37.2.3 NV Index Deletion 209
37.2.4 High-Endurance (Hybrid) Indices 209
37.2.5 Reading an NV Index 210
37.2.6 Updating an Index 211
37.2.7 NV Index in a Policy 214

37.3 Owner and Platform Evict Objects 214
37.4 State Saved by TPM2_Shutdown() 215

37.4.1 Background 215
37.4.2 NV Orderly Data 215
37.4.3 NV Clear Data 216
37.4.4 NV Reset Data 217

37.5 Persistent NV Data 218
37.6 NV Rate Limiting 220
37.7 NV Other Considerations 220

37.7.1 Power Interruption 220
37.7.2 External NV 220
37.7.3 PCR in NV 221

38 Multi-Tasking 222
39 Errors and Response Codes 223
39.1 Error Reporting 223
39.2 TPM State After an Error 223
39.3 Resource Exhaustion Warnings 223

39.3.1 Introduction 223
39.3.2 Transient Resources 223
39.3.3 Temporary Resources 224

39.4 Response Code Details 224
40 General Purpose I/O 226
41 Minimums 227
41.1 Introduction 227
41.2 Authorization Sessions 227
41.3 Transient Objects 227
41.4 NV Counters and Bit Fields 227
Annex A (normative) RSA 228

© ISO/IEC 2015 – All rights reserved

A.1 Introduction 228
A.2 RSAEP 229
A.3 RSADP 229
A.4 RSAES_OAEP 229
A.5 RSAES_PKCSV1_5 229
A.6 RSASSA_PKCS1v1_5 229
A.7 RSASSA_PSS 230
A.8 RSA Cryptographic Primitives 231
A.8.1 Introduction 231
A.8.2 TPM2_RSA_Encrypt() 231
A.8.3 TPM2_RSA_Decrypt() 231
A.9 Secret Sharing 231
A.9.1 Overview 231
A.9.2 RSA Encryption of Salt 231
A.9.3 RSA Secret Sharing for Duplication 232
A.9.4 RSA Secret Sharing for Credentials 232
Annex B (normative) ECC 233
B.1 Introduction 233
B.2 Split Operations 233
B.2.1 Introduction 233
B.2.2 Commit Random Value 233
B.2.3 TPM2_Commit() 234
B.2.4 TPM2_EC_Ephemeral() 235
B.2.5 Recovering the Private Ephemeral Key 236
B.3 ECC-Based Secret Sharing 236
B.4 EC Signing 236
B.4.1 ECDSA 236
B.4.2 ECDAA 236
B.4.3 EC Schnorr 239
B.5 Secret Sharing 240
B.5.1 ECDH 240
B.5.2 ECDH Encryption of Salt 241
B.5.3 ECC Secret Sharing for Duplication 241
B.5.4 ECC Secret Sharing for Credentials 241
B.6 ECC Primitive Operations 241
B.6.1 Introduction 241
B.6.2 TPM2_ECDH_KeyGen() 241
B.6.3 TPM2_ECDH_ZGen() 241
B.6.4 Two-phase Key Exchange 242
Annex C (normative) Support for SMx Family of Algorithms 244
C.1 Introduction 244
C.1 SM2 244
C.1.1 Introduction 244
C.1.2 SM2 Digital Signature Algorithm 245
C.1.3 SM2 Key Exchange 247

 © ISO/IEC 2015 – All rights reserved

C.2 SM3 248
C.3 SM4 248
Annex D (informative) Key Generation 249
D.1 Introduction 249
D.2 RSA Key Generation 249
D.2.1 Background 249
D.2.2 Prime Generation 249
D.2.3 Key Generation Algorithm 250
D.3 ECC Ordinary Keys 251
D.4 ECC Primary key 251
Annex E (informative) Policy Examples 252
E.1 Introduction 252
E.2 ISO/IEC 11889 (first edition) Compatible Authorization 252
Annex F (informative) Acknowledgements and contributors 254
F.1 Acknowledgements 254
F.2 Contributors 254
Bibliography 256

