

# OCF Security Specification

VERSION 2.2.1 | December 2020



**OPEN** CONNECTIVITY  
FOUNDATION™

CONTACT [admin@openconnectivity.org](mailto:admin@openconnectivity.org)

Copyright Open Connectivity Foundation, Inc. © 2020.  
All Rights Reserved.

## LEGAL DISCLAIMER

NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND OF LICENSE IN ITS CONTENT, EITHER EXPRESSLY OR IMPLIEDLY, OR TO ANY INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY ANY OF THE AUTHORS OR DEVELOPERS OF THIS DOCUMENT. THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OPEN INTERCONNECT CONSORTIUM, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

The OCF logo is a trademark of Open Connectivity Foundation, Inc. in the United States or other countries. \*Other names and brands may be claimed as the property of others.

Copyright © 2017-2020 Open Connectivity Foundation, Inc. All rights reserved.

Copying or other form of reproduction and/or distribution of these works are strictly prohibited

# CONTENTS

|    |  |  |     |
|----|--|--|-----|
| 17 |  |  |     |
| 18 | Introduction.....  |  | xiv |
| 19 | 1 Scope.....   |  | 1   |
| 20 | 2 Normative References .....   |  | 1   |
| 21 | 3 Terms, definitions, and abbreviated terms .....                            |  | 4   |
| 22 | 3.1 Terms and definitions.....   |  | 4   |
| 23 | NOTE: The Owner Transfer Method selected will determine the specifics of the |  |     |
| 24 | DOC used.....  |  | 7   |
| 25 | 3.2 Symbols and abbreviated terms .....                                      |  | 8   |
| 26 | 4 Document Conventions and Organization .....                                |  | 9   |
| 27 | 4.1 Conventions.....   |  | 9   |
| 28 | 4.2 Notation .....   |  | 10  |
| 29 | 4.3 Data types .....   |  | 11  |
| 30 | 4.4 Document structure.....  |  | 11  |
| 31 | 5 Security Overview.....   |  | 12  |
| 32 | 5.1 Security Model of Operation.....   |  | 12  |
| 33 | 5.2 Access Control.....  |  | 17  |
| 34 | 5.2.1 Access Control General .....   |  | 17  |
| 35 | 5.2.2 ACL Architecture .....   |  | 19  |
| 36 | 5.3 Onboarding Overview .....  |  | 20  |
| 37 | 5.3.1 Onboarding General .....   |  | 20  |
| 38 | 5.3.2 Onboarding Steps.....  |  | 22  |
| 39 | 5.3.3 Establishing a Device Owner .....                                      |  | 23  |
| 40 | 5.3.4 Provisioning for Normal Operation .....                                |  | 24  |
| 41 | 5.3.5 OCF Compliance Management System.....                                  |  | 24  |
| 42 | 5.4 Provisioning.....  |  | 24  |
| 43 | 5.4.1 Provisioning General .....   |  | 24  |
| 44 | 5.4.2 Access Control Provisioning .....                                      |  | 25  |
| 45 | 5.4.3 Credential Provisioning.....   |  | 25  |
| 46 | 5.4.4 Role Provisioning .....  |  | 25  |
| 47 | 5.5 Secure Resource Manager (SRM).....                                       |  | 25  |
| 48 | 5.6 Credential Overview.....   |  | 26  |
| 49 | 5.7 Event Logging.....   |  | 26  |
| 50 | 5.7.1 Event Logging General .....  |  | 26  |
| 51 | 5.8 End-to-End Security of Unicast Messages.....                             |  | 28  |
| 52 | 5.9 Overview of Simple Secure Multicast .....                                |  | 28  |
| 53 | 6 Security for the Discovery Process .....                                   |  | 30  |
| 54 | 6.1 Preamble .....   |  | 30  |
| 55 | 6.2 Security Considerations for Discovery.....                               |  | 30  |
| 56 | 7 Security Provisioning .....  |  | 32  |
| 57 | 7.1 Device Identity .....  |  | 32  |
| 58 | 7.1.1 General Device Identity .....  |  | 32  |
| 59 | 7.1.2 Device Identity for Devices with UAID [Deprecated].....                |  | 32  |
| 60 | 7.2 Device Ownership.....  |  | 32  |

|     |       |  |    |
|-----|-------|--|----|
| 61  | 7.3   | Device Ownership Transfer Methods.....                               | 33 |
| 62  | 7.3.1 | OTM implementation requirements .....                                | 33 |
| 63  | 7.3.2 | SharedKey Credential Calculation .....                               | 34 |
| 64  | 7.3.3 | Certificate Credential Generation.....                               | 35 |
| 65  | 7.3.4 | Just-Works OTM.....  | 35 |
| 66  | 7.3.5 | Random PIN based OTM .....   | 37 |
| 67  | 7.3.6 | Manufacturer Certificate Based OTM .....                             | 40 |
| 68  | 7.3.7 | Vendor Specific OTMs .....   | 43 |
| 69  | 7.3.8 | Establishing Owner Credentials .....                                 | 44 |
| 70  | 7.3.9 | Security Profile Assignment .....                                    | 47 |
| 71  | 7.4   | Provisioning .....   | 48 |
| 72  | 7.4.1 | Provisioning Flows.....  | 48 |
| 73  | 8     | Device Onboarding State Definitions .....                            | 49 |
| 74  | 8.1   | Device Onboarding General.....                                       | 49 |
| 75  | 8.2   | Device Onboarding-Reset State Definition .....                       | 50 |
| 76  | 8.3   | Device Ready-for-OTM State Definition.....                           | 51 |
| 77  | 8.4   | Device Ready-for-Provisioning State Definition .....                 | 52 |
| 78  | 8.5   | Device Ready-for-Normal-Operation State Definition.....              | 53 |
| 79  | 8.6   | Device Soft Reset State Definition .....                             | 53 |
| 80  | 9     | Security Credential Management.....                                  | 55 |
| 81  | 9.1   | Preamble .....   | 55 |
| 82  | 9.2   | Credential Lifecycle .....   | 55 |
| 83  | 9.2.1 | Credential Lifecycle General.....                                    | 55 |
| 84  | 9.2.2 | Creation .....   | 55 |
| 85  | 9.2.3 | Deletion.....  | 55 |
| 86  | 9.2.4 | Refresh .....  | 55 |
| 87  | 9.2.5 | Revocation .....   | 55 |
| 88  | 9.3   | Credential Types.....  | 56 |
| 89  | 9.3.1 | Preamble.....  | 56 |
| 90  | 9.3.2 | Pair-wise Symmetric Key Credentials .....                            | 56 |
| 91  | 9.3.3 | Group Symmetric Key Credentials .....                                | 56 |
| 92  | 9.3.4 | Asymmetric Authentication Key Credentials.....                       | 57 |
| 93  | 9.3.5 | Asymmetric Key Encryption Key Credentials.....                       | 57 |
| 94  | 9.3.6 | Certificate Credentials .....  | 57 |
| 95  | 9.3.7 | Password Credentials.....  | 58 |
| 96  | 9.3.8 | Credentials for direct provisioning an OSCORE Security Context ..... | 58 |
| 97  | 9.3.9 | Credentials for Simple Secure Multicast .....                        | 58 |
| 98  | 9.4   | Certificate Based Key Management .....                               | 59 |
| 99  | 9.4.1 | Overview .....   | 59 |
| 100 | 9.4.2 | X.509 Digital Certificate Profiles .....                             | 60 |
| 101 | 9.4.3 | Certificate Revocation List (CRL) Profile [Deprecated].....          | 68 |
| 102 | 9.4.4 | Resource Model .....   | 68 |
| 103 | 9.4.5 | Certificate Provisioning.....  | 69 |
| 104 | 9.4.6 | CRL Provisioning [Deprecated].....                                   | 69 |
| 105 | 10    | Device Authentication.....   | 70 |

|     |        |  |    |
|-----|--------|--|----|
| 106 | 10.1   | Device Authentication General.....                             | 70 |
| 107 | 10.2   | Device Authentication with Symmetric Key Credentials .....     | 70 |
| 108 | 10.3   | Device Authentication with Raw Asymmetric Key Credentials..... | 70 |
| 109 | 10.4   | Device Authentication with Certificates .....                  | 70 |
| 110 | 10.4.1 | Device Authentication with Certificates General .....          | 70 |
| 111 | 10.4.2 | Role Assertion with Certificates .....                         | 71 |
| 112 | 10.4.3 | OCF PKI Roots .....  | 72 |
| 113 | 10.4.4 | PKI Trust Store.....   | 72 |
| 114 | 10.4.5 | Path Validation and extension processing.....                  | 72 |
| 115 | 11     | Message Integrity and Confidentiality .....                    | 73 |
| 116 | 11.1   | Preamble .....   | 73 |
| 117 | 11.2   | Session Protection with DTLS .....                             | 73 |
| 118 | 11.2.1 | DTLS Protection General.....                                   | 73 |
| 119 | 11.2.2 | Unicast Session Semantics.....                                 | 73 |
| 120 | 11.3   | Cipher Suites .....  | 73 |
| 121 | 11.3.1 | Cipher Suites General .....                                    | 73 |
| 122 | 11.3.2 | Cipher Suites for Device Ownership Transfer .....              | 73 |
| 123 | 11.3.3 | Cipher Suites for Symmetric Keys.....                          | 74 |
| 124 | 11.3.4 | Cipher Suites for Asymmetric Credentials .....                 | 74 |
| 125 | 12     | Access Control .....   | 76 |
| 126 | 12.1   | ACL Generation and Management .....                            | 76 |
| 127 | 12.2   | ACL Evaluation and Enforcement.....                            | 76 |
| 128 | 12.2.1 | ACL Evaluation and Enforcement General.....                    | 76 |
| 129 | 12.2.2 | Host Reference Matching .....                                  | 76 |
| 130 | 12.2.3 | Resource Wildcard Matching .....                               | 76 |
| 131 | 12.2.4 | Multiple Criteria Matching .....                               | 77 |
| 132 | 12.2.5 | Subject Matching using Wildcards .....                         | 77 |
| 133 | 12.2.6 | Subject Matching using Roles.....                              | 77 |
| 134 | 12.2.7 | ACL Evaluation.....  | 78 |
| 135 | 13     | Security Resources .....                                       | 80 |
| 136 | 13.1   | Security Resources General .....                               | 80 |
| 137 | 13.2   | Device Owner Transfer Resource .....                           | 82 |
| 138 | 13.2.1 | Device Owner Transfer Resource General.....                    | 82 |
| 139 | 13.2.2 | OCF defined OTMs.....  | 85 |
| 140 | 13.3   | Credential Resource .....                                      | 85 |
| 141 | 13.3.1 | Credential Resource General.....                               | 85 |
| 142 | 13.3.2 | Properties of the Credential Resource .....                    | 91 |
| 143 | 13.3.3 | Key Formatting .....   | 93 |
| 144 | 13.3.4 | Credential Refresh Method Details [Deprecated] .....           | 94 |
| 145 | 13.4   | Certificate Revocation List .....                              | 94 |
| 146 | 13.4.1 | CRL Resource Definition [Deprecated] .....                     | 94 |
| 147 | 13.5   | ACL Resources.....   | 94 |
| 148 | 13.5.1 | ACL Resources General .....                                    | 94 |
| 149 | 13.5.2 | OCF Access Control List (ACL) BNF defines ACL structures. .... | 94 |
| 150 | 13.5.3 | ACL Resource .....   | 95 |

