INTERNATIONAL STANDARD

ISO/IEC 11889-1

Second edition
2015-12-15

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)

Documents
Document Sharing Hub
© ISO/IEC 2015

Full document details: ISO_IEC 11889-1_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

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 11889-1_2015

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

AVA Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

11.2 TPM Command Processing Overview ... 33
11.3 I/O Buffer ... 37
11.4 Cryptography Subsystem ... 37
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

AVA Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

13.8 TPM Ownership ... 70
13.8.1 Taking Ownership ... 70
13.8.2 Releasing Ownership ... 70
14 Primary Seeds ... 72
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

AVA Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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
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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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
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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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
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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

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

AVA Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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
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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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
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 ... 255

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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
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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

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*

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

AVA Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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**

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

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.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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>

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

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

Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

4 Symbols and Abbreviated Terms

4.1 Symbols

For the purposes of this document, the following symbol definitions apply unless the text is in the `Courier` font.

A || B	concatenation of B to A
|x|	the length of x in bits
⌈x⌉	the smallest integer not less than x
⌊x⌋	the largest integer not greater than x
A ≔ B	assignment of the results of the expression on the right (B) to the parameter on the left
A = B	equivalence (A is the same as B)
{ A }	an optional element
A ⊕ B	bitwise exclusive OR of elements
A & B	logical AND of elements
A | B	the logical OR of elements
{A | B}	selection of elements
{A : B}	an inclusive range of elements between A and B
<A, B, … >	an ordered list of elements (a tuple)
0…0	a context-sensitive number of octets of zero
F()	denotes a function **F**
F(*p* == *x*)	denotes a function or TPM command **F** with parameter *p* set to value *x*
H()	denotes the hash function
[*n*]*P*	multiplication of point *P* by the integer value *n*
A • B	multiplication of two integer values A and B
A→B	denotes a reference to element B within structure A
A mod B	A modulus B

Text in the **`Courier`** font indicates code written according to the C language standard.

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

4.2 Abbreviations

For the purposes of ISO/IEC 11889, the following abbreviations apply.

Abbreviation	Description
TPM	Prefix for an indication passed from the system interface of the TPM to a Protected Capability defined in ISO/IEC 11889
AK	Attestation Key
BIOS	Basic Input/Output System
CA	Certificate Authority
CFB	Cipher Feedback mode
CPU	Central Processing Unit
CRTM	Core Root of Trust for Measurement
CTR	Counter mode
D-RTM	dynamic RTM
DA	dictionary attack
DoS	Denial of Service
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
EA	Enhanced Authorization
EAL	evaluated assurance level
ECC	Elliptic Curve Cryptography
ECDAA	ECC-based Direct Anonymous Attestation
ECDH	Elliptic Curve Diffie-Hellman
EK	Endorsement Key
EPS	Endorsement Primary Seed
FIPS	Federal Information Processing Standard
FUM	Field Upgrade mode
GPIO	General Purpose I/O
HMAC	Hash Message Authentication Code
I/O	Input/Output
IV	Initialization Vector
KDF	key derivation function
LPC	Low Pin Count
LSb	Least Significant bit
LSO	Least Significant Octet
MSb	Most Significant bit
MSO	Most Significant Octet
NIST	National Institute of Standards and Technology
NP	new parent

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

Abbreviation	Description
NV	non-volatile
NVRAM	Non-Volatile Random Access Memory
OAEP	Optimal Asymmetric Encryption Padding
OEM	Original Equipment Manufacturer
OIAP	Object-Independent Authorization Protocol
OID	Object Identifier in ASN.1 format
OSAP	Object-Specific Authorization Protocol
PCR	platform configuration register(s)
POST	Power on Self-Test
PP	Physical Presence
PPS	Platform Primary Seed
PRF	pseudo-random function
PRNG	pseudo-random number generator
PSS	Probabilistic Signature Scheme
QN	Qualified Name
RNG	random number generator
RSA	Rivest, Shamir and Adleman
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
S-RTM	static RTM
SHA	Secure Hash Algorithm
SMM	System Management mode
SPS	Storage Primary Seed
SRK	Storage Root Key
TBB	trusted building block
TCB	trusted computing base
TCG	Trusted Computing Group
TPM	Trusted Platform Module
TPM2_	Prefix for a command defined in ISO/IEC 11889
TSS	TCG Software Stack
UEFI	Unified Extensible Firmware Interface

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

5 Conventions

5.1 Bit and Octet Numbering and Order

An integer value is considered to be an array of one or more octets. The octet at offset zero within the array is the most significant octet (MSO) of the integer. Bit number 0 of that integer is its least significant bit and the is the least significant bit in the last octet in the array.

EXAMPLE A 32-bit integer is an array of four octets; the MSO is at offset [0], and the most significant bit is bit number 31. Bit zero of this 32-bit integer is the least significant bit in the octet at offset [3] in the array.

NOTE 1 Array indexing is zero-based.

NOTE 2 This definition does not match the “network bit order” used in many IETF documents, such as RFC 4034. In those documents, the most significant bit of a datum has the lowest bit number. It is conventional practice to send that bit first when using a serial network protocol, and the bits are numbered in the order in which they are sent. ISO/IEC 11889 numbers bits according to the power of two to which they correspond within a datum. This numbering corresponds to the normal convention for bit numbering in hardware registers that hold integer values rather than fixed-point numbers.

The first listed member of a structure is at the lowest offset within the structure and the last listed member is at the highest offset within the structure.

For a character string (letters delimited by “”), the first character of the string contains the MSO.

5.2 Sized Buffer References

ISO/IEC 11889 makes extensive use of a data structure called a *sized buffer*. A sized buffer has a size field followed by an array of octets equal in number to the value in the size field.

The structure will have an identifying name. When ISO/IEC 11889 references the size field of the structure, the structure name is followed by “.size” (a period followed by the word “size”). When ISO/IEC 11889 references the octet array of the structure, the structure name is followed by “.buffer” (a period followed by the word “buffer”).

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

5.3 Numbers

Numbers are decimal unless a different radix is indicated.