Tables

Table 1 — KDF Label Parameters 16
Table 2 — Block Cipher Parameters 41
Table 3 — Hierarchy Control Setting Combinations 67
Table 4 — Equations for Computing Entity Names 78
Table 5 — Separators 87
Table 6 — *Tag* Values 88
Table 7 — Use of Authorization/Session Blocks 91
Table 8 — Description of *sessionAttributes* 93
Table 9 — Command Layout for Example Command 96
Table 10 — Example Command Showing *authorizationSize* 97
Table 11 — Response Layout for Example Command 97
Table 12 — Example Response Showing *parameterSize* 98
Table 13 — Password Authorization of Command 101
Table 14 — Password Acknowledgment in Response 101
Table 15 — Session-Based Authorization of Command 103
Table 16 — Session-Based Acknowledgment in Response 103
Table 17 — Schematic of TPM2_StartAuthSession Command 107
Table 18 — Handle Parameters for TPM2_StartAuthSession 108
Table 19 — Format to Start Unbounded, Unsalted Session 109
Table 20 — Format to Start Bound Session 111

© ISO/IEC 2015 – All rights reserved

Table 21 — Format to Start Salted Session 112
Table 22 — Format to Start Salted and Bound Session 113
Table 23 — Mapping of Hierarchy Attributes 156
Table 24 — Allowed Hierarchy Settings 156
Table 25 — Mapping of Functional Attributes 163
Table 26 — ISO/IEC 11889 (first edition) Correspondence 164
Table 27 — Public Area Parameters 167
Table 28 — Sensitive Area Parameters 168
Table 29 — Standard Attestation Structure 190
Table 30 — Contents of the ORDERLY_DATA Structure 216
Table 31 — Contents of the STATE_CLEAR_DATA Structure 216
Table 32 — Contents of the STATE_RESET_DATA Structure 217
Table 33 — Contents of the PERSISTENT_DATA Structure 218

 © ISO/IEC 2015 – All rights reserved

Figures

Figure 1 — Attestation Hierarchy 30
Figure 2 — Architectural Overview 36
Figure 3 — Command Execution Flow 40
Figure 4 — Random Number Generation 51
Figure 5 — TPM Startup Sequences 60
Figure 6 — On-Demand Self-Test 62
Figure 7 — Failure Mode Behavior 64
Figure 8 — Resuming FUM after _TPM_Init 66
Figure 9 — Field Upgrade Mode 67
Figure 10 — Command Structure 90
Figure 11 — Response Structure 90
Figure 12 — Command/Response Header Structure 91
Figure 13 — Authorization Layout for Command 95
Figure 14 — Authorization Layout for Response 95
Figure 15 — A Policy Evaluation 121
Figure 16 — Two Different Policy Expressions 122
Figure 17 — A Four-Term Policy 122
Figure 18 — Policy with an OR 123
Figure 19 — Policy where only one OR Branch is Evaluated 124
Figure 20 — A 12-input OR Policy 124
Figure 21 — Use of TPM2_PolicyAuthorize() to Avoid PCR Brittleness 131
Figure 22 — Creating a Private Structure 148
Figure 23 — Symmetric Protection of Hierarchy 150
Figure 24 — Duplication Process with Inner and Outer Wrapper 154
Figure 25 — Duplication Process with Outer Wrapper and No Inner Wrapper 155
Figure 26 — Duplication Process with Inner Wrapper and TPM_RH_NULL as NP 156
Figure 27 — Duplication Process with no Inner Wrapper and TPM_RH_NULL as NP 156
Figure 28 — Duplication Groups 158
Figure 29 — Protection Groups 158
Figure 30 — Creating a Identity Structure 164
Figure 31 — Response Code Evaluation 228

© ISO/IEC 2015 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

ISO/IEC 11889-1 was prepared by the Trusted Computing Group (TCG) and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

This second edition cancels and replaces the first edition (ISO/IEC 11889-1:2009), which has been technically revised.