|     |         |  |     |
|-----|---------|--|-----|
| 151 | 13.6    | Access Manager ACL Resource [Deprecated].....                        | 100 |
| 152 | 13.7    | Signed ACL Resource [Deprecated].....                                | 100 |
| 153 | 13.8    | Provisioning Status Resource .....                                   | 100 |
| 154 | 13.9    | Certificate Signing Request Resource.....                            | 105 |
| 155 | 13.10   | Roles Resource .....   | 106 |
| 156 | 13.11   | Auditable Events List Resource .....                                 | 107 |
| 157 | 13.11.1 | Auditable Events List Resource General .....                         | 107 |
| 158 | 13.12   | Security Virtual Resources (SVRs) and Access Policy .....            | 110 |
| 159 | 13.13   | SVRs, Discoverability and OCF Endpoints .....                        | 111 |
| 160 | 13.14   | Additional Privacy Consideration for Core Resources .....            | 111 |
| 161 | 13.15   | Easy Setup Resource Device State.....                                | 112 |
| 162 | 13.16   | List of Auditable Events .....                                       | 114 |
| 163 | 13.17   | Security Domain Information Resource .....                           | 116 |
| 164 | 14      | Security Hardening Guidelines/ Execution Environment Security .....  | 117 |
| 165 | 14.1    | Preamble .....   | 117 |
| 166 | 14.2    | Execution Environment Elements.....                                  | 117 |
| 167 | 14.2.1  | Execution Environment Elements General .....                         | 117 |
| 168 | 14.2.2  | Secure Storage.....  | 117 |
| 169 | 14.2.3  | Secure execution engine .....  | 120 |
| 170 | 14.2.4  | Trusted input/output paths .....                                     | 120 |
| 171 | 14.2.5  | Secure clock.....  | 120 |
| 172 | 14.2.6  | Approved algorithms.....   | 120 |
| 173 | 14.2.7  | Hardware tamper protection.....                                      | 121 |
| 174 | 14.3    | Secure Boot.....   | 121 |
| 175 | 14.3.1  | Concept of software module authentication.....                       | 121 |
| 176 | 14.3.2  | Secure Boot process .....  | 123 |
| 177 | 14.3.3  | Robustness Requirements.....   | 123 |
| 178 | 14.4    | Attestation .....  | 123 |
| 179 | 14.5    | Software Update .....  | 123 |
| 180 | 14.5.1  | Overview .....   | 123 |
| 181 | 14.5.2  | Recognition of Current Differences .....                             | 124 |
| 182 | 14.5.3  | Software Version Validation.....                                     | 125 |
| 183 | 14.5.4  | Software Update.....   | 125 |
| 184 | 14.5.5  | Recommended Usage.....   | 125 |
| 185 | 14.6    | Non-OCF Endpoint interoperability.....                               | 126 |
| 186 | 14.7    | Security Levels .....  | 126 |
| 187 | 14.8    | Security Profiles.....   | 127 |
| 188 | 14.8.1  | Security Profiles General .....                                      | 127 |
| 189 | 14.8.2  | Identification of Security Profiles (Normative) .....                | 127 |
| 190 | 14.8.3  | Security Profiles .....  | 129 |
| 191 | 15      | Device Type Specific Requirements.....                               | 134 |
| 192 | 15.1    | Bridging Security .....  | 134 |
| 193 | 15.1.1  | Universal Requirements for Bridging to another Ecosystem .....       | 134 |
| 194 | 15.1.2  | Additional Security Requirements specific to Bridged Protocols ..... | 135 |
| 195 |         | Annex A (informative) Access Control Examples.....                   | 137 |

|     |                       |   |     |
|-----|-----------------------|---|-----|
| 196 | 16                    | Alternative in-transit protection mechanisms.....                         | 137 |
| 197 | 16.1                  | Introduction to in-transit protection mechanisms .....                    | 137 |
| 198 | 16.2                  | End-to-End Security of Unicast Messages using OSCORE.....                 | 137 |
| 199 | 16.2.1                | Introduction to End-to-End Security of Unicast Messages using OSCORE ...  | 137 |
| 200 | 16.2.2                | OSCORE ID Namespace Prefix .....  | 137 |
| 201 | 16.2.3                | OSCORE protection and verification of unicast OCF CRUDN messages .....    | 138 |
| 202 | 16.2.4                | Direct provisioning of an OSCORE Security Context.....                    | 139 |
| 203 | 16.3                  | Simple Secure Multicast.....  | 140 |
| 204 | 16.3.1                | Introduction to Simple Secure Multicast .....                             | 140 |
| 205 | 16.3.2                | Assumptions and Prerequisites for Simple Secure Multicast .....           | 141 |
| 206 | 16.3.3                | OSCORE protection and verification of Simple Secure Multicast Requests .. | 142 |
| 207 | 16.3.4                | Creating OSCORE Security Context for Simple Secure Multicast.....         | 143 |
| 208 | A.1                   | Example OCF ACL Resource .....  | 144 |
| 209 | Annex B (Informative) | Execution Environment Security Profiles .....                             | 146 |
| 210 | Annex C (normative)   | Resource Type definitions.....  | 147 |
| 211 | C.1                   | List of Resource Type definitions .....                                   | 147 |
| 212 | C.2                   | Access Control List-2 .....   | 147 |
| 213 | C.2.1                 | Introduction .....  | 147 |
| 214 | C.2.2                 | Well-known URI .....  | 147 |
| 215 | C.2.3                 | Resource type .....   | 147 |
| 216 | C.2.4                 | OpenAPI 2.0 definition.....   | 147 |
| 217 | C.2.5                 | Property definition .....   | 155 |
| 218 | C.2.6                 | CRUDN behaviour .....   | 156 |
| 219 | C.3                   | Credential .....  | 156 |
| 220 | C.3.1                 | Introduction .....  | 156 |
| 221 | C.3.2                 | Well-known URI .....  | 156 |
| 222 | C.3.3                 | Resource type .....   | 156 |
| 223 | C.3.4                 | OpenAPI 2.0 definition.....   | 156 |
| 224 | C.3.5                 | Property definition .....   | 166 |
| 225 | C.3.6                 | CRUDN behaviour .....   | 167 |
| 226 | C.4                   | Certificate Signing Request.....  | 167 |
| 227 | C.4.1                 | Introduction .....  | 167 |
| 228 | C.4.2                 | Well-known URI .....  | 167 |
| 229 | C.4.3                 | Resource type .....   | 167 |
| 230 | C.4.4                 | OpenAPI 2.0 definition.....   | 167 |
| 231 | C.4.5                 | Property definition .....   | 169 |
| 232 | C.4.6                 | CRUDN behaviour .....   | 169 |
| 233 | C.5                   | Device Owner Transfer Method.....   | 169 |
| 234 | C.5.1                 | Introduction .....  | 169 |
| 235 | C.5.2                 | Well-known URI .....  | 169 |
| 236 | C.5.3                 | Resource type .....   | 169 |
| 237 | C.5.4                 | OpenAPI 2.0 definition.....   | 169 |
| 238 | C.5.5                 | Property definition .....   | 173 |
| 239 | C.5.6                 | CRUDN behaviour .....   | 174 |
| 240 | C.6                   | Device Provisioning Status .....  | 175 |

|     |                       |   |     |
|-----|-----------------------|---|-----|
| 241 | C.6.1                 | Introduction .....  | 175 |
| 242 | C.6.2                 | Well-known URI .....  | 175 |
| 243 | C.6.3                 | Resource type .....   | 175 |
| 244 | C.6.4                 | OpenAPI 2.0 definition.....   | 175 |
| 245 | C.6.5                 | Property definition .....   | 179 |
| 246 | C.6.6                 | CRUDN behaviour .....   | 182 |
| 247 | C.7                   | Asserted Roles .....  | 182 |
| 248 | C.7.1                 | Introduction .....  | 182 |
| 249 | C.7.2                 | Well-known URI .....  | 182 |
| 250 | C.7.3                 | Resource type .....   | 182 |
| 251 | C.7.4                 | OpenAPI 2.0 definition.....   | 182 |
| 252 | C.7.5                 | Property definition .....   | 191 |
| 253 | C.7.6                 | CRUDN behaviour .....   | 191 |
| 254 | C.8                   | Security Profile .....  | 192 |
| 255 | C.8.1                 | Introduction .....  | 192 |
| 256 | C.8.2                 | Well-known URI .....  | 192 |
| 257 | C.8.3                 | Resource type .....   | 192 |
| 258 | C.8.4                 | OpenAPI 2.0 definition.....   | 192 |
| 259 | C.8.5                 | Property definition .....   | 194 |
| 260 | C.8.6                 | CRUDN behaviour .....   | 194 |
| 261 | C.9                   | Auditable Event List .....  | 195 |
| 262 | C.9.1                 | Introduction .....  | 195 |
| 263 | C.9.2                 | Well-known URI .....  | 195 |
| 264 | C.9.3                 | Resource type .....   | 195 |
| 265 | C.9.4                 | OpenAPI 2.0 definition.....   | 195 |
| 266 | C.9.5                 | Property definition .....   | 199 |
| 267 | C.9.6                 | CRUDN behaviour .....   | 201 |
| 268 | C.10                  | Security Domain Information .....                                       | 201 |
| 269 | C.10.1                | Introduction .....  | 201 |
| 270 | C.10.2                | Well-known URI .....  | 202 |
| 271 | C.10.3                | Resource type .....   | 202 |
| 272 | C.10.4                | OpenAPI 2.0 definition.....   | 202 |
| 273 | C.10.5                | Property definition .....   | 204 |
| 274 | C.10.6                | CRUDN behaviour .....   | 205 |
| 275 | Annex D (informative) | OID definitions .....   | 206 |
| 276 | Annex E (informative) | Security considerations specific to Bridged Protocols .....             | 208 |
| 277 | E.1                   | Security Considerations specific to the AllJoyn Protocol .....          | 208 |
| 278 | E.2                   | Security Considerations specific to the Bluetooth LE Protocol .....     | 208 |
| 279 | E.3                   | Security Considerations specific to the oneM2M Protocol .....           | 208 |
| 280 | E.4                   | Security Considerations specific to the U+ Protocol .....               | 208 |
| 281 | E.5                   | Security Considerations specific to the Z-Wave Protocol.....            | 209 |
| 282 | E.6                   | Security Considerations specific to the Zigbee Protocol .....           | 210 |
| 283 | E.7                   | Security Considerations specific to the the EnOcean Radio Protocol..... | 211 |
| 284 |                       |   |     |