Unless the number appears in a table intended to be machine readable, the radix is a subscript following the digits of the number. Only radix values of 2 and 16 are used in ISO/IEC 11889.

Radix 16 (hexadecimal) numbers have a space separator between groups of two hexadecimal digits.

EXAMPLE 1 $40\ FF\ 12\ 34_{16}$

Radix 2 (binary) numbers use a space separator between groups of four binary digits.

EXAMPLE 2 $0100\ 1110\ 0001_2$

For numbers using a binary radix, the number of digits indicates the number of bits in the representation.

EXAMPLE 3 20_{16} is a hexadecimal number that contains exactly 8 bits and has a decimal value of 32.

EXAMPLE 4 $10\ 0000_2$ is a binary number that contains exactly 6 bits and has a decimal value of 32.

EXAMPLE 5 $0\ 20_{16}$ is a hexadecimal number that contains exactly 12 bits and has a decimal value of 32.

A number in a machine-readable table may use the “0x” prefix to denote a base 16 number. In this format, the number of digits is not always indicative of the number of bits in the representation.

EXAMPLE 6 0x20 is a hexadecimal number with a value of 32, and the number of bits is determined by the context.

5.4 KDF Label Parameters

Specific strings representing octets of data are used normatively in this International Standard as the label parameter for key derivation functions. The octets are represented as quoted ASCII characters and exhaustively listed in Table 1 with their length and values to disambiguate them from regular words used in quotes.

Table 1 — KDF Label Parameters

Label	***Sequence size***	***Octet data***
“ATH”	4	$41\ 54\ 48\ 00_{16}$
“CFB”	4	$43\ 46\ 42\ 00_{16}$
“DUPLICATE”	10	$44\ 55\ 50\ 4C\ 49\ 43\ 41\ 54\ 45\ 00_{16}$
“IDENTITY”	9	$49\ 44\ 45\ 4E\ 54\ 49\ 54\ 59\ 00_{16}$
“INTEGRITY”	10	$49\ 4E\ 54\ 45\ 47\ 52\ 49\ 54\ 59\ 00_{16}$
“OBFUSCATE”	10	$4F\ 42\ 46\ 55\ 53\ 43\ 41\ 54\ 45\ 00_{16}$
“SECRET”	7	$53\ 45\ 43\ 52\ 45\ 54\ 00_{16}$
“STORAGE”	8	$53\ 54\ 4F\ 52\ 41\ 47\ 45\ 00_{16}$
“XOR”	4	$58\ 4F\ 52\ 00_{16}$
NOTE See clause 11.4.9.3 for justification for the terminating octet of 00_{16}.		

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

6 ISO/IEC 11889 Organization

ISO/IEC 11889 defines the Trusted Platform Module (TPM), a device that enables trust in computing platforms in general. It is broken into parts to make the role of each part clear. All parts are required in order to constitute a complete standard.

For a complete definition of all requirements necessary to build a TPM, the designer will need to use the appropriate platform-specific specification to understand all of the requirements for a TPM in a specific application or make appropriate choices as an implementer.

Those wishing to create a TPM need to be aware that ISO/IEC 11889 does not provide a complete picture of the options and commands necessary to implement a TPM. To implement a TPM the designer needs to refer to the relevant platform-specific specification to understand the options and settings required for a TPM in a specific type of platform or make appropriate choices as an implementer.

EXAMPLE The number of platform configuration registers and their attributes are not defined in ISO/IEC 11889. Those values would be specified by a platform specific specification or alternatively determined by an implementer.

ISO/IEC 11889 contains four parts, as follows:

Part 1: Architecture

This part of ISO/IEC 11889 contains a narrative description of the properties, functions, and methods of a TPM. Unless otherwise noted, this narrative description is *informative*. This part of ISO/IEC 11889 contains descriptions of some of the data manipulation routines that are used by ISO/IEC 11889. The normative behavior for these routines is in C code in ISO/IEC 11889-3 and ISO/IEC 11889-4. Algorithms and processes specified in this part of ISO/IEC 11889 may be made normative by reference from ISO/IEC 11889-2, ISO/IEC 11889-3 or ISO/IEC 11889-4.

Part 2: Structures

ISO/IEC 11889-2 contains a *normative* description of the constants, data types, structures, and unions for the TPM interface. Unless otherwise noted: (1) all tables and C code in ISO/IEC 11889-2 are normative, and (2) normative content in ISO/IEC 11889-2 takes precedence over any other part of ISO/IEC 11889.

Part 3: Commands

ISO/IEC 11889-3 contains: (1) a *normative* description of commands, (2) tables describing the command and response formats, and (3) C code that illustrates the actions performed by a TPM. Within ISO/IEC 11889-3, command and response tables have the highest precedence, followed by the C code, followed by the narrative description of the command. ISO/IEC 11889-3 is subordinate to ISO/IEC 11889-2.

A TPM need not be implemented using the C code in ISO/IEC 11889-3. However, any implementation should provide equivalent or, in most cases, identical results as observed at the TPM interface or demonstrated through evaluation.

Part 4: Supporting routines

ISO/IEC 11889-4 presents C code that describes the algorithms and methods used by the command code in ISO/IEC 11889-3. The code in ISO/IEC 11889-4 augments ISO/IEC 11889-2 and ISO/IEC 11889-3 to provide a complete description of a TPM, including the supporting framework for the code that performs the command actions.

Any ISO/IEC 11889-4 code may be replaced by code that provides similar results when interfacing to the action code in ISO/IEC 11889-3. The behavior of ISO/IEC 11889-4 code not included in an annex is

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

normative, as observed at the interfaces with ISO/IEC 11889-3 code. Code in an annex is provided for completeness, that is, to allow a full implementation of ISO/IEC 11889.

NOTE ISO/IEC 11889 does not provide code for lower-level cryptographic algorithms and use of external libraries is required for a complete implementation.