ISO/IEC 11889 consists of the following parts, under the general title *Information technology — Trusted Platform Module Library*:

– *Part 1: Architecture*

– *Part 2: Structures*

– *Part 3: Commands*

– *Part 4: Supporting routines*

This corrected version of ISO/IEC 11889-1:2015 includes the insertion of Figures 7, 9, 15, 18, 21 and 23.

© ISO/IEC 2015 – All rights reserved

Introduction

Collectively the four parts of ISO/IEC 11889 specify the architecture, data structures, command interface and behavior of a Trusted Platform Module (TPM). A properly constructed platform that incorporates a TPM meeting the requirements may enable establishing trust in platform scenarios involving security and privacy.

TPMs require hardware protections to provide three roots of trust: storage, measurement, and reporting. Basing TPM roots of trust in hardware is an improvement over software based solutions whose protections are vulnerable to malicious software. The architecture defines a TPM that is a passive component that receives commands and returns responses. The commands defined have meticulous descriptions and perform primitive actions on data confidential to the TPM. Typical implementations integrate a TPM in the context of a platform like a laptop or a mobile device. By sending commands to a TPM and processing the responses, security benefits accrue for the platform as a whole. Properly constructed platforms with a TPM can provide hardware based roots of trust for storage, measurement and/or reporting.

The root of trust for storage consists primarily of creating, managing and protecting cryptographic keys and other data values. Artifacts protected by or associated with encryption keys, like passwords, certificates or other credentials, can be used for authentication and many other security scenarios. Cryptographic keys can be created with restrictions on their use or management.

EXAMPLE 1 Cryptographic keys can be created that require a password to be used, have a single purpose (e.g. signing), or cannot be exported elsewhere.

Separate from cryptographic key management, the root of trust for storage also allows data values to be stored in a TPM so they are protected from unauthorized modification or can only be changed in defined ways.

EXAMPLE 2 A data value protected by the TPM might be defined so it can be incremented but not decremented.

Typical use cases can include storing security policy information that can only be updated by an authorized entity or incrementing a value to keep track of failed password entry attempts.

The root of trust for measurement is intended to reflect what software is running on a platform in a trusted way. This root consists of the TPM and other components of a platform that start a chain of measurements. Before software runs, its measurement is calculated and stored in the TPM by sending a command. Later software can add more measurements, but cannot erase its own measurement that was recorded before it started running. If the measurement process unconditionally starts when a platform is powered on, the TPM ends up holding an accurate measurement of all the software running on the system from each power-on. Across power cycles, the same software running on a platform results in the same measurements being present in the TPM. Combining the root of trust for measurement with the root of trust for storage, it is possible to create keys or store data using the TPM that only software with specific measurements can use. The result is specific software on a platform can maintain its own keys, confidential data or security policy information that is not accessible and not able to be tampered with by other software that may run on the platform.

The root of trust for reporting helps entities external to a platform establish trust in platform software measurements or cryptographic keys by proving the values exist in a TPM. TPMs have Endorsement Keys that are essentially unique identities for a TPM. Through commands the TPM provides, it is possible to prove keys exist in a TPM with a specific Endorsement Key. Establishing that a key exists in a TPM permits services external to the TPM to trust a key is protected by a TPM's root of trust for storage. Also, if a key is known to be in a TPM, and the key is used to sign measurements of software, evidence of what software is running on a platform can be shared with an external entity in a trusted way. This enhances privacy protections as the need to share the TPM's Endorsement Key identity directly with a remote verifier can be avoided, resulting in a platform that can anonymously prove to a remote verifier what software is running.