## FIGURES

|     |   |     |
|-----|---|-----|
| 285 |   |     |
| 286 | Figure 1 – OCF Interaction.....   | 10  |
| 287 | Figure 2 – OCF Layers for direct Device-to-Device interaction .....                     | 12  |
| 288 | Figure 3 – OCF Layers for interactions via one OCF Proxy .....                          | 14  |
| 289 | Figure 4 – OCF Layers for interactions via two OCF Proxies.....                         | 14  |
| 290 | Figure 5 – Single request reaches a group of Servers.....                               | 16  |
| 291 | Figure 6 – OCF Layers for Simple Secure Multicast .....                                 | 16  |
| 292 | Figure 7 – OCF Security Enforcement Points .....  | 17  |
| 293 | Figure 8 – Use case-1 showing simple ACL enforcement .....                              | 19  |
| 294 | Figure 9 – Onboarding overview .....  | 21  |
| 295 | Figure 10 – OCF Onboarding Process .....  | 23  |
| 296 | Figure 11 – OCF's SRM Architecture .....  | 26  |
| 297 | Figure 12 – Store Events in local storage.....  | 27  |
| 298 | Figure 13 – Relationship diagram for Simple Secure Multicast messages .....             | 29  |
| 299 | Figure 14 – Setup and usage of Secure Simple Multicast.....                             | 29  |
| 300 | Figure 15 – Discover New Device Sequence.....   | 33  |
| 301 | Figure 16 – A Just Works OTM .....  | 36  |
| 302 | Figure 17 – Random PIN-based OTM .....  | 38  |
| 303 | Figure 18 – Manufacturer Certificate Based OTM Sequence .....                           | 42  |
| 304 | Figure 19 – Vendor-specific Owner Transfer Sequence.....                                | 44  |
| 305 | Figure 20 – Symmetric Owner Credential Provisioning Sequence .....                      | 46  |
| 306 | Figure 21 – Example of Client-directed provisioning.....                                | 48  |
| 307 | Figure 22 – Device state model.....   | 50  |
| 308 | Figure 23 – Client-directed Certificate Transfer.....                                   | 69  |
| 309 | Figure 24 – Asserting a role with a certificate role credential. ....                   | 72  |
| 310 | Figure 25 – OCF Security Resources .....  | 80  |
| 311 | Figure 26 – "/oic/sec/cred" Resource and Properties.....                                | 81  |
| 312 | Figure 27 – "/oic/sec/acl2" Resource and Properties.....                                | 81  |
| 313 | Figure 28 – "/oic/sec/ael" Resource and Properties.....                                 | 82  |
| 314 | Figure 29 – Example of Soft AP and Easy Setup Resource in different Device states ..... | 112 |
| 315 | Figure 30 – Software Module Authentication .....  | 122 |
| 316 | Figure 31 – Verification Software Module.....   | 122 |
| 317 | Figure 32 – Software Module Authenticity .....  | 123 |
| 318 | Figure 33 – State transitioning diagram for software download .....                     | 124 |
| 319 | Figure 34 – Simple Multicast requests .....   | 140 |
| 320 | Figure A-1 – Example "/oic/sec/acl2" Resource.....                                      | 145 |
| 321 | Figure E-1 Security Considerations for BLE Bridge .....                                 | 208 |
| 322 | Figure E-2 Security Considerations for Z-Wave Bridge.....                               | 209 |
| 323 | Figure E-3 Security Considerations for Zigbee Bridge .....                              | 211 |
| 324 | Figure E-4 Security Considerations for EnOcean Bridge .....                             | 212 |



## Tables

|   |     |
|---|-----|
| Table 1 – Discover New Device Details.....                            | 34  |
| Table 2 – A Just Works OTM Details.....                               | 36  |
| Table 3 – Random PIN-based OTM Details.....                           | 38  |
| Table 4 – Manufacturer Certificate Based OTM Details .....            | 42  |
| Table 5 – Vendor-specific Owner Transfer Details .....                | 44  |
| Table 6 – Symmetric Owner Credential Assignment Details .....         | 46  |
| Table 7 – Steps describing Client -directed provisioning .....        | 49  |
| Table 8 – X.509 v1 fields for Root CA Certificates.....               | 60  |
| Table 9 - X.509 v3 extensions for Root CA Certificates .....          | 61  |
| Table 10 - X.509 v1 fields for Intermediate CA Certificates .....     | 61  |
| Table 11 – X.509 v3 extensions for Intermediate CA Certificates ..... | 61  |
| Table 12 – X.509 v1 fields for End-Entity Certificates.....           | 62  |
| Table 13 – X.509 v3 extensions for End-Entity Certificates .....      | 62  |
| Table 14 – ACE2 Wildcard Matching Strings Description.....            | 76  |
| Table 15 – Definition of the "/oic/sec/doxm" Resource .....           | 82  |
| Table 16 – Properties of the "/oic/sec/doxm" Resource .....           | 82  |
| Table 17 – Properties of the "oic.sec.didtype" type .....             | 84  |
| Table 18 – Properties of the "oic.sec.doxmtype" type.....             | 85  |
| Table 19 – Definition of the "/oic /sec/cred" Resource.....           | 86  |
| Table 20 – Properties of the "/oic/sec/cred" Resource.....            | 87  |
| Table 21 – Properties of the "oic.sec.creds" Property.....            | 88  |
| Table 22: Properties of the "oic.sec.credusagetype" Property .....    | 90  |
| Table 23 – Properties of the "oic.sec.pubdatatype" Property .....     | 90  |
| Table 24 – Properties of the "oic.sec.privdatatype" Property .....    | 90  |
| Table 25 – Properties of the "oic.sec.optdatatype" Property .....     | 91  |
| Table 26 – Definition of the "oic.sec.roletype" type. ....            | 91  |
| Table 27 – Definition of the "oic.sec.oscoretype" type.....           | 91  |
| Table 28 – 128-bit symmetric key .....                                | 93  |
| Table 29 – 256-bit symmetric key .....                                | 93  |
| Table 30 – BNF Definition of OCF ACL .....                            | 94  |
| Table 31 – Value Definition of the "oic.sec.crudntype" Property ..... | 96  |
| Table 32 – Definition of the "oic/sec/acl2" Resource .....            | 96  |
| Table 33 – Properties of the "/oic/sec/acl2" Resource .....           | 97  |
| Table 34 – "oic.sec.ace2" data type definition. ....                  | 98  |
| Table 35 – "oic.sec.ace2.resource-ref" data type definition. ....     | 98  |
| Table 36 – Value definition "oic.sec.conntype" Property .....         | 98  |
| Table 37 – Definition of the "/oic/sec/pstat" Resource .....          | 100 |
| Table 38 – Properties of the "/oic/sec/pstat" Resource .....          | 101 |
| Table 39 – Properties of the ".oic.sec.dostype" Property .....        | 102 |

|     |   |     |
|-----|---|-----|
| 366 | Table 40 – Definition of the "oic.sec.dpmttype" Property .....                              | 104 |
| 367 | Table 41 – Value Definition of the "oic.sec.dpmttype" Property (Low-Byte) .....             | 104 |
| 368 | Table 42 – Value Definition of the "oic.sec.dpmttype" Property (High-Byte).....             | 104 |
| 369 | Table 43 – Definition of the "oic.sec.pomtype" Property .....                               | 104 |
| 370 | Table 44 – Value Definition of the "oic.sec.pomtype" Property .....                         | 105 |
| 371 | Table 45 – Definition of the "/oic/sec/csr" Resource .....                                  | 105 |
| 372 | Table 46 – Properties of the "oic.r.csr" Resource .....                                     | 105 |
| 373 | Table 47 – Definition of the "/oic/sec/roles" Resource .....                                | 107 |
| 374 | Table 48 – Properties of the "/oic/sec/roles" Resource .....                                | 107 |
| 375 | Table 49 – Definition of the "/oic/sec/ael" Resource .....                                  | 108 |
| 376 | Table 50 – Properties of the "/oic/sec/ael" Resource.....                                   | 108 |
| 377 | Table 51 – "oic.sec.aee" data type definition.....  | 110 |
| 378 | Table 52 – Core Resource Properties Access Modes given various Device States .....          | 111 |
| 379 | Table 53 – List of mandatory Auditable Events and corresponding Property values.....        | 114 |
| 380 | Table 54 – List of recommended Auditable Events and corresponding Property values .....     | 115 |
| 381 | Table 55 –Definition of the "oic.r.sdi" Resource Type.....                                  | 116 |
| 382 | Table 56 – Properties of the "oic.r.sdi" Resource Type.....                                 | 116 |
| 383 | Table 57 – Examples of Sensitive Data.....  | 118 |
| 384 | Table 58 – Description of the software update bits .....                                    | 124 |
| 385 | Table 59 – Definition of the "/oic/sec/sp" Resource .....                                   | 128 |
| 386 | Table 60 – Properties of the "/oic/sec/sp" Resource .....                                   | 128 |
| 387 | Table 61 – Dependencies of VOD Behaviour on Bridge state, as clarification of               |     |
| 388 | accompanying text.....  | 135 |
| 389 | Table 62 – OSCORE Identifier Namespace Prefix.....  | 138 |
| 390 | Table B.1 – OCF Security Profile .....  | 146 |
| 391 | Table C.1 – Alphabetized list of security Resources.....                                    | 147 |
| 392 | Table C-1 – The Property definitions of the Resource with type "rt" = "oic.r.acl2". .....   | 155 |
| 393 | Table C-2 – The CRUDN operations of the Resource with type "rt" = "oic.r.acl2". .....       | 156 |
| 394 | Table C-3 – The Property definitions of the Resource with type "rt" = "oic.r.cred". .....   | 166 |
| 395 | Table C-4 – The CRUDN operations of the Resource with type "rt" = "oic.r.cred". .....       | 167 |
| 396 | Table C-5 – The Property definitions of the Resource with type "rt" = "oic.r.csr". .....    | 169 |
| 397 | Table C-6 – The CRUDN operations of the Resource with type "rt" = "oic.r.csr". .....        | 169 |
| 398 | Table C-7 – The Property definitions of the Resource with type "rt" = "oic.r.doxm". .....   | 173 |
| 399 | Table C-8 – The CRUDN operations of the Resource with type "rt" = "oic.r.doxm". .....       | 175 |
| 400 | Table C-9 – The Property definitions of the Resource with type "rt" = "oic.r.pstat". .....  | 179 |
| 401 | Table C-10 – The CRUDN operations of the Resource with type "rt" = "oic.r.pstat". .....     | 182 |
| 402 | Table C-11 – The Property definitions of the Resource with type "rt" = "oic.r.roles". ..... | 191 |
| 403 | Table C-12 – The CRUDN operations of the Resource with type "rt" = "oic.r.roles". .....     | 192 |
| 404 | Table C-13 – The Property definitions of the Resource with type "rt" = "oic.r.sp". .....    | 194 |
| 405 | Table C-14 – The CRUDN operations of the Resource with type "rt" = "oic.r.sp". .....        | 195 |

|     |   |     |
|-----|---|-----|
| 406 | Table C-15 – The Property definitions of the Resource with type "rt" = "oic.r.ael". | 199 |
| 407 | Table C-16 – The CRUDN operations of the Resource with type "rt" = "oic.r.ael".     | 201 |
| 408 | Table C-17 – The Property definitions of the Resource with type "rt" = "oic.r.sdi". | 204 |
| 409 | Table C-18 – The CRUDN operations of the Resource with type "rt" = "oic.r.sdi".     | 205 |
| 410 | Table E.1 GAP security mode   | 208 |
| 411 | Table E.2 TLS 1.2 Cipher Suites used by U+  | 209 |
| 412 | Table E.3 Z-Wave Security Class   | 210 |
| 413 | Table E.4 Zigbee 3.0 Security Levels to the Network, and Application Support layers | 210 |
| 414 | Table E.5 EnOcean Radio Protocol security levels                                    | 211 |
| 415 |   |     |
| 416 |   |     |

## Change History

(DELETE BEFORE PUBLISHING)

| Release | Date        | Description   |
|---------|-------------|---|
| 1       | 13-Nov-2020 | (Baseline; OCF_Security_Specification_v2.2.0_-ISO-IEC)<br><br>Bug 1657 - End-to-End Security (Security Spec).<br>Bug 1967 - Distributed Key Management for Secure Multicast<br>Bug 2335 - Clarify when OBT loses "admin" privileges during onboarding.<br>Bug 2640 - DOXM support for multicast and query using "owned" is not documented nor in schemas.<br>Bug 3047 - Writing "oxmsel" and "owned".<br>Bug 3305 - Devices shall limit exposed network ports.<br><br>Editorial Cleanup:<br>Remove unused abbreviations |
| 2       | 13-Nov-2020 | Regenerate schema annex from github   |
| 3       | 27-Nov-2020 | Remove unused Normative References.<br>Minor editorial fixes.<br>Bug 3286 - clean up RFNOP.   |

## Introduction

This document, and all the other parts associated with this document, were developed in response to worldwide demand for smart home focused Internet of Things (IoT) devices, such as appliances, door locks, security cameras, sensors, and actuators; these to be modelled and securely controlled, locally and remotely, over an IP network.