Extensive modification of the code provided in ISO/IEC 11889-4 annexes is expected for any TPM implementation. Modifications are required in order to interface the TPM code with actual TPM hardware rather than the simulation framework provided. In addition, modifications of the code in ISO/IEC 11889-4 annexes would be necessary in order to meet the needs of applicable evaluation regimes.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

7 Compliance

Unless the ISO/IEC 11889-3 general description of a command indicates that the command is mandatory, a compliant TPM need not implement the command. However, if implemented, the command is required to have the behavior defined in ISO/IEC 11889- 3. A platform-specific specification will indicate the commands from ISO/IEC 11889 that are required to be implemented in order to be compliant with that platform-specific specification.

The code in ISO/IEC 11889 is the reference implementation that describes required TPM behavior as observed from the TPM interface. The C-code may be reorganized or rewritten in any desired implementation language and remain compatible with ISO/IEC 11889 as long as the observable behavior is equivalent.

Even though the code in the reference implementation has undergone extensive testing, it is likely that some errors exist and one or more of those errors could lead to a TPM failure or exploit. Regardless of any other statement about normative behavior, one should not assume that a TPM exploit or failure is an intended behavior. It is not necessary to reproduce such a behavior in order to be compliant with ISO/IEC 11889.

NOTE 1 Please report bugs in the reference code to the TCG (admin@trustedcomputinggroup.org) so that the reference code may be brought into compliance with ISO/IEC 11889.

The response codes in ISO/IEC 11889 are normative. An implementation performing a check prescribed by ISO/IEC 11889 is required to return the indicated error if the check fails. The order in which checks are performed is not normative. This means that a command with multiple errors could return different response codes on different TPMs. However, the response code returned is required to be the normative response code used to indicate the specific failure.

Capacities and algorithms of a TPM implementation may vary from the reference implementation; in this case, the same error would not occur in the same situation. However, these differences should not cause a different response code to be returned when the nature of the error is the same as in the reference implementation. ISO/IEC 11889-4 contains major subsystems that may change for each instance of a TPM. When the subsystem is rewritten, an equivalent interface should be provided, and the errors returned are required to match those of the reference implementation.

EXAMPLE 1 An example where a TPM implementation might not return the same error as the reference implementation is one with more memory which might be able to satisfy a request where the reference implementation would have returned an error.

EXAMPLE 2 Examples of TPM capacities are the amount of nonvolatile storage space, the number of keys that maybe simultaneously loaded, the number of PCR, the number of PCR banks supported, etc.

EXAMPLE 3 The NV subsystem of the reference implementation is not representative of the actual implementation of most physical NV implementations but is a crude analog.

NOTE 2 A constraint on the design of the TPM was the process of compliance-testing of different TPM implementations. If a TPM implementation has modularity similar to the reference implementation, then TPM tests that assume a modular design will be able to produce reliable test results on each TPM implementation.

The reference implementation uses static and stack-based allocation of resources and does not do allocations on a heap. However, a TPM implementation may use heap-based memory management in which case some error conditions and codes will differ. These differences are limited, and the allowed response codes and error conditions are defined in 39.3.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

8 Changes from Previous Versions

This version of ISO/IEC 11889 introduces these additional features to the TPM family:

- Definition of an interface that allows variability of underlying cryptographic algorithms – ISO/IEC 11889 (first edition) is constrained by its data structures to using RSA and SHA1. The ISO/IEC 11889 structure and interface defines support for a wide range of hash and asymmetric algorithms along with limited support for use of various block, symmetric ciphers. Of particular note is the addition of support for the elliptic curve (ECC) family of asymmetric algorithms.
- Unification of authorization methods – ISO/IEC 11889 (first edition) has different schemes to authorize the use, delegated use, and migration of objects. ISO/IEC 11889 provides a uniform framework for using authorization capabilities so they may be combined in unique ways to provide more flexibility.
- Expansion of authorization methods – ISO/IEC 11889 allows authorization with clear-text passwords and Hash Message Authentication Code (HMAC). It also allows construction of an arbitrarily complex authorization policy for an object using multiple authorization qualifiers.
- Dedicated BIOS support – ISO/IEC 11889 adds a Storage hierarchy controlled by platform firmware, letting the OEM benefit from the cryptographic capabilities of the TPM regardless of the support provided to the OS.
- Simplified control model – ISO/IEC 11889 needs no special provisioning process to be useful to applications. Although objects on which the TPM operates may have limitations, all commands are available all the time. This lets application developers rely on TPM capabilities being available whenever a TPM is present.

A TPM compatible with ISO/IEC 11889 need not be compatible with ISO/IEC 11889 (first edition).

ISO/IEC 11889 defines the operations a TPM performs and the structures used for communication between the TPM and the host system. It does not define an electrical interface to the TPM, nor does it specify which subset of ISO/IEC 11889 commands and resources are required for a specific platform. Please refer to platform-specific specifications for this information.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

9 Trusted Platforms

9.1 Trust

In the context of Trusted Computing Group (TCG) specifications, “trust” is meant to convey an expectation of behavior. However, predictable behavior does not necessarily constitute behavior that is worthy of trust.

EXAMPLE We expect that a bank will behave like a bank, and we expect that a thief will behave like a thief.

In order to determine the expected behavior of a platform, it is necessary to determine its identity as it relates to the platform behavior. Physically different platforms may have identical behavior. If they are constructed of components (hardware and software) that have identical behavior, then their trust properties should be the same.

The TCG defines schemes for establishing trust in a platform that are based on identifying its hardware and software components. The Trusted Platform Module (TPM) provides methods for collecting and reporting these identities. A TPM used in a computer system reports on the hardware and software in a way that allows determination of expected behavior and, from that expectation, establishment of trust.

9.2 Trust Concepts

9.2.1 Trusted Building Block

A trusted building block (TBB) is a component or collection of components required to instantiate a Root of Trust. Typically platform-specific, a TBB is part of a Root of Trust that does not have Shielded Locations.

EXAMPLE One example of a TBB is the combination of the CRTM, the connection between CRTM storage and a motherboard, the path between CRTM storage and the CPU, the connection between the TPM and a motherboard, and the path between the CPU and the TPM. This combination comprises the Root of Trust for Reporting (RTR).

A TBB is a component that is expected to behave in a way that does not compromise the goals of trusted platforms.