Not all privacy and security use cases enabled by this International Standard are relevant for all platforms. For this reason, this International Standard is defined as a generic library of commands, cryptographic algorithms and capabilities for which a subset can be used to meet the needs of a specific

© ISO/IEC 2015 – All rights reserved

platform or implementation, with the flexibility to meet diverse and even contradictory global requirement sets. To address the needs of platform types, market segments, regulations, assurance criteria, certification programs, etc., the structure allows complimentary materials to be developed by interested parties constraining the generic library to address specific requirement sets. To promote product interoperability, implementers are encouraged to consult existing specifications augmenting this International Standard with domain specific considerations for specific applications and platform types. A platform specific context is able to articulate scenarios, functionality, relevant security and privacy goals, implementation considerations and methods of assurance. Because of the generic nature of this International Standard, certification programs are likely to be based on additional specifications that further define platform specific implementations and security characteristics. Having a single TPM library that implementations across different platforms types can use to add security and privacy benefits lowers complexity and permits reuse of software and security analysis across the trusted computing ecosystem.

EXAMPLE 3 Opt-in versus secure by default are examples of contradictory requirements for TPM provisioning that are both supported by this International Standard.

This International Standard is useful for a variety of audiences. Two key audiences are TPM implementers and adopters. Clauses 6 and 7 in this part of ISO/IEC 11889 are useful to orientate audiences to the organization of the four parts and how they are used to build a compliant implementation. Implementers will need to use all four parts to build a compliant implementation. Adopters can benefit from the architectural concepts in this part of ISO/IEC 11889 when developing scenarios and incorporating a TPM into a platform design. ISO/IEC 11889-2 and ISO/IEC 11889-3 are beneficial for adopters to understand the syntax and semantics of using individual TPM commands.

Implementers and adopters of this International Standard need to carefully assess the appropriateness of controllability, security and privacy capabilities and algorithms implemented for satisfying their goals. In assessing algorithms, implementers and adopters should diligently evaluate available information. Solutions involving cryptography are dependent on the solution architecture and on the properties of cryptographic algorithms supported. Over time, cryptographic algorithms can develop deficiencies for reasons like advances in cryptographic techniques or increased computing power. Solutions that support a diversity of algorithms can remain durable when subsets of supported algorithms wane in usefulness. Therefore, implementers intent on providing robust solutions are responsible for evaluating both algorithm appropriateness and diversity.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured the ISO and IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from:

Fujitsu Limited **1-1, Kamikodanaka 4-chrome, Nakahara-ku, Kawasaki-shi, Kanagawa, 211-8588 Japan**
Microsoft Corporation **One Microsoft Way, Redmond, WA 98052**
Enterasys Networks, Inc **50 Minuteman Road, US-Andover, MA 01810**

 © ISO/IEC 2015 – All rights reserved

Lenovo **1009 Think Place, US-Morrisville, NC 27560-8496**
Advanced Micro devices, Inc. - AMD **7171 Southwest Parkway, Mailstop B100.3, US-Austin, Texas 78735**
Hewlett-Packard Company **P.O. Box 10490, US-Palo Alto, CA 94303-0969**
Infineon Technologies AG - Neubiberg **Am Campeon 1-12, DE-85579 Neubiberg**
Sun Microsystems Inc. - Menlo Park, CA **10 Network Circle, UMPK10-146, US-Menlo Park, CA 94025**
IBM Corporation **North Castle Drive, US-Armonk, N.Y. 10504**
Intel Corporation **5200 Elam Young Parkway, US-Hillsboro, OR 97123**

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

© ISO/IEC 2015 – All rights reserved

Information technology — Trusted Platform Module Library — Part 1: Architecture

1 Scope

This part of ISO/IEC 11889 defines the architectural elements of the Trusted Platform Module (TPM), a device which enables trust in computing platforms in general. Some TPM concepts are explained adequately in the context of the TPM itself. Other TPM concepts are explained in the context of how a TPM helps establish trust in a computing platform. When describing how a TPM helps establish trust in a computing platform, this part of ISO/IEC 11889 provides some guidance for platform requirements. However, the scope of ISO/IEC 11889 is limited to TPM requirements.

This part of ISO/IEC 11889 illustrates TPM security and privacy techniques in the context of a platform through the use of cryptography. It includes definitions of how different cryptographic techniques are implemented by a TPM. The scope of ISO/IEC 11889 does not include cryptographic analysis or guidance about the applicability of different algorithms for specific uses cases.