While some inter-device communication existed, no universal language had been developed for the IoT. Device makers instead had to choose between disparate frameworks, limiting their market share, or developing across multiple ecosystems, increasing their costs. The burden then falls on end users to determine whether the products they want are compatible with the ecosystem they bought into, or find ways to integrate their devices into their network, and try to solve interoperability issues on their own.

In addition to the smart home, IoT deployments in commercial environments are hampered by a lack of security. This issue can be avoided by having a secure IoT communication framework, which this standard solves.

The goal of these documents is then to connect the next 25 billion devices for the IoT, providing secure and reliable device discovery and connectivity across multiple OSs and platforms. There are multiple proposals and forums driving different approaches, but no single solution addresses the majority of key requirements. This document and the associated parts enable industry consolidation around a common, secure, interoperable approach.

## 1 Scope

This document defines security objectives, philosophy, Resources and mechanism that impacts OCF base layers of ISO/IEC 30118-1. ISO/IEC 30118-1 contains informative security content. The OCF Security Specification contains security normative content and may contain informative content related to the OCF base or other OCF documents.

## 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 30118-1 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 1: Core specification

<https://www.iso.org/standard/53238.html>

Latest version available at:

[https://openconnectivity.org/specs/OCF\\_Core\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Core_Specification.pdf)

ISO/IEC 30118-3 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 3: Bridging specification

<https://www.iso.org/standard/74240.html>

Latest version available at:

[https://openconnectivity.org/specs/OCF\\_Bridging\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Bridging_Specification.pdf)

OCF Wi-Fi Easy Setup, Information technology – Open Connectivity Foundation (OCF) Specification – Part 7: Wi-Fi Easy Setup specification

Latest version available at:

[https://openconnectivity.org/specs/OCF\\_Wi-Fi\\_Easy\\_Setup\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification.pdf)

OCF Cloud Specification, Information technology – Open Connectivity Foundation (OCF) Specification – Part 8: Cloud Specification

Latest version available at:

[https://openconnectivity.org/specs/OCF\\_Cloud\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Cloud_Specification.pdf)

OCF Cloud Security Specification - Open Connectivity Foundation (OCF) Specification – Cloud Security Specification

Latest version available at:

[https://openconnectivity.org/specs/OCF\\_Cloud\\_Security\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Cloud_Security_Specification.pdf)

OCF Onboarding Tool Specification - Open Connectivity Foundation (OCF) Specification – Onboarding Tool Specification

Latest version available at:

[https://openconnectivity.org/specs/OCF\\_Onboarding\\_Tool\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Onboarding_Tool_Specification.pdf)

OCF Cloud API for Cloud Services Specification - Open Connectivity Foundation (OCF) Cloud API for Cloud Services Specification

Latest version available at:

[https://openconnectivity.org/specs/OCF\\_Cloud\\_API\\_For\\_Cloud\\_Services\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Cloud_API_For_Cloud_Services_Specification.pdf) JSON SCHEMA, draft version 4, <http://json-schema.org/latest/json-schema-core.html>.

IETF RFC 2315, *PKCS #7: Cryptographic Message Syntax Version 1.5*, March 1998,

<https://tools.ietf.org/html/rfc2315>

IETF RFC 2898, *PKCS #5: Password-Based Cryptography Specification Version 2.0*, September 2000, <https://tools.ietf.org/html/rfc2898>



483 IETF RFC 2986, *PKCS #10: Certification Request Syntax Specification Version 1.7*, November  
484 2000, <https://tools.ietf.org/html/rfc2986>

485 IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, July 2005,  
486 <https://tools.ietf.org/html/rfc4122>

487 IETF RFC 4279, *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)*, December  
488 2005, <https://tools.ietf.org/html/rfc4279>

489 IETF RFC 4492, *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security*  
490 *(TLS)*, May 2006, <https://tools.ietf.org/html/rfc4492>

491 IETF RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2*, August 2008,  
492 <https://tools.ietf.org/html/rfc5246>

493 IETF RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation*  
494 *List (CRL) Profile*, May 2008, <https://tools.ietf.org/html/rfc5280>

495 IETF RFC 5489, *ECDHE\_PSK Cipher Suites for Transport Layer Security (TLS)*, March 2009,  
496 <https://tools.ietf.org/html/rfc5489>

497 IETF RFC 5545, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*,  
498 September 2009, <https://tools.ietf.org/html/rfc5545>

499 IETF RFC 5755, *An Internet Attribute Certificate Profile for Authorization*, January 2010,  
500 <https://tools.ietf.org/html/rfc5755>

501 IETF RFC 6347, *Datagram Transport Layer Security Version 1.2*, January 2012,  
502 <https://tools.ietf.org/html/rfc6347>

503 IETF RFC 6655, *AES-CCM Cipher Suites for Transport Layer Security (TLS)*, July 2012,  
504 <https://tools.ietf.org/html/rfc6655>

505 IETF RFC 7228, *Terminology for Constrained-Node Networks*, May 2014,  
506 <https://tools.ietf.org/html/rfc7228>

507 IETF RFC 7250, *Using Raw Public Keys in Transport Layer Security (TLS) and Datagram*  
508 *Transport Layer Security (DTLS)*, June 2014, <https://tools.ietf.org/html/rfc7250>

509 IETF RFC 7251, *AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS*, June 2014,  
510 <https://tools.ietf.org/html/rfc7251>

511 IETF RFC 7252, *The Constrained Application Protocol (CoAP)*, June 2014,  
512 <https://tools.ietf.org/html/rfc7252>

513 IETF RFC 8152, *CBOR Object Signing and Encryption (COSE)*, July 2017,  
514 <https://tools.ietf.org/html/rfc8152>

515 IETF RFC 8520, *Manufacturer Usage Description Specification*, Mar 2019,  
516 <https://tools.ietf.org/html/rfc8520>

517 IETF RFC 8613, *Object Security for Constrained RESTful Environments (OSCORE)*, July 2019,  
518 <https://tools.ietf.org/html/rfc8613>

519 oneM2M Release 3 Specifications, <http://www.onem2m.org/technical/published-drafts>

520 OpenAPI specification, aka *Swagger RESTful API Documentation Specification*, Version 2.0  
521 <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md>



## **3 Terms, definitions, and abbreviated terms**

### **3.1 Terms and definitions**

For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1, ISO/IEC 30118-3 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

#### **3.1.1**

##### **Access Management Service (AMS)**

service that dynamically constructs ACL Resources in response to a Device Resource request

Note 1 to entry: An AMS can evaluate access policies remotely and supply the result to a Server which allows or denies a pending access request. An AMS is authorised to provision ACL Resources.

#### **3.1.2**

##### **Credential Management Service (CMS)**

Device that is authorized to provision credential Resources

#### **3.1.3**

##### **Device Class**

IETF RFC 7228 defined device class

#### **3.1.4**

##### **Device Ownership Transfer Service (DOTS)**

logical entity that establishes device ownership

#### **3.1.5**

##### **End-Entity**

any certificate holder which is not a Root or Intermediate Certificate Authority

Note 1 to entry: Typically, a device certificate.

#### **3.1.6**

##### **Intermediary**

Device that implements both Client and Server roles and may perform protocol translation, virtual device to physical device mapping or Resource translation

#### **3.1.7**

##### **OCF Cipher Suite**

set of algorithms and parameters that define the cryptographic functionality of a Device. The OCF Cipher Suite includes the definition of the public key group operations, signatures, and specific hashing and encoding used to support the public key.

#### **3.1.8**

##### **OCF Rooted Certificate Chain**

collection of X.509 v3 certificates in which each certificate chains to a trust anchor certificate which has been issued by a certificate authority under the direction, authority, and approval of the Open Connectivity Foundation Board of Directors as a trusted root for the OCF ecosystem.

#### **3.1.9**

##### **Onboarding Tool (OBT)**

tool that implements *DOTS*(3.1.4), *AMS*(3.1.1), and *CMS*(3.1.2) functionality

#### **3.1.10**

##### **Out of Band Communication Channel**

any mechanism for delivery of a secret from one party to another, not specified by OCF

### 3.1.11

#### **Owner Credential (OC)**

credential, provisioned to a Device, for the purposes of mutual authentication of the Device and *OBT*(3.1.9) during subsequent interactions, identified by having a Subject UUID matching the Resource Owner Id of the Device Ownership Transfer Resource hosted by a Device that has the credential

### 3.1.12

#### **Role (Network context)**

stereotyped behavior of a Device; one of [Client, Server or Intermediary]

### 3.1.13

#### **Role Identifier**

Property of an OCF credentials Resource or element in a role certificate that identifies a privileged role that a Server Device associates with a Client Device for the purposes of making authorization decisions when the Client Device requests access to Device Resources.

### 3.1.14

#### **Secure Resource Manager (SRM)**

module in the OCF Core that implements security functionality that includes management of security Resources such as ACLs, credentials and Device owner transfer state.

### 3.1.15

#### **Security Virtual Resource (SVR)**

Resource supporting security features.

Note 1 to entry: For a list of all the SVRs please see clause 13.

### 3.1.16

#### **Trust Anchor**

well-defined, shared authority, within a trust hierarchy, by which two cryptographic entities (e.g. a Device and an *OBT*(3.1.9)) can assume trust

### 3.1.17

#### **Device Configuration Resource (DCR)**

Resource that is any of the following:

- a) a Discovery Core Resource, or
- b) a Security Virtual Resource, or
- c) a Wi-Fi Easy Setup Resource ("oic.r.easysetup", "oic.r.wificonf", "oic.r.devconf"), or
- d) a CoAP Cloud Configuration Resource ("oic.r.coapcloudconf"), or
- e) a Software Update Resource ("oic.r.softwareupdate"), or
- f) a Maintenance Resource ("oic.wk.mnt").