9.2.2 Trusted Computing Base

A trusted computing base (TCB) is the collection of system resources (hardware and software) that is responsible for maintaining the security policy of the system. An important attribute of a TCB is that it be able to prevent itself from being compromised by any hardware or software that is not part of the TCB.

The TPM is not the trusted computing base of a system. Rather, a TPM is a component that allows an independent entity to determine if the TCB has been compromised. In some uses, the TPM can help prevent the system from starting if the TCB cannot be properly instantiated.

9.2.3 Trust Boundaries

The combination of TBB and Roots of Trust form a trust boundary, within which measurement, storage, and reporting may be accomplished for a minimal configuration. In systems that are more complex, it may be necessary for the CRTM to establish trust in other code, by making measurements of that other code and recording the measurement in a PCR. If the CRTM transfers control to that other code regardless of the measurement, then the trust boundary is expanded. If the CRTM will not run that code unless its

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

measurement is the expected value, the trust boundary remains the same because the measured code is an expected extension of the CRTM.

9.2.4 Transitive Trust

Transitive trust is a process whereby the Roots of Trust establish the trustworthiness of an executable function, and trust in that function is then used to establish the trustworthiness of the next executable function.

Transitive trust may be accomplished either by: (1) knowing that a function enforces a trust policy before it allows a subsequent function to take control of the TCB, or (2) using measurements of subsequent functions so that an independent evaluation may establish the trust. The TPM may support either of these methods.

9.2.5 Trust Authority

When the RTM begins to execute the CRTM, the entity that may vouch for the correctness of the TBB is the entity that created the TBB. For typical systems, this is the platform manufacturer. In other words, the manufacturer is the authority on what constitutes a valid TBB, and its reputation is what allows someone to trust a given TBB.

As the system transitions to code outside the CRTM, the transitive trust chain is maintained by measurement of that code. If execution of that code is conditional on its measurement, then the authority for that code remains unchanged. That is, if the platform manufacturer's CRTM does not run code outside the CRTM unless that code has a specific measured value, then the platform manufacturer remains the trust authority regardless of who provided that code.

In modern architectures, where firmware and software components come from many different suppliers, it is often not feasible for platform manufacturers to know the signers of all code that runs on a platform. Therefore, they may not remain the authority on platform state for very long. The measurements recorded in the RTS then determine the chain of authority for the current system state.

Two different methods allow evaluation of the trust authority for a platform.

1) Code is measured (hashed), and its value is recorded in the RTS. If the code is run regardless of its measurement, then the authority for the trust is the digest of the code reported by the RTR. That is, the measurements speak for themselves, and the verifier needs either to have knowledge of the measurements that constitute trustworthy code or knowledge of the measurements that indicates malicious or vulnerable code.
2) Code is signed so that the identity of the authority for the code is known. If this identity is recorded in the RTS, the evaluation can be changed. Instead of being based on knowing the digest of the code, it can be based on identities of the signers of the code.

Because trusted sources of code may sometimes produce code with security vulnerabilities, support for revocation is often required. To allow revocation of specific code modules, it is often necessary to use a hybrid solution where both authorities and details are recorded. This simplifies the process of determining whether a module from a specific vendor has been revoked.

NOTE If the code is measured (hashed) and not signed, it is harder to know if a specific measurement is valid unless there is a centralized database of all known digests of revoked code. When the identity of the authority is known, one can contact the vendor to determine if it has revoked code with a given hash.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

9.3 Trusted Platform Module

A TPM is a system component that has state that is separate from the system on which it reports (the host system). The only interaction between the TPM and the host system is through the interface defined in ISO/IEC 11889.

TPMs are implemented on physical resources, either directly or indirectly. A TPM may be constructed using physical resources that are permanently and exclusively dedicated to the TPM, and/or using physical resources that are temporarily assigned to the TPM. All of a TPM's physical resources may be located within the same physical boundary, or different physical resources may be within different physical boundaries.

Some TPMs are implemented as single-chip components that are attached to systems (typically, a PC) using a low-performance interface. The TPM component has a processor, RAM, ROM, and Flash memory. The only interaction with the TPM is through the LPC. The host system cannot directly change the values in TPM memory other than through the I/O buffer that is part of the interface.

EXAMPLE An example of low performance interface is the Low Pin Count or LPC.

Another reasonable implementation of a TPM is to have the code run on the host processor while the processor is in a special execution mode. For these TPMs, parts of system memory are partitioned by hardware so that the memory used by the TPM is not accessible by the host processor unless it is in this special mode. Further, when the host processor switches modes, it always begins execution at specific entry points. This version of a TPM would have many of the same attributes as the stand-alone component in that the only way for the host to cause the TPM to modify its internal state is with well-defined interfaces. There are several different schemes for achieving this mode switching including System Management Mode, Trust Zone™, and processor virtualization.

Definition of the interaction between the host and the TPM is the primary objective of ISO/IEC 11889. Prescribed commands instruct the TPM to perform prescribed actions on data held with the TPM. A primary purpose of these commands is to allow determination of the trust state of a platform. The ability of a TPM to accomplish its objective depends on the proper implementation of Roots of Trust.

9.4 Roots of Trust

9.4.1 Introduction

TCG-defined methods rely on Roots of Trust. These are system elements that must be trusted because misbehavior is not detectable. The set of roots required by the TCG provides the minimum functionality necessary to describe characteristics that affect a platform's trustworthiness.

While it is not possible to determine if a Root of Trust is behaving properly, it is possible to know how roots are implemented. Manufacturers can provide assurances that the root has been implemented in a way that renders it trustworthy.

EXAMPLE A certificate may identify the manufacturer and evaluated assurance level (EAL) of a TPM.

A certification can provide confidence in the Roots of Trust implemented in the TPM. In addition, a certificate from a platform manufacturer may provide assurance that the TPM was properly installed on a machine that is compliant with TCG specifications so that the Root of Trust provided by the platform may be trusted. See 9.5.2 for more information on certification.

The TCG requires three Roots of Trust in a trusted platform:

- Root of Trust for Measurement (RTM),

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

- Root of Trust for Storage (RTS), and
- Root of Trust for Reporting (RTR).