TPM requirements in this part of ISO/IEC 11889 are general, covering concepts like integrity protection, isolation and confidentially. Defining a specific strength of function or assurance level is out of scope for ISO/IEC 11889. This approach limits the guarantees provided by ISO/IEC 11889 itself, but it does allow the TPM architectural elements defined to be adapted to meet diverse implementation and platform specific needs.

© ISO/IEC 2015 – All rights reserved

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- ISO/IEC 9797-2, *Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function*
- ISO/IEC 10116:2006, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*
- ISO/IEC 11889-2, *Information technology — Trusted Platform Module Library — Part 2: Structures*
- ISO/IEC 11889-3, *Information technology — Trusted Platform Module Library — Part 3: Commands*
- ISO/IEC 11889-4, *Information technology — Trusted Platform Module Library — Part 4: Supporting routines*
- TCG Algorithm Registry, available at <http://www.trustedcomputinggroup.org/resources/tcg_algorithm_registry>

 © ISO/IEC 2015 – All rights reserved

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
ancestor

<object loaded in a Trusted Platform Module> Storage Key that needs to have been loaded prior to loading an object

3.2
authValue

octet string containing a value that is used for access authorization

Note 1 to entry: The value is used as a password or to derive a key for an Hash Message Authentication Code calculation.

3.3
authPolicy

digest value produced by an execution of policy commands and used for access authorization

3.4
bound

authValue of the Object is not included in the Hash Message Authentication Code authorization for the authorization session

3.5
canonical form

data structure in the format used for transport to and from the Trusted Platform Module

Note 1 to entry: See clause 3.25.

3.6
CLEAR

bit with a value of zero (0), or the action of causing a bit to have a value of zero (0)

3.7
command

discrete Trusted Platform Module function that is exposed externally and recognizable by a Trusted Platform Module's command processor as well as the values sent to the Trusted Platform Module to indicate the operation to be performed

3.8
commandCode

numeric identifier of the operation to be performed by a Trusted Platform Module

© ISO/IEC 2015 – All rights reserved

3.9
context

collection of data that provides qualifying information about a data object to differentiate it from others of the same type or to differentiate one version of a data object from another

3.10
cpHash

hash of the command code, Object names, and parameters of a command

3.11
descendant

<Storage Key> Object whose loading is conditional on the Storage Key having been previously loaded

3.12
digest

result of a hash operation

3.13
duplicate

allowing a Protected Object created by a Trusted Platform Module to be used on a different Trusted Platform Module

3.14
ECDH

Diffie-Hellman secure secret sharing process using elliptic curve operations

3.15
Ephemeral Key

key created as part of a protocol that is not used again after the protocol is complete

3.16
EmptyAuth

Empty Buffer used as an authorization value

3.17
Empty Buffer

sized array with no data, indicated by a size field of zero followed by an array containing no elements

3.18
Empty Point

Elliptic curve cryptography point with Empty Buffers for both the x and y coordinates

 © ISO/IEC 2015 – All rights reserved

3.19
Empty Policy

Empty Buffer used when a policy value is required

Note 1 to entry: As a *policyValue*, an Empty Buffer will satisfy no policy

Note 2 to entry: No policy can be satisfied by an Empty Policy because an Empty Policy has zero length but a *policyDigest* is the size of a hash digest and a digest is never zero length.

3.20
Endorsement Authorization

Authorization using either *endorsementAuth* or *endorsementPolicy*

3.21
Extend
Extended

operation that replaces the current value of a digest with the hash of a buffer constructed by concatenating new data to the current value of the digest

Note 1 to entry: See clause 11.4.7

Note 2 to entry: The new data concatenated is normally a digest.