### 3.1.18

#### **Non-Configuration Resource (NCR)**

Resource that is not a Device Configuration Resource (3.1.17)

### 3.1.19

#### **OCF Security Domain**

set of onboarded OCF Devices that are provisioned with credentialing information for confidential communication with one another

### 3.1.20

#### **Owned (or "in Owned State")**

having the "owned" Property of the "/oic/sec/doxm" Resource equal to "TRUE"

614 **3.1.21**  
615 **Unowned (or "in Unowned State")**  
616 having the "owned" Property of the "/oic/sec/doxm" Resource equal to "FALSE"

617 **3.1.22**  
618 **OCF Onboarding**  
619 initial establishment of ownership over a Device, and initial provisioning of the Device for normal  
620 operation

621 **3.1.23**  
622 **Auditable Event**  
623 system activity that may be indicative of a violation of security policy

624 **3.1.24**  
625 **Auditable Event Entry**  
626 record of the details of an Auditable Event

627 **3.1.25**  
628 **End User**  
629 person using the [particular] product

630 **3.1.26**  
631 **End-to-End Secure**  
632 securely encapsulate information so that *OCF Proxies* (3.1.28) on the end-to-end delivery path do  
633 not need to be trusted with the confidentiality, integrity and freshness of that information

634 **3.1.27**  
635 **End-to-End Security of Unicast Messages**  
636 interoperable mechanism which End-to-End Secures the exchange of unicast OCF CRUDN  
637 messages

638 **3.1.28**  
639 **OCF Proxy**  
640 functionality which can interpret the OCF compliant URIs of request messages intended for  
641 resources on another OCF Server and can route those request messages accordingly

642 **3.1.29**  
643 **Origin Client**  
644 Client which originally generated a request, as opposed to the Client functionality of a Proxy which  
645 is forwarding a request from another Device

646 **3.1.30**  
647 **OSCORE Master Secret**  
648 "Master Secret" as defined in clause 3.1 of IETF RFC 8613

649 **3.1.31**  
650 **OSCORE Recipient ID**  
651 "Recipient ID" as defined in clause 3.1 of IETF RFC 8613

652 **3.1.32**  
653 **OSCORE Security Context**  
654 "Security Context" as defined in clause 3.1 of IETF RFC 8613

655 **3.1.33**  
656 **OSCORE Sender ID**  
657 "Sender ID" as defined in clause 3.1 of IETF RFC 8613

658 **3.1.34**  
659 **OSCORE Sender Sequence Number**  
660 "Sender Sequence Number" as defined in clause 3.1 of IETF RFC 8613

661 **3.1.35**  
662 **Target Server**  
663 Server to which a request is addressed, as opposed to the Server functionality of a *OCF Proxy*  
664 (3.1.28) which receives a request to be forwarded to another Device

665 **3.1.36**  
666 **Simple Secure Multicast**  
667 delivery of UPDATE request messages from a Client to a group of Servers using network-layer  
668 multicast, where the messages are protected with a simple security mechanism

669 **3.1.37**  
670 **Simple Secure Multicast Client Context**  
671 *OSCORE Security Context* (3.1.32) parameters provisioned to the Client of a *Simple Secure*  
672 *Multicast Group* (3.1.38) to enable End-to-End Security of *Simple Secure Multicast Requests*  
673 (3.1.39) sent to Servers of that *Simple Secure Multicast Group* (3.1.38)

674 **3.1.38**  
675 **Simple Secure Multicast Group**  
676 group of Servers and one (1) associated Client provisioned with credentials to enable *Simple*  
677 *Secure Multicast* (3.1.36) from the Client to the set of Servers

678 **3.1.39**  
679 **Simple Secure Multicast Request**  
680 OSCORE-protected UPDATE request message delivered from a Client to a group of Servers using  
681 *Simple Secure Multicast* (3.1.36)

682 **3.1.40**  
683 **Simple Secure Multicast Server Context**  
684 OSCORE Security Context parameters provisioned to Servers of a Simple Secure Multicast Group  
685 (3.1.38) to enable End-to-End Security of *Simple Secure Multicast Requests* (3.1.39) sent by the  
686 Client of that *Simple Secure Multicast Group* (3.1.38)

687 **3.1.41**  
688 **Device Onboarding Connection (DOC)**  
689 special DTLS connection established for the purposes of onboarding the Device securely when a  
690 Device is in RFOTM

691 NOTE: The Owner Transfer Method selected will determine the specifics of the DOC used.

692 **3.1.42**  
693 **Ready For Normal Operation State**  
694 state of a Device in which *NCRs* (3.1.18) can be accessed

695 **3.1.43**  
696 **Ready For Owner Transfer Mechanism State**  
697 state of a Device in which a Device can be Onboarded

698 **3.1.44**  
699 **Ready For Provisioning State**  
700 state of a Device in which *SVRs* (3.1.15) can be configured

|     |   |  |
|-----|---|--|
| 701 | <b>3.1.45</b>   |  |
| 702 | <b>Reset State</b>  |  |
| 703 | state of a Device in which the configurable Properties of Device's resources are reset to the       |  |
| 704 | manufacturer default and the Device becomes <i>Unowned</i> (3.1.21)                                 |  |
| 705 | <b>3.1.46</b>   |  |
| 706 | <b>Soft Reset State</b>   |  |
| 707 | state of a Device in which SVRs (3.1.15) can be configured, with slightly more Properties available |  |
| 708 | than in RFPRO   |  |
| 709 | <b>3.2</b>  | <b>Symbols and abbreviated terms</b>             |
| 710 | AC  | Access Control                                   |
| 711 | ACE   | Access Control Entry                             |
| 712 | ACL   | Access Control List                              |
| 713 | AEAD  | Authenticated Encryption with Authenticated Data |
| 714 | NOTE: Defined in IETF RFC 8152  |  |
| 715 | AEE   | Auditable Event Entry                            |
| 716 | AES   | Advanced Encryption Standard                     |
| 717 | AMS   | Access Management Service                        |
| 718 | CMS   | Credential Management Service                    |
| 719 | COSE  | CBOR Object Signing and Encryption               |
| 720 | NOTE: Defined in IETF RFC 8152  |  |
| 721 | CRUDN   | CREATE, RETREIVE, UPDATE, DELETE, NOTIFY         |
| 722 | CSR   | Certificate Signing Request                      |
| 723 | DOC   | Device Onboarding Connection                     |
| 724 | ECC   | Elliptic Curve Cryptography                      |
| 725 | ECDSA   | Elliptic Curve Digital Signature Algorithm       |
| 726 | EKU   | Extended Key Usage                               |
| 727 | DOTS  | Device Ownership Transfer Service                |
| 728 | ID  | Identity/Identifier                              |
| 729 | JSON  | JavaScript Object Notation.                      |
| 730 | NVRAM   | Non-Volatile Random-Access Memory                |
| 731 | OC  | Owner Credential                                 |
| 732 | OCSP  | Online Certificate Status Protocol               |
| 733 | OBT   | Onboarding Tool                                  |
| 734 | OID   | Object Identifier                                |

|     |                                |  |
|-----|--------------------------------|--|
| 735 | OSCORE                         | Object Security for Constrained RESTful Environments |
| 736 | NOTE: Defined in IETF RFC 8613 |  |
| 737 | OTM                            | Owner Transfer Method                                |
| 738 | PE                             | Policy Engine  |
| 739 | PIN                            | Personal Identification Number                       |
| 740 | PPSK                           | PIN-authenticated pre-shared key                     |
| 741 | PRF                            | Pseudo Random Function                               |
| 742 | PSI                            | Persistent Storage Interface                         |
| 743 | PSK                            | Pre Shared Key                                       |
| 744 | RBAC                           | Role Based Access Control                            |
| 745 | RM                             | Resource Manager                                     |
| 746 | RNG                            | Random Number Generator                              |
| 747 | RESET                          | Reset State  |
| 748 | RFNOP                          | Ready For Normal Operation State                     |
| 749 | RFOTM                          | Ready For Owner Transfer Mechanism State             |
| 750 | RFPRO                          | Ready For Provisioning State                         |
| 751 | SBAC                           | Subject Based Access Control                         |
| 752 | SEE                            | Secure Execution Environment                         |
| 753 | SRESET                         | Soft Reset State                                     |
| 754 | SRM                            | Secure Resource Manager                              |
| 755 | SSM                            | Simple Secure Multicast                              |
| 756 | SVR                            | Security Virtual Resource                            |
| 757 | URI                            | Uniform Resource Identifier                          |
| 758 | VOD                            | Virtual OCF Device                                   |

## 759 **4 Document Conventions and Organization**

### 760 **4.1 Conventions**

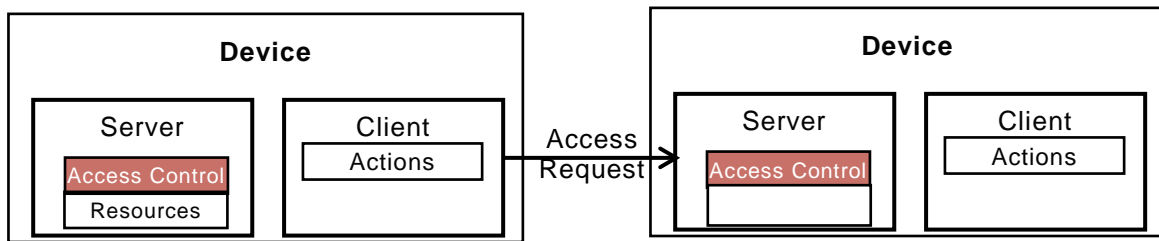
761 This document defines Resources, protocols and conventions used to implement security for OCF  
762 core framework and applications.

763 For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1 apply.

764 In this document, to be consistent with the IETF usages for RESTful operations, the RESTful  
765 operation words CRUDN, CREATE, RETRIVE, UPDATE, DELETE, and NOTIFY will have all letters  
766 capitalized. Any lowercase uses of these words have the normal technical English meaning.

767 Figure 1 depicts interaction between OCF Devices.





**Figure 1 – OCF Interaction**

Devices may implement a Client role that performs Actions on Servers. Actions access Resources managed by Servers. The OCF stack enforces access policies on Resources. End-to-end Device interaction can be protected using session protection protocol (e.g. DTLS) or with data encryption methods.

## 4.2 Notation

In this document, features are described as required, recommended, allowed or DEPRECATED as follows:

### **Required (or shall or mandatory).**

These basic features shall be implemented to comply with OCF Core Architecture. The phrases "shall not", and "PROHIBITED" indicate behaviour that is prohibited, i.e. that if performed means the implementation is not in compliance.

### **Recommended (or should).**

These features add functionality supported by OCF Core Architecture and should be implemented. Recommended features take advantage of the capabilities OCF Core Architecture, usually without imposing major increase of complexity. Notice that for compliance testing, if a recommended feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines. Some recommended features could become requirements in the future. The phrase "should not" indicates behaviour that is permitted but not recommended.

### **Allowed (may or allowed).**

These features are neither required nor recommended by OCF Core Architecture, but if the feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines.

### **Conditionally allowed (CA)**

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is allowed, otherwise it is not allowed.

### **Conditionally required (CR)**

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is required. Otherwise the definition or behaviour is allowed as default unless specifically defined as not allowed.

### **DEPRECATED**

Although these features are still described in this document, they should not be implemented except for backward compatibility. The occurrence of a deprecated feature during operation of an implementation compliant with the current document has no effect on the implementation's

802 operation and does not produce any error conditions. Backward compatibility may require that a  
803 feature is implemented and functions as specified but it shall never be used by implementations  
804 compliant with this document.

805 Strings that are to be taken literally are enclosed in "double quotes".

806 Words that are emphasized are printed in *italic*.

#### 807 **4.3 Data types**

808 See ISO/IEC 30118-1.

#### 809 **4.4 Document structure**

810 Informative clauses may be found in the Overview clauses, while normative clauses fall outside of  
811 those clauses.

812 The Security Specification may use the OpenAPI specification as the API definition language. The  
813 mapping of the CRUDN actions is specified in ISO/IEC 30118-1.

814

## 815

## 816

817  
818  
819  
820  
821  
822  
823  
824

825  
826  
827



829

- 830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842

843 3) The Client submits a request to the Server.