Trust in the Roots of Trust can be achieved through a variety of means including technical evaluation by competent experts.

9.4.2 Root of Trust for Measurement (RTM)

The RTM sends integrity-relevant information (measurements) to the RTS. Typically, the RTM is the CPU controlled by the Core Root of Trust for Measurement (CRTM). The CRTM is the first set of instructions executed when a new chain of trust is established. When a system is reset, the CPU begins executing the CRTM. The CRTM then sends values that indicate its identity to the RTS. This establishes the starting point for a chain of trust. See 9.5.5 for a more detailed description of integrity measurement.

9.4.3 Root of Trust for Storage (RTS)

The TPM memory is shielded from access by any entity other than the TPM. Because the TPM can be trusted to prevent inappropriate access to its memory, the TPM can act as an RTS.

Some of the information in TPM memory locations is not sensitive and the TPM does not protect it from disclosure. Other information is sensitive and the TPM does not allow access to the information without proper authority.

EXAMPLE 1 An example of non-sensitive data is the current contents of a platform configuration register (PCR) containing a digest.

EXAMPLE 2 An example of sensitive data in a Shielded Location is the private part of an asymmetric key.

Sometimes, the TPM uses the contents of one Shielded Location to gate access to another Shielded Location.

EXAMPLE 3 Access to (use of) a private key for signing might be conditioned on PCR having specific values.

9.4.4 Root of Trust for Reporting (RTR)

9.4.4.1 Description

The RTR reports on the contents of the RTS. An RTR report is typically a digitally signed digest of the contents of selected values within a TPM.

NOTE Not all Shielded Locations are directly accessible.

EXAMPLE 1 The values of the private part of keys and authorizations are in Shielded Locations on which the TPM will not report.The values on which the RTR reports typically are

- evidence of a platform configuration in PCR,
- audit logs, and
- key properties.

EXAMPLE 2 An example of a command providing evidence of platform configuration is TPM2_Quote().

EXAMPLE 3 An example of a command providing audit logs is TPM2_GetCommandAuditDIgest ().

EXAMPLE 4 An example of a command providing key properties TPM2_Certify().

Documents
Document Sharing Hub

 © ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

The interaction between the RTR and RTS is critical. The design and implementation of this interaction should mitigate tampering that would prevent accurate reporting by the RTR. An instantiation of the RTS and RTR will

- be resistant to all forms of software attack and to the forms of physical attack implied by the TPM's Protection Profile, and
- supply an accurate digest of all sequences of presented integrity metrics.

9.4.4.2 Identity of the RTR

The TPM contains cryptographically verifiable identities for the RTR. The identity is in the form of asymmetric aliases (Endorsement Keys or EKs) derived from a common seed. Each seed value and its aliases should be statistically unique to a TPM. That is, the probability of two TPMs having the same EK should be insignificant. (See clause 14 for more information about primary seeds.)

The seed may be used to generate multiple asymmetric keys, all of which would represent the same TPM and RTR.

9.4.4.3 RTR Binding to a Platform

The TPM reports on the state of the platform by quoting the PCR values. For assurance that these PCR values accurately reflect that state, it is necessary to establish the binding between the RTR and the platform. A Platform Certificate can provide proof of this binding. The Platform Certificate is assurance from the certifying authority of the physical binding between the platform (the RTM) and the RTR.

9.4.4.4 Platform Identity and Privacy Considerations

The uniqueness of an EK and its cryptographic verifiability raises the issue of whether direct use of that identity could result in aggregation of activity logs. Analysis of the aggregated activity could reveal personal information that a user of a platform would not otherwise approve for distribution to the aggregators.

To counter undesired aggregation, TCG encourages the use of domain-specific signing keys and restrictions on the use of an EK. The Privacy Administrator controls use of an EK, including the process of binding another key to the EK.

NOTE Privacy Administrator's control of the EK differs from Owner control of the RTS providing separation of the security and identity uses of the TPM.

Unless the EK is certified by a trusted entity, its trust and privacy properties are no different from any other asymmetric key that can be generated by pure software methods. Therefore, by itself, the public portion of the EK is not privacy sensitive.

9.5 Basic Trusted Platform Features

9.5.1 Introduction

At a minimum, a trusted platform provides the three Roots of Trust described previously. All three roots use certification and attestation to provide evidence of the accuracy of information. A trusted platform will also offer Protected Locations (see 9.5.4) for the keys and data objects entrusted to it. Finally, a trusted platform may provide integrity measurement to ensure the trustworthiness of a platform by logging changes to platform state; this is done by recording logged entries in PCR for later validation as being correct and unaltered. These basic TPM concepts are now described in detail.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

9.5.2 Certification

The nominal method of establishing trust in a key is with a certificate indicating that the processes used for of creating and protecting the key meets necessary security criteria. A certificate may be provided by shipping the TPM with an embedded key (that is, an Endorsement Key) along with a Certificate of Authenticity for the EK. The EK and its certificate may be used to associate credentials (certificates) with other TPM keys; this process is specified in 9.5.3.3. When a certified key has attributes that let it sign TPM-created data, it may attest to the TPM-resident record of platform characteristics that affect the integrity (trustworthiness) of a platform.

NOTE The EK does not have to be installed when the TPM is shipped. At the factory, an EK could be generated from the Endorsement Seed and a Certificate of Authenticity created for that EK. The EK does not have to be permanently installed in the TPM. When the TPM is in possession of a customer, the customer could, at their discretion, have the TPM use the Endorsement Seed and recreate the EK for which they have a Certificate of Authenticity.

9.5.3 Attestation and Authentication

9.5.3.1 Types of Attestation

Trusted platforms employ a hierarchy of attestations:

1) An external entity attests to a TPM in order to vouch that the TPM is genuine and complies with ISO/IEC 11889. This attestation takes the form of an asymmetric key embedded in a genuine TPM, plus a credential that vouches for the public key of that pair.

   NOTE 1 A credential that is used to vouch for the embedded asymmetric key is commonly called an "Endorsement Certificate."