3.22
External Object

Object that can be loaded into a Trusted Platform Module without being a member of a specific hierarchy

3.23
Failure mode

mode in which the Trusted Platform Module returns TPM_RC_FAILURE in response to all commands except TPM2_GetTestResult() or TPM2_GetCapability()

3.24
import

operation that allows a Protected Object not created by a Trusted Platform Module to be incorporated into a hierarchy of the Trusted Platform Module

3.25
internal form

data structure using a layout that is specific to an implementation that might or might not be the same as the canonical form

3.26
Lockout Authorization

authorization using either *lockoutAuth* or *lockoutPolicy*

3.27
non-volatile

data that is retained even when power is removed

© ISO/IEC 2015 – All rights reserved

3.28
NULL

<pointer> context-sensitive value that is a system-defined value indicating that the pointer does not reference data

3.29
NULL

<structure identified by an algorithm identifier> TPM_ALG_NULL value indicating that no additional data is present

3.30
NULL Password
NULL Auth

authorization where the authorization value is the Empty Buffer resulting in an authorization that is a fixed sequence of 9 octets of 40 00 00 09 00 00 00 00 00_{16}

3.31
NULL Signature

signature with the TPM_ALG_NULL signature scheme that contains no data

3.32
NULL-terminated

sequence of non-zero values followed by a value containing zero

3.33
NULL Ticket

ticket structure with *tag* set to a value that is correct for the context, *hierarchy* is TPM_RH_NULL, and *digest* is an Empty Buffer

3.34
Object

key or data that has a public portion and, optionally, a sensitive portion and which is a member of a hierarchy

3.35
octet

a group of eight adjacent bits

Note 1 to entry: On most modern computers, this is the smallest addressable unit of data.

[SOURCE: ISO/IEEE 11073-30300:2004(en), 3.1.26]

3.36
orderly shutdown

when the Trusted Platform Module has completed TPM2_Shutdown() before power to the Trusted Platform Module is removed or _TPM_Init is asserted

 © ISO/IEC 2015 – All rights reserved

3.37
ordinary key

key produced with a seed taken from the Trusted Platform Module random number generator

3.38
Owner Authorization

authorization using either *ownerAuth* or *ownerPolicy*

3.39
PCR

one or more platform configuration registers each containing a digest

3.40
PCR.alg

hash algorithm associated with a specific platform configuration register

3.41
PCR bank

collection of platform configuration registers identified by a hash algorithm, with each platform configuration register in the bank containing a digest computed using the bank identifier's hash algorithm

3.42
PCR.digest

digest value associated with a specific platform configuration register

3.43
Permanent Entity

Trusted Platform Module resource with an architecturally defined handle that does not change

Note 1 to entry: The value of a Permanent Entity can change

3.44
Persistent Object

Trusted Platform Module resource created by a Protected Capability that persists in Trusted Platform Module memory across power cycles and Trusted Platform Module resets

3.45
Platform Authorization

authorization using either *platformAuth* or *platformPolicy*

3.46
policyDigest

digest uniquely representing an ordered set of policy commands and operands used to determine if a policy authorizing an action has been satisfied

© ISO/IEC 2015 – All rights reserved

3.47
policySession→cpHash

policy session context value that, if not the Empty Buffer, is the *cpHash* value that the authorized command needs to have for the authorization to be valid

3.48
platform firmware

code added to the platform by its manufacturer that is needed for booting and proper platform operation

Note 1 to entry: Commonly, but not exclusively, referred to as BIOS or UEFI or SMM code

3.49
Primary Key

key derived from a Primary Seed that is associated with the hierarchy of the Primary Seed

3.50
Primary Object

Primary Key or a data blob with a sensitive area that is encrypted using a symmetric key derived from the public area of the object and a Primary Seed

3.51
private area

encrypted and integrity protected blob that contains the sensitive area of an object

3.52
Primary Seed

large random value contained within a Trusted Platform Module from which Primary Keys and Primary Objects are derived

3.53
Protected Capability

operation performed by the Trusted Platform Module on data in a Shielded Location in response to a command sent to the Trusted Platform Module

3.54
Protected Object

Object with an encrypted sensitive portion, the sensitive portion of which the Trusted Platform Module will only decrypt when it is in a Shielded Location