844 4) The Server receives the request.

845 a) If the request is received over an unsecured channel, the Server treats the request as

846 anonymous and no "deviceUUID" or "roleid" are associated with the request.

847 b) If the request is received over a secured channel, then the Server associates the request

848 with the "deviceUUID" of the Client and all valid "roleid" values of the Client by default.

849 c) The Server then consults the Access Control List (ACL), and looks for an Access Control

850 Entry (ACE) matching the following criteria:

851 i) The requested Resource matches a Resource reference in the ACE

852 ii) The requested operation is permitted by the "permissions" of the ACE, and

853 iii) The "subjectUUID" contains either one of a special set of wildcard values or, if the

854 Device is not anonymous, the subject matches the Client "deviceUUID" associated with

855 the request or a valid "roleid" associated with the request. The special wildcard values

856 authorize all Devices communicating over either authenticated and encrypted sessions

857 or unsecured sessions to interact according to the ACE.

858 If there is a matching ACE, then access to the Resource is permitted; otherwise access

859 is denied. Access is enforced by the Server's Secure Resource Manager (SRM).

860 5) The Server sends a response back to the Client.

861 OCF also supports exchange of messages between an Origin Client and Target Server facilitated

862 at one or more entities acting as OCF Proxies.

863 NOTE 1: Any number of OCF Proxies may be on the path between the Origin Client and Target Server, although this

864 number is expected to be small in practice.

865 In some scenarios, an OCF Proxy acts as a Server to incoming OCF CRUDN request messages:

866 processing the OCF CRUDN request messages; and then sending appropriate OCF CRUDN

867 request messages onwards towards the Target Server. The OCF Proxy can also process the

868 corresponding incoming OCF CRUDN response message and send appropriate OCF CRUDN

869 request messages back towards the Origin Client.

870 This approach implies that the owner of the Security Domain (containing the Origin Client and

871 Target Server) is willing to trust all OCF Proxies on the message delivery path with the

872 confidentiality, integrity and freshness of the OCF CRUDN messages. Alternatively, the Origin

873 Client and Target Server can apply End-to-End Security of Unicast Messages which enables

874 securing the exchange of OCF CRUDN messages so that OCF Proxies do not need to be trusted

875 with the confidentiality and integrity of the OCF CRUDN messages.

876 The security model of operation when using OCF Proxies without End-to-End Security of Unicast

877 Messages is described in OCF Cloud Specification, OCF Cloud Security Specification, and C2C

878 API.

879 Figure 3 and Figure 4 depict the security model of operation when using OCF Proxies and End-to-

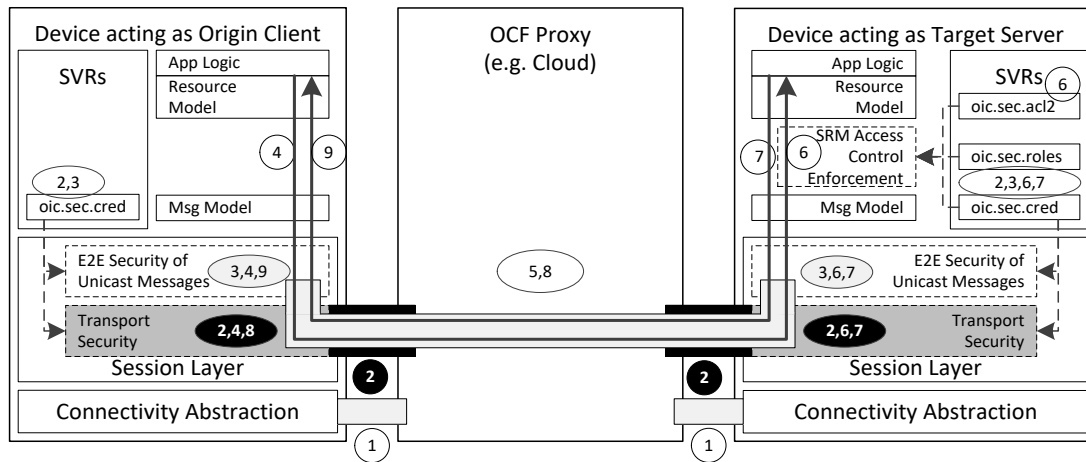
880 End Security of Messages is applied; see also the following steps. Figure 3 illustrates an example

881 with one OCF Proxy. Figure 4 illustrates a more complex example with two OCF Proxies using OCF

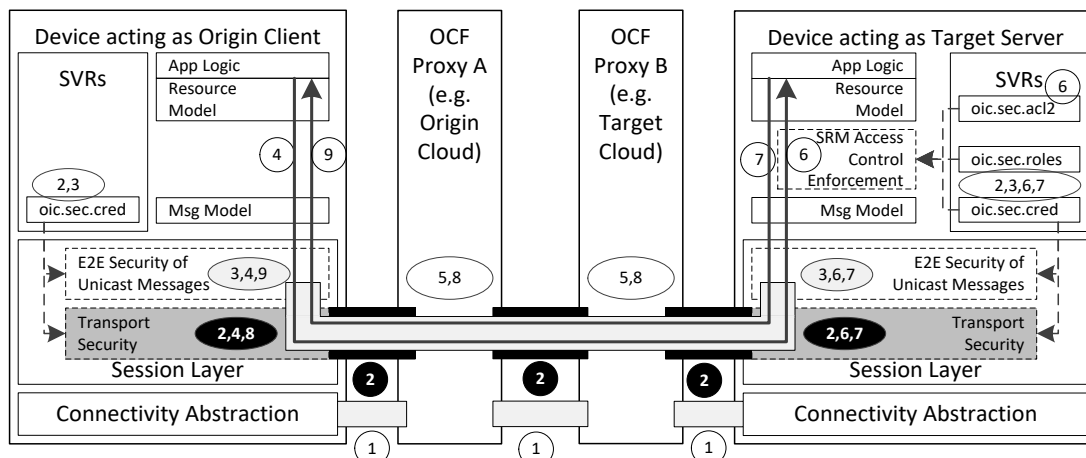
882 Cloud API for Cloud Services Specification; see notes 1 and 2.

883 NOTE 2: If the OCF Proxies in Figure 4 are OCF Clouds, OCF Proxy A is the Origin Cloud to which the Origin Client is

884 registered, and OCF Proxy B is the Target Cloud to which the Target Server is registered.



**Figure 3 – OCF Layers for interactions via one OCF Proxy**



**Figure 4 – OCF Layers for interactions via two OCF Proxies**

- 1) Pairwise network connections are established.
- 2) Messages are exchanged over each network connection via pairwise mutually-authenticated secure transport connection.
- 3) The Origin Client and Target Server establish an End-to-End Secured channel which is mutually-authenticated using credentials held in the "/oic/sec/cred" Resources of the Origin Client and Target Server.
- 4) The Origin Client generates an OCF CRUDN request message to the Target Server. The Origin Client encapsulates the OCF CRUDN request message into an End-to-End Secured request message of the End-to-End Secured channel (established in step 3). Information identifying the Target Server is left un-encrypted in the End-to-End Secured request message, so OCF Proxies can use the identifying information to route the End-to-End Secured request message correctly. The Origin Client sends the End-to-End Secured request message to its OCF Proxy, over the optionally secured transport connection established with that OCF Proxy. See Note 3.

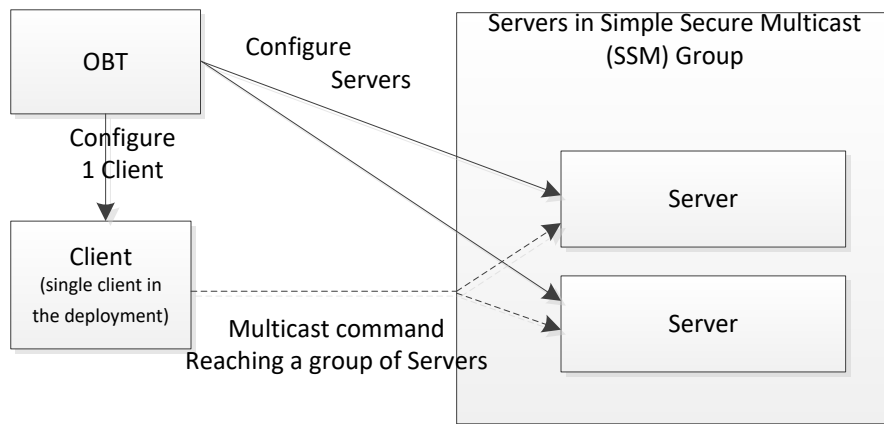
- 5) Each OCF Proxy on the path extracts the identifying information of the Target Server from the request message and, subject to the OCF Proxy's policies governing End-to-End Secured request messages, forwards the end-to- End-to-End Secured request message towards the Target Server over an optionally secured transport connection. See notes 3, 4 and 5.
- 6) The Target Server verifies and decrypts the End-to-End Secured request message as a message of the End-to-End Secured channel (established at step 3) to extract the encapsulated OCF CRUDN request message from the Origin Client. The OCF CRUDN request message is treated as being received over an authenticated encrypted ("auth-crypt") connection and associated with a "deviceUUID". The "deviceUUID" is associated with the credential in the "/oic/sec/cred" Resource used to establish the End-to-End Secured channel in step 3.
- 7) The Target Server determines whether access to the resource is permitted as described in step 4c of the Security model for direct Device-to-Device interaction shown in Figure 2.
- 8) The Target Server generates an OCF CRUDN response message and encapsulates the OCF CRUDN response message into an End-to-End Secured response message of the End-to-End Secured channel (established at step 3). The Target Secure sends the End-to-End Secured response message to its OCF Proxy, over the optionally secured transport connection on which the corresponding request was received. See Note 3.
- 9) Each OCF Proxy on the path forwards the End-to-End Secured response message towards the Origin Client over the optionally secured transport connection on which the corresponding request message was received. See Note 3.
- 10) The Origin Client verifies and decrypts the End-to-End Secured response message as a message of the End-to-End Secured channel (established at step 3) to extract the encapsulated OCF CRUDN response message from the Target Server.

NOTE 3: While in transit, the OCF CRUDN message might be secured by up to two independent layers of Security: a layer of End-to-End Security of Unicast Messages (using OSCORE), and an independent layer of transport Security (using DTLS or TLS).

NOTE 4: This document does not address details of how an OCF Proxy determines if its policies permit forwarding the request message towards the identified Target Server. If an OCF Proxy permits forwarding a request message towards a Target Server, then it is assumed that the OCF Proxy also permits forwarding the corresponding response message(s) over the transport connection on which the corresponding request message was received.