2) An external entity attests to a platform in order to vouch that the platform contains a Root-of-Trust-for-Measurement, a genuine TPM, plus a trusted path between the RTM and the TPM. This attestation takes the form of a credential that vouches for information including the public key of the asymmetric key pair in the TPM.

   NOTE 2 A credential used to vouch for the platform is commonly called a "Platform Certificate."

3) An external entity called an “Attestation CA” attests to an asymmetric key pair in a TPM in order to vouch that a key is protected by an unidentified but genuine TPM, and has particular properties. This attestation takes the form of a credential that vouches for information including the public key of the key pair. An Attestation CA typically relies upon attestations of type 1 and 2 in order to produce attestation of type 3.

   NOTE 3 The credential created by the CA is commonly called an "Attestation Key Certificate."

4) A trusted platform attests to an asymmetric key pair in order to vouch that a key pair is protected by a genuine but unidentified TPM and has particular properties. This attestation takes the form of a signature signed by the platform’s TPM over information that describes the key pair, using an attestation-key protected by the TPM, plus attestation of type 3 that vouches for that attestation-key.

   NOTE 4 This type of attestation is done using TPM2_Certify().

5) A trusted platform attests to a measurement in order to vouch that a particular software/firmware state exists in a platform. This attestation takes the form of a signature over a software/firmware measurement in a PCR using an attestation-key protected by the TPM, plus attestation of type 3 or 4 for that attestation-key.

   NOTE 5 This is type of attestation is commonly called a "quote" and is done with TPM2_Quote().

AVA Documents
Document Sharing Hub

 © ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

6) An external entity attests to a software/firmware measurement in order to vouch for particular software/firmware. This attestation takes the form of a credential that vouches for information including the value of a measurement and the state it represents.

NOTE 6 This is commonly called "third-party certification."

Attestation of types 3 and 4 entail the use of a key to sign the contents of Shielded Locations. An Attestation Key (AK) is a particular type of signing key that has a restriction on its use, in order to prevent forgery (the signing of external data that has the same format as genuine attestation data). The restriction is that an AK may be used only to sign a digest that the TPM has created. If an AK is known to be protected by a TPM (by virtue of attestation of type 3 or 4), it may be relied on to report accurately on Shielded Location content, and not sign externally provided data that appears to be valid and TPM-produced but is not.

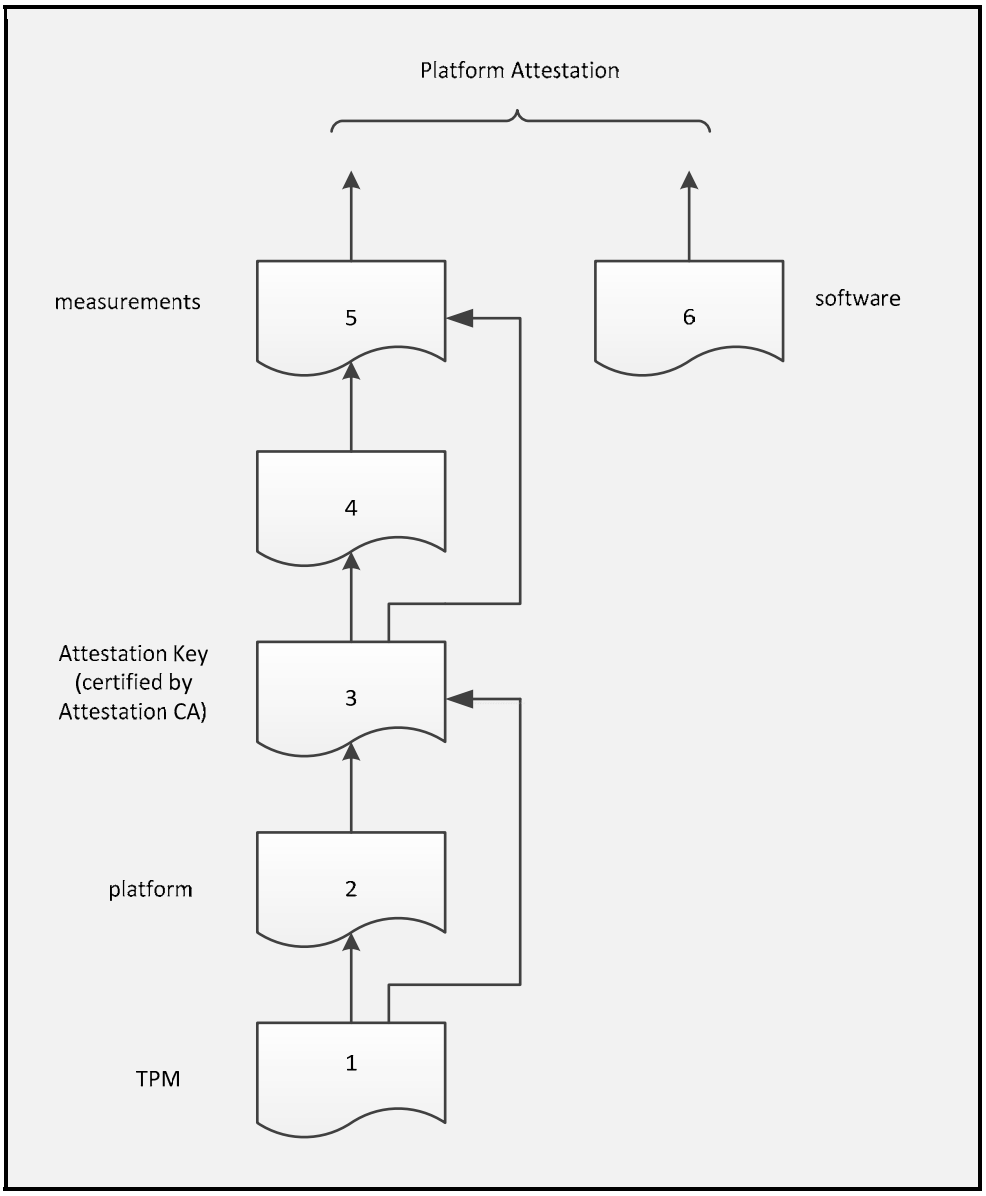

Figure 1 — Attestation Hierarchy

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 11889-1_2015

9.5.3.2 Attestation Keys

When the TPM creates a message to sign from internal TPM state, a special value (TPM_GENERATED_VALUE) is used as the message header. A TPM-generated message always begins with this value.