3.55
Random Access Memory
RAM

memory that can be accessed in any order and which has no endurance limitations

 © ISO/IEC 2015 – All rights reserved

3.56
reset interval

period between two successive Trusted Platform Module Resets and the interval during which the *resetCount* is not changed

3.57
response

values returned by the Trusted Platform Module when it completes processing of a command

3.58
Resume PCR

platform configuration register with a value that is preserved over a Trusted Platform Module Resume sequence

3.59
Root of Trust

component that needs to always behave in the expected manner because its misbehavior cannot be detected

Note 1 to entry: The complete set of Roots of Trust has at least the minimum set of functions to enable a description of the platform characteristics that affect the trustworthiness of the platform.

3.60
rpHash

hash of the response code and the parameters of a response

3.61
Sealed Data Object

encrypted, user-defined, data blob that is associated with a hierarchy and loaded using TPM2_Load() or TPM2_CreatePrimary()

3.62
sensitive area

contains the confidential or secret parts of an object that need to be encrypted and integrity protected when not in a Shielded Location on a Trusted Platform Module

3.63
sequence object

transient data structure used to hold hash state that has a handle and can be context swapped

Note 1 to entry: See clause 30

3.64
session

transient Trusted Platform Module structure that maintains the state associated with a sequence of authorizations or an audit digest

© ISO/IEC 2015 – All rights reserved

3.65
SET

bit with a value of one (1), or the action of causing a bit to have a value of one (1)

3.66
Shielded Location

location on a Trusted Platform Module that contains data that is shielded from access by any entity other than the Trusted Platform Module and which can be operated on only by a Protected Capability

3.67
Shutdown(CLEAR)

abbreviated form of the command TPM2_Shutdown() with the *startupType* parameter set to TPM_SU_CLEAR

3.68
Shutdown(STATE)

abbreviated form of the command TPM2_Shutdown() with the *startupType* parameter set to TPM_SU_STATE

3.69
sizeof(x)

operator that returns the number of octets in the operand 'x'

3.70
Startup(CLEAR)

abbreviated form of the command TPM2_Startup() with the *startupType* parameter set to TPM_SU_CLEAR

3.71
Startup(STATE)

abbreviated form of the command TPM2_Startup with the *startupType* parameter set to TPM_SU_STATE

3.72
Storage Key

key that can have descendant keys

3.73
Temporary Object

object that become unusable after a Trusted Platform Module Reset and that cannot be converted into a Persistent Object

3.74
temporary resource

data object created during the execution of a command that does not persist in TPM memory after the command completes

 © ISO/IEC 2015 – All rights reserved

3.75
TPM_GENERATED_VALUE

32-bit number FF 54 43 47_{16} used to tag structures that are generated by a Trusted Platform Module

3.76
TPM Reset

resetting of all Trusted Platform Module internal state to default values due to Startup(CLEAR)

3.77
TPM Resource Manager
TRM

software executing on a system with a Trusted Platform Module that ensures that the resources necessary to execute Trusted Platform Module commands are present in the Trusted Platform Module

3.78
TPM Restart

Startup(CLEAR) that initializes all platform configuration registers but preserves most other Trusted Platform Module state from the previous Shutdown(STATE)

3.79
TPM Resume

Startup(STATE) that initializes some platform configuration registers but preserves most Trusted Platform Module state from the previous Shutdown(STATE)

3.80
transient resource

object or session that can be explicitly loaded and unloaded from Trusted Platform Module memory by the Trusted Platform Module Resource Manager and is cleared from Trusted Platform Module memory when the Trusted Platform Module is initialized (TPM2_Startup())

3.81
Trusted Platform Module
TPM

implementation of ISO/IEC 11889

3.82
user-installable software

any software that can be installed on a platform other than platform firmware

3.83
volatile data

data that is lost when power is removed

3.84
Zero Digest

non-zero-length digest with all octets set to zero

© ISO/IEC 2015 – All rights reserved