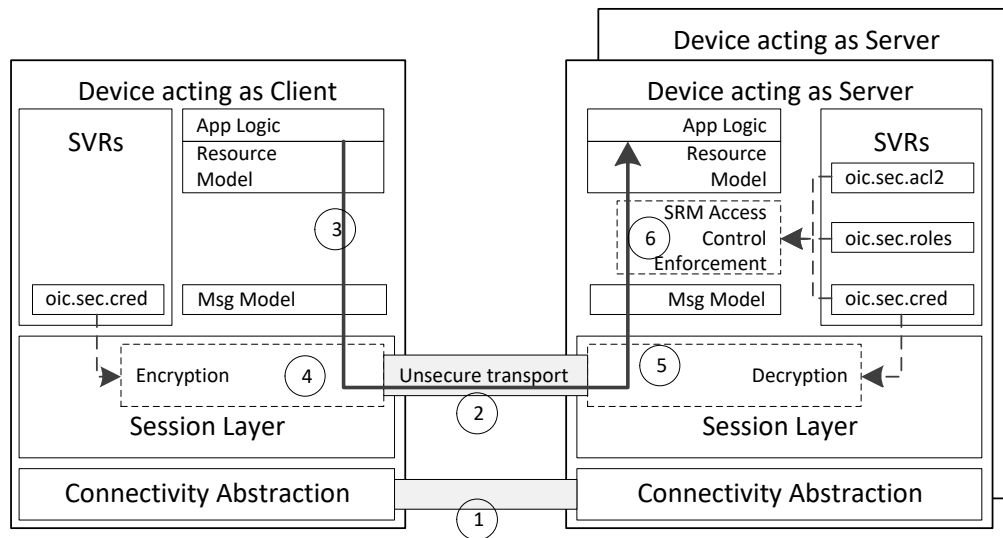
NOTE 5: This document does not address how OCF Proxy A determines that OCF Proxy B is the correct OCF Proxy to forward the request message to. The OCF Cloud API for Cloud Services Specification provides the details for the case where the OCF Proxy A and OCF Proxy B are OCF Clouds.

As shown in Figure 5, Simple Secure Multicast (SSM) enables a Client to securely communicate an UPDATE request to a group of Servers with a single non-confirmable UPDATE request delivered via networking-layer multicast.



**Figure 5 – Single request reaches a group of Servers**

The Security model for SSM is described in Figure 6 and the accompanying steps.



**Figure 6 – OCF Layers for Simple Secure Multicast**

- 1) The Client and Servers in the SSM Group are configured with encryption/decryption. The Client knows the preconfigured multicast address to use and how to create the actual payload of the command to send.
- 2) Messages are exchanged over an unsecure transport connection.
- 3) The Client generates an UPDATE request message to the Servers.
- 4) The Client encapsulates the UPDATE request message into an End-to-End Secured request message of the unsecured channel. The multicast address is left unencrypted in the Secured request message.

The Client sends the Secured UPDATE request message to the multicast URL of the Servers, using the URL of the multicast enabled resource.

5) The Servers decrypt the message. The UPDATE request message is treated as being received over an authenticated encrypted ("auth-crypt") connection and associated with a "deviceUUID" (which can be the Device UUID of the Client).

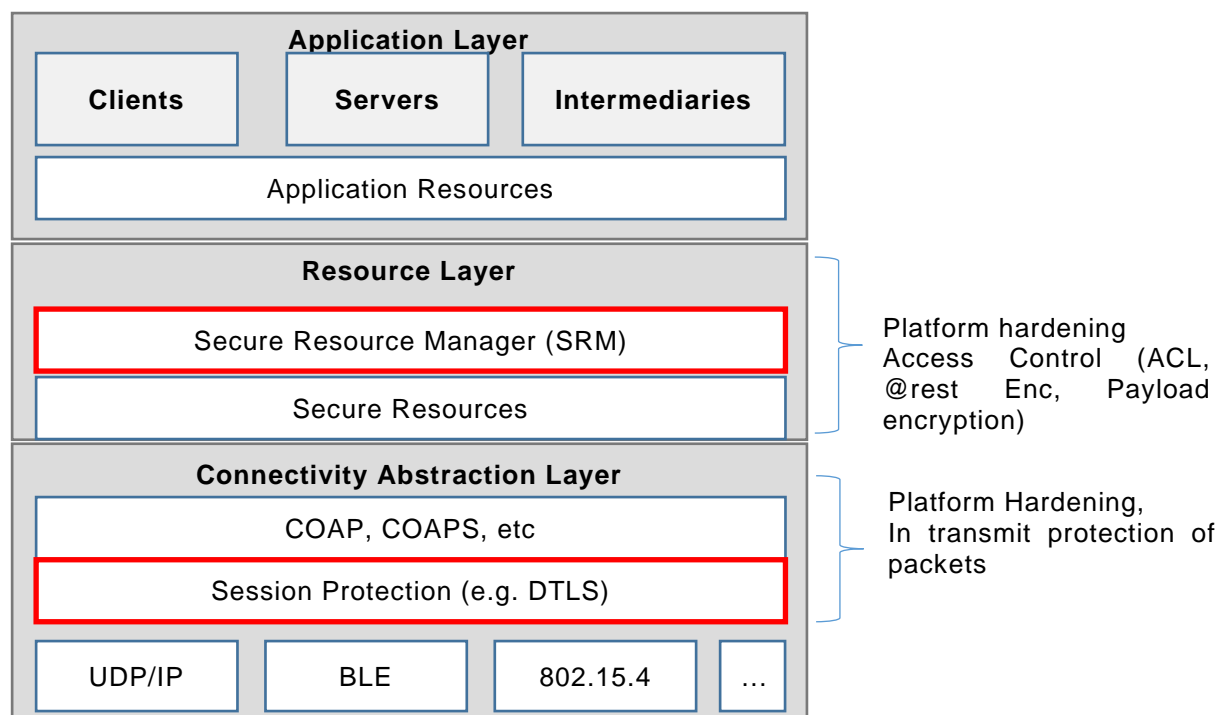
6) The Server determines whether access to the Resource is permitted as described in step 4c of the Security model for direct Device-to-Device interaction shown in Figure 2.

Resource protection includes protection of data both while at rest and during transit. Aside from access control mechanisms, the OCF Security Specification does not include specification of secure storage of Resources. Secure storage may be accomplished through the use of hardware security or encryption of data at rest. The exact implementation of secure storage is subject to a set of hardening requirements that are specified in clause 14 and may be subject to certification guidelines.

Data in transit protection is specified fully as a normative part of this document. This document supports data in transit data protection at the transport layer through use of mechanisms such as DTLS and end-to-end data-in-transit protection through OSCORE.

NOTE 6: DTLS will provide packet by packet protection, rather than protection for the OCF CRUDN message as whole. For instance, if the integrity of the entire OCF CRUDN message as a whole is required, separate end-to-end Security (for example, using OSCORE) should be applied before passing the packet down to the transport layer.

Figure 7 depicts OCF Security Enforcement Points.



**Figure 7 – OCF Security Enforcement Points**

## 5.2 Access Control

### 5.2.1 Access Control General

The OCF framework assumes that Resources are hosted by a Server and are made available to Clients subject to access control and authorization mechanisms. The Resources at the Server are protected through implementation of access control, authentication and confidentiality protection.



979 This clause provides an overview of access control through the use of Access Control Lists.  
980 However, access control in OCF is agnostic regarding transport and connectivity abstraction layers.

981 Implementation of access control relies on a-priori definition of a set of access policies for the  
982 Resource. The policies are stored locally in an ACL Resource provisioned by an Access  
983 Management Service (AMS) in the form of Access Control Entries (ACE). The lack of such an  
984 associated ACE results in the Resource being inaccessible. Multiple types of access control  
985 mechanisms may be applied:

- 986 – Subject-based access control (SBAC), where the ACE matches the identity of the Client against  
987 the subject included in the policy defined for the Resource. Asserting the identity of the Client  
988 requires an authentication process.
- 989 – Role-based Access Control (RBAC), where the ACE matches a role identifier included in the  
990 policy for the Resource to a role identifier associated with the Client.
- 991 – Wildcard-based Access Control, where the ACE matches a connection type, used to access the  
992 Resource (i.e. any mutually-authenticated connection).

993  
994 The ACE only applies if the ACE matches both the subject (i.e. Client) and the requested Resource.  
995 There are multiple ways a subject could be matched, (1) Device UUID, (2) Role Identifier or (3)  
996 wildcard. The way in which the Client connects to the Server may be relevant for making access  
997 control decisions. Wildcard matching on authenticated vs. unauthenticated and encrypted vs.  
998 unencrypted connection allows an access policy to be broadly applied to subject classes.

999 Example Wildcard Matching Policy:

```
1000 "aclist2": [  
1001   {  
1002     "subject": {"conntype" : "anon-clear" },  
1003     "resources": [  
1004       { "wc": "*" }  
1005     ],  
1006     "permission": 31  
1007   },  
1008   {  
1009     "subject": {"conntype" : "auth-crypt" },  
1010     "resources": [  
1011       { "wc": "*" }  
1012     ],  
1013     "permission": 31  
1014   },  
1015 ]
```

1016 Details of the format for ACL are defined in clause 12. The ACL is composed of one or more ACEs.

1017 Some Resources, such as Collections, generate requests to linked Resources when appropriate  
1018 Interfaces are used. In such cases, additional access control considerations are necessary.  
1019 Additional access control considerations for Collections when using the batch OCF Interface are  
1020 found in clause 12.2.7.3. ACL Resource requires the same security protection as other sensitive  
1021 Resources when it comes to both storage and handling by the SRM.

**5.2.2 ACL Architecture**

The Server examines the Resource(s) requested by the client before processing the request. The access control Resource is searched to find one or more ACE entries that match the Client and the requested Resources. If a match is found, then permission and period constraints are applied. If more than one match is found, then each ACE entry is evaluated for a match independently.

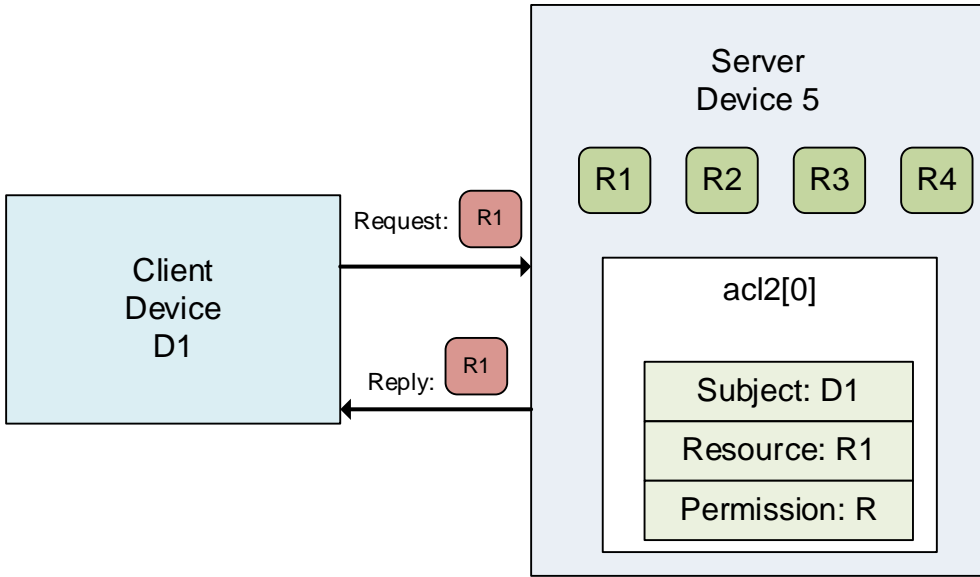
The Server uses the connection context to determine whether the subject has authenticated or not and whether data confidentiality has been applied or not. If the user has authenticated, then subject matching may happen at increased granularity based on role or device identity.

Each ACE contains the permission set that will be applied for a given Client. Permissions consist of a combination of CREATE, RETREIVE, UPDATE, DELETE and NOTIFY (CRUDN) actions. Clients authenticate as a Device and optionally operating with one or more roles. Devices may acquire elevated access permissions when asserting a role. For example, an "oic.role.owner" role might expose additional Resources and OCF Interfaces not normally accessible.