EXAMPLE 1 TPM2_Quote() is an example of a command where TPM creates a message to sign from internal TPM state.

When the TPM digests an externally provided message, it checks the first few octets of the message to ensure that they do not have the same value as TPM_GENERATED_VALUE. When the digest is complete, the TPM produces a ticket that indicates the message did not start with TPM_GENERATED_VALUE. When an AK is used to sign the digest, the caller provides the ticket so that the TPM can determine that the message used to create the digest was not a possible forgery of TPM attestation data.

NOTE The digest in the ticket must match the digest being presented to the AK for signing.

EXAMPLE 2 If an attacker produced a message block that was identical to a TPM-generated quote, that message block would start with TPM_GENERATED_VALUE to indicate that it is a proper TPM quote. When the TPM performs a digest of this block, it notes that the first octets are the same as TPM_GENERATED_VALUE. It will not generate the ticket indicating that the message is safe to sign, so an AK cannot be used to sign this digest. Similarly, an entity checking an attestation made by an AK can check that the message signed begins with TPM_GENERATED_VALUE in order to verify the message is indeed a TPM-generated quote.

Values signed by an AK may be assured to reflect TPM state, but AKs may also be used for general signing purposes.

An AK does not have much value to a remote challenger if the AK cannot be associated with the platform that it represents. This association is made using the identity certification process.

9.5.3.3 Attestation Key Identity Certification

Any TPM user that can create a key on a TPM can create a restricted-use signing key. The key creator may then ask a third party to provide a certificate for it. The third party may request that the caller provide some evidence that the key being certified is a TPM-resident key.

EXAMPLE 1 An example of an entity that might provide a certificate for a key is an attestation Certificate Authority (CA).

Evidence of TPM residency may be provided using a previously generated certificate for another key on the same TPM. An EK or Platform Certificate may provide this evidence.

NOTE 1 There is no requirement that certificates come only from a attestation CA. The method described above is an example of a scheme that could be used when privacy is required.

If a certified key may sign, it may be used to certify that some other object is resident on the same TPM. This allows the new AK to be linked to a certified key. A CA may use the certification from the TPM to produce a traditional certificate for the new key.

If the certified key is a decryption key and may not sign, then an alternative method is used to allow the new key or data object to be reliably certified. For this alternative certification, the identity of the Object to be certified and a certificate for the decryption key are provided to the CA. From the certificate, the CA determines the public key for the decryption key. The CA then produces a conditional certificate for the Object to be certified.

EXAMPLE 2 An EK is an example of a decryption key that may not sign.

AVA Documents
Document Sharing Hub

 © ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

The certificate is made conditional by performing some operation on the credential to produce a value that is required to be known before the credential can be used. This process produces a credential qualifier that is given to the TPM containing both the certified decryption key and the key to be certified.

EXAMPLE 3 Symmetrically encrypting a credential is an example of an operation to make its use conditional.

NOTE 2 A common credential qualifier would be a symmetric key that was used to encrypt the credential. Another option for the credential qualifier would be for it to be all or a portion of the signature of the certificate. Other options are possible.

The credential qualifier is protected using methods that are dependent on the type of the certified decryption key. The general method is specified in clause 24. Additional methods appropriate to RSA keys are specified in A.9.4 and additional methods appropriate to ECC key are in B.5.4. The protection process produces an encrypted blob, an HMAC over the blob, and a secret value that can only be recovered by the certified decryption key.

TPM2_ActivateCredential() is used to access the credential qualifier. The TPM recovers the secret value and uses it to generate the keys necessary to decrypt and validate the HMAC and encrypted blob. If the credential qualifier is recovered successfully, and the key being certified by the credential is loaded on the TPM, then the contents of the credential qualifier are returned to the caller. They may then use that value to complete the key certification.

NOTE 3 The protection process used for credential qualifiers is almost identical to the process used for key import. In order to make sure that there is no misuse of the encrypted structures, an application-specific value is used in the key recovery process. In the case of a credential qualifier, the label “IDENTITY” is used in the KDF that generates the keys (symmetric and HMAC) from the seed value. (See clause 5.4 for the definition of the label.)

TPM2_ActivateCredential() associates a credential with any object. The choice of attributes for an Object to be certified is at the discretion of the CA. Because a unique identifier for the Object is included in the integrity hash, the TPM enforces the credential’s accessibility only if the Object matches the criteria set by the CA as expressed in the object identifier.

9.5.4 Protected Location

When the sensitive portion of an object is not held in a Shielded Location on the TPM, it is encrypted. When encrypted, but not on the TPM, it is not protected from deletion, but it is protected from disclosure of its sensitive portions. Wherever it is stored, it is in a Protected Location.

Objects in long-term protected storage need to be loaded into the TPM for use. The application that created the objects manages their movement from long-term storage to the TPM.

Since a TPM has limited memory, it may be unable to hold all objects required by all applications simultaneously. The TPM supports swapping of object contexts by a TPM Resource Manager (TRM) so that the TPM can service these multiple applications. The object contexts are encrypted before being returned to the TRM by the TPM. If the object is needed later, the TRM can reload the context into the TPM providing a cache-like behavior.

Encryption of Protected Locations uses multiple seeds and keys that never leave the TPM. One of these is the Context Key. It is a symmetric key used to encrypt data when it is temporarily swapped out of the TPM so that a different working set of objects may be loaded. Other sensitive values that never leave the TPM are the Primary Seeds. These seeds are the root of the storage hierarchies that protect objects that are retained by applications. A Primary Seed is a random number used to generate protection keys for other objects; these objects may be Storage Keys that contain protection keys that are then used to protect still more objects.

Primary Seeds may be changed, and when they are changed, the objects they protected will no longer be usable.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

EXAMPLE The Storage Primary Seed (SPS) creates the Storage hierarchy for owner-related data, and that seed changes when the owner changes.