Servers host ACL Resources locally. Local ACLs allow greater autonomy in access control processing.

The following use cases describe the operation of access control:

Use Case 1: As depicted in Figure 8, Server Device hosts 4 Resources (R1, R2, R3 and R4). Client Device D1 requests access to Resource R1 hosted at Server Device 5. ACL[0] corresponds to Resource R1 and includes D1 as an authorized subject. Thus, Device D1 receives access to Resource R1 because the local ACL "/oic/sec/acl2/0" matches the request.



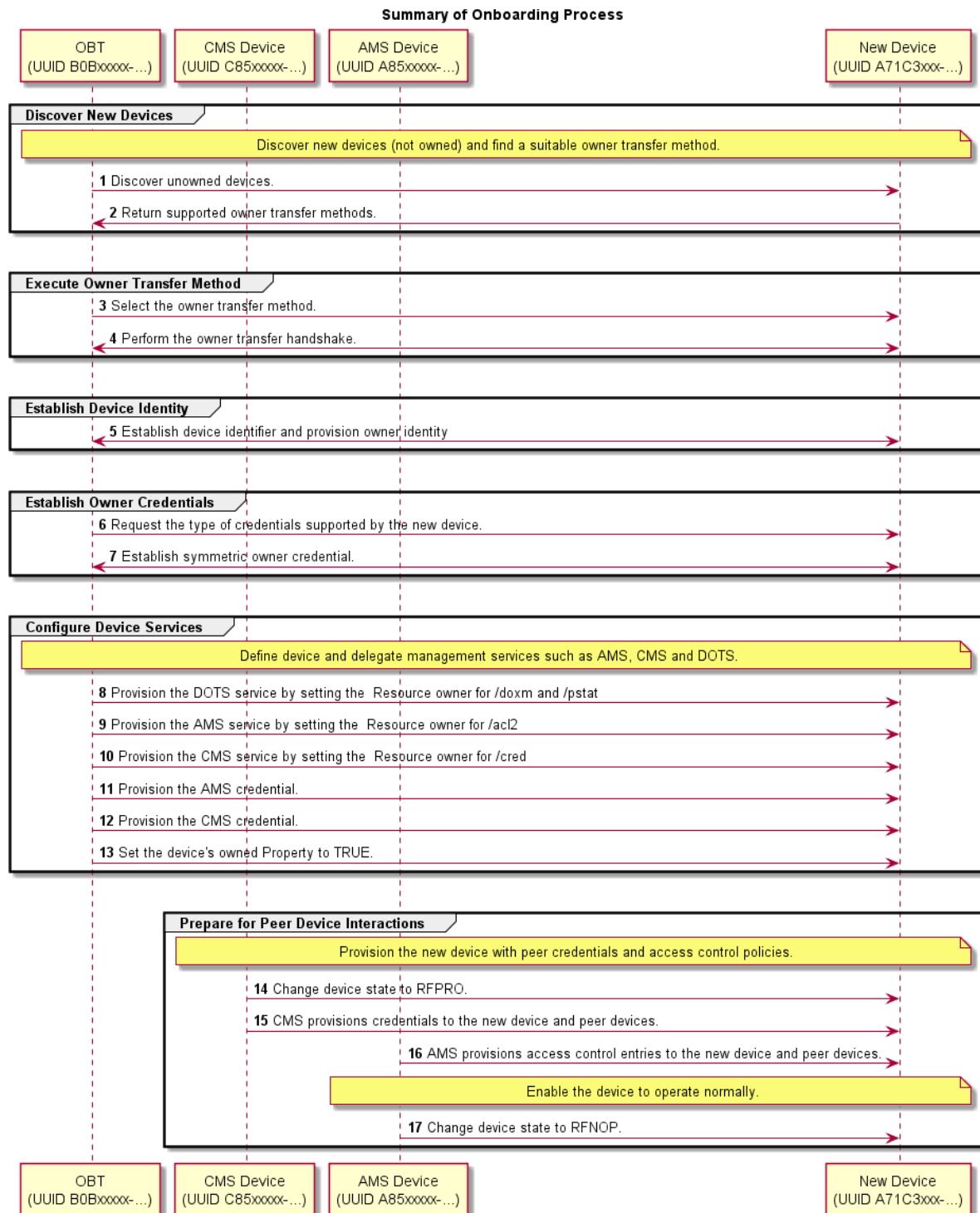
**Figure 8 – Use case-1 showing simple ACL enforcement**

### 1045 **5.3 Onboarding Overview**

#### 1046 **5.3.1 Onboarding General**

1047 Before a Device becomes operational in an OCF environment and is able to interact with other  
1048 Devices, it needs to be appropriately onboarded. The first step in onboarding a Device is to  
1049 configure the ownership where the legitimate user that owns/purchases the Device uses an  
1050 Onboarding tool (OBT) and using the OBT uses one of the Owner Transfer Methods (OTMs) to  
1051 establish ownership. Once ownership is established, the OBT provisions the Device, at the end of  
1052 which the Device becomes operational and is able to interact with other Devices in an OCF  
1053 environment.

1054 Figure 9 depicts an overview of Onboarding.

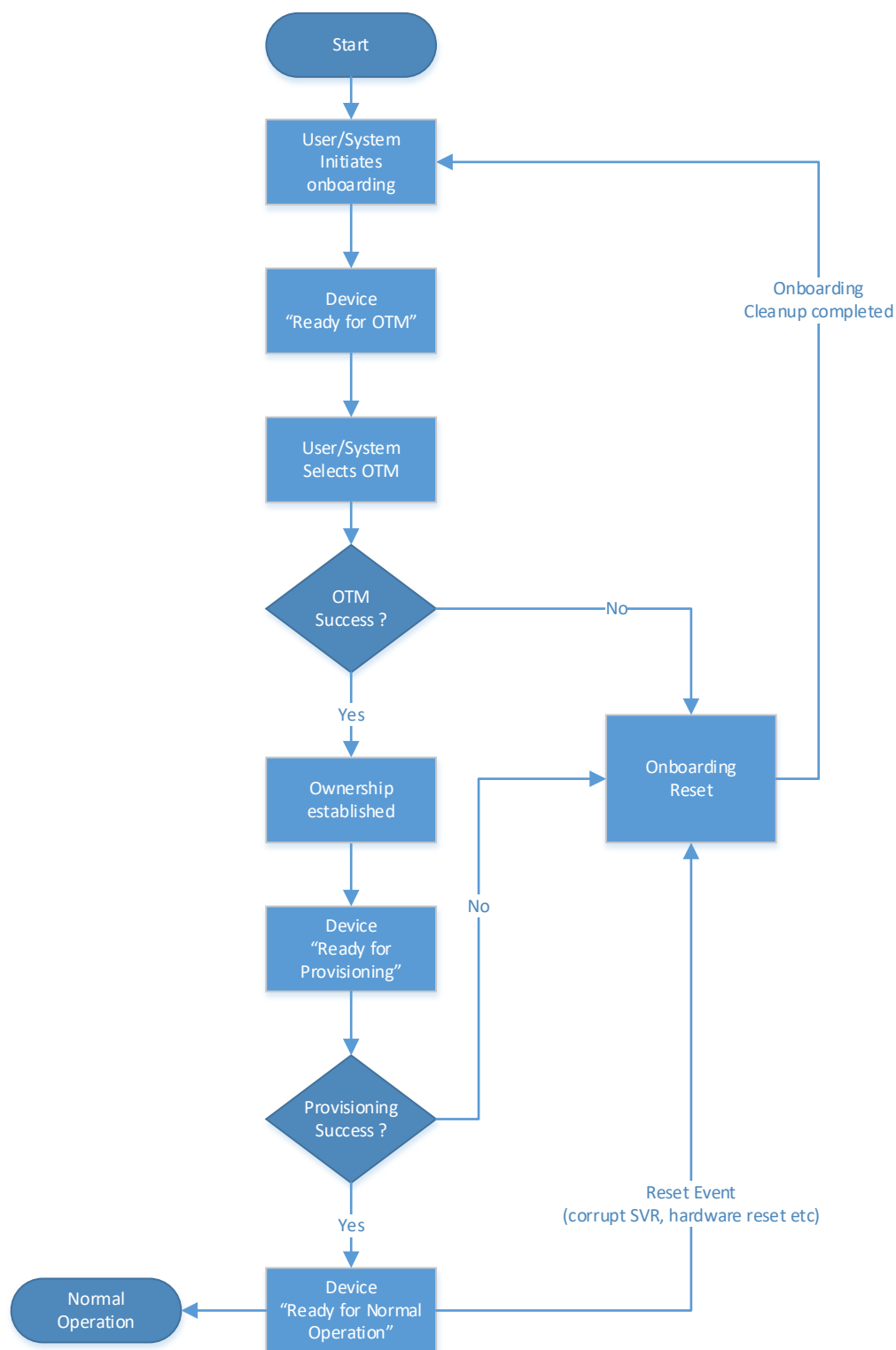


**Figure 9 – Onboarding overview**

This clause explains the onboarding and security provisioning process but leaves the provisioning of non-security aspects to other OCF documents. In the context of security, all Devices are required to be provisioned with minimal security configuration that allows the Device to securely interact/communicate with other Devices in an OCF environment. This minimal security configuration is defined as the Onboarded Device RFNOP and is specified in 8.

### 5.3.2 Onboarding Steps

The flowchart in Figure 10 shows the typical steps that are involved during onboarding. Although onboarding may include a variety of non-security related steps, the diagram focus is mainly on the security related configuration to allow a new Device to function within an OCF environment. Onboarding typically begins with the Device becoming an Owned Device followed by configuring the Device for the environment that it will operate in. This would include setting information such as who may access the Device and what actions may be performed as well as what permissions the Device has for interacting with other Devices.



**Figure 10 – OCF Onboarding Process**

### 5.3.3 Establishing a Device Owner

The objective behind establishing Device ownership is to allow the OCF Security Domain Owner to assert itself as the owner and manager of the Device and introduce the Device into the OCF Security Domain. This is done through the use of a DOTS that includes the creation of an ownership

Copyright Open Connectivity Foundation, Inc. © 2016-2020. All rights Reserved

1076 context between the new Device and the DOTS and asserts operational control and management  
1077 of the Device. The DOTS is hosted on an OBT.

1078 The DOTS uses one of the OTMs specified in 7.3 to securely establish Device ownership.

1079 An OTM establishes a new owner (the operator of DOTS) that is authorized to manage the Device.  
1080 Ownership Transfer accomplishes the following:

- 1081 – The DOTS provisions an Owner Credential (OC) to the "creds" Property in the "/oic/sec/cred"  
1082 Resource of the Device. This OC allows the Device and DOTS to mutually authenticate during  
1083 subsequent interactions. The OC associates the DOTS Device UUID with the "rowneruuid"  
1084 Property of the "/oic/sec/doxm" Resource establishing it as the Resource owner.
- 1085 – The Device owner establishes trust in the Device through the OTM.
- 1086 – Provisioning of appropriate credentials for the Device to be a member of the OCF Security  
1087 Domain.

#### 1088 **5.3.4 Provisioning for Normal Operation**

1089 Once the Device has the necessary information to initiate provisioning, the next step is to provision  
1090 additional security configuration that allows the Device to become operational. This may include  
1091 setting various parameters and may also involve multiple steps. Also provisioning of ACL's for the  
1092 various Resources hosted by the Server on the Device is done at this time. The provisioning step