9.5.5 Integrity Measurement and Reporting

The Core Root of Trust for Measurement (CRTM) is the starting point of measurement. This process makes the initial measurements of the platform that are Extended into PCR in the TPM. For measurements to be meaningful, the executing code needs to control the environment in which it is running, so that the values recorded in the TPM are representative of the initial trust state of the platform.

A power-on reset creates an environment in which the platform is in a known initial state, with the main CPU running code from some well-defined initial location. Since that code has exclusive control of the platform at that time, it may make measurements of the platform from firmware. From these initial measurements, a chain of trust may be established. Because this chain of trust is created once when the platform is reset, no change of the initial trust state is possible, so it is called a static RTM (S-RTM).

An alternative method of initializing the platform is available on some processor architectures. It lets the CPU act as the CRTM and apply protections to portions of memory it measures. This process lets a new chain of trust start without rebooting the platform. Because the RTM may be re-established dynamically, this method is called dynamic RTM (D-RTM). Both S-RTM and D-RTM may take a system in an unknown state and return it to a known state. The D-RTM has the advantage of not requiring the system to be rebooted.

An integrity measurement is a value that represents a possible change in the trust state of the platform. The measured object may be anything of meaning but is often

- a data value,
- the hash of code or data, or
- an indication of the signer of some code or data.

The RTM (usually, code running on the CPU) makes these measurements and records them in RTS using Extend. The Extend process (see 17.2) allows the TPM to accumulate an indefinite number of measurements in a relatively small amount of memory.

The digest of an arbitrary set of integrity measurements is statistically unique, and an evaluator might know the values representing particular sequences of measurements. To handle cases where PCR values are not well known, the RTM keeps a log of individual measurements. The PCR values may be used to determine the accuracy of the log, and log entries may be evaluated individually to determine if the change in system state indicated by the event is acceptable.

Implementers play a role in determining how event data is partitioned. TCG's platform-specific specifications provide additional insight into specifying platform configuration and representation as well as anticipated consumers of measurement data.

Integrity reporting is the process of attesting to integrity measurements recorded in a PCR. The philosophy behind integrity measurement, logging, and reporting is that a platform may enter any state possible — including undesirable or insecure states — but is required to accurately report those states. An independent process may evaluate the integrity states and determine an appropriate response.

Documents
Document Sharing Hub

 © ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

10 TPM Protections

10.1 Introduction

This part of ISO/IEC 11889 describes the protections provided by the Trusted Platform Module. This clause describes the properties of selected capabilities and selected data locations for a TPM that has been evaluated according to a Protection Profile and a TPM that has not been modified by physical means.

TPM protections are based on the concepts of Protected Capabilities and Protected Objects. A Protected Capability is an operation that must be performed correctly for a TPM to be trusted. A Protected Object is data (including keys) that must be protected for a TPM operation to be trusted. Protected Objects in the TPM reside in Shielded Locations; the TPM may manipulate the contents of Shielded Locations only by using Protected Capabilities. Protected Objects outside Shielded Locations have their integrity and confidentiality protected cryptographically.

Since a Protected Object may reside outside of Shielded Location protections, the definition of “access” to a Protected Object denotes disclosure of its contents, not modification. Such objects are not protected against loss or tampering. However, before loading a Shielded Location with an outside object, the TPM will use a secure hash function to validate that the object was properly protected and not altered. If the integrity check fails, the TPM returns an error and does not load the object.

The only operations on Shielded Locations of a TPM are the Protected Capabilities defined in ISO/IEC 11889 and the vendor-specific operations that meet the requirements of 10.4.

10.2 Protection of Protected Capabilities

A Protected Capability may be modified only by other Protected Capabilities in the same TPM. Thus, the process of updating TPM firmware is required to be a Protected Capability.

10.3 Protection of Shielded Locations

As noted, access to any data on a TPM requires use of a Protected Capability. Therefore, all information on a TPM is in a Shielded Location. The contents of a Shielded Location are not disclosed unless the disclosure is intended by the definition of the Protected Capability. A TPM is not allowed to export data from a Shielded Location other than by using a Protected Capability.

NOTE Data in an I/O buffer that can be modified by the host is not “on” the TPM, even though the I/O buffer may be shielded from access while the TPM is processing a command or generating a response.

10.4 Exceptions and Clarifications

Vendor-specific operations may access and modify Shielded Locations on a TPM under the following circumstances.

- A vendor-specific operation may use the standard TPM authorization mechanism.
- A vendor-specific capability may read any TPM-resident structure that is not required to be in a Shielded Location at all times if the usage of that structure is authorized per the structure’s authorization mechanism.

  EXAMPLE A vendor-specific command could use the public portion of a key. If the key is a user key, no authorization would be required.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015

NOTE Among other things, the exception above enables access to a Shielded Location so the structure's access authorization can be checked.

- Vendor-specific operations may use a sequence of Protected Capabilities.
- Vendor-specific operations may use the standard TPM command interface or use a vendor-defined interface.

These clarifications serve to approve specific legitimate interpretations of the requirements.

- A vendor-specific operation that takes advantage of exceptions and clarifications to the "protection" requirements should be defined as part of the security target of the TPM. Such a vendor-specific command or capability should be evaluated to determine whether it meets Platform-specific TPM and System Security Targets.
- If a TPM stores vendor-specific cipher-text that is protected against subversion to the same or greater extent as internal TPM-resources stored outside the TPM with TCG-defined methods, then that cipher-text does not require protection from physical attack. If the TPM stores only vendor-specific cipher-text that does not require protection from physical attack, that location may be excluded from analysis when determining whether the TPM complies with the "physical protection" requirements specified by TCG.
- If a TPM uses external memory for non-volatile storage of TPM state (including seeds and proof values), movement of the TPM state to and from the NV memory constitutes a vendor-defined operation that is allowed by ISO/IEC 11889. The protection profile of that TPM should include a description of the protections of that data to insure confidentiality and integrity of the data and to mitigate against rollback attacks.

Documents
Document Sharing Hub

© ISO/IEC 2015 – All rights reserved

Full document details: ISO_IEC 11889-1_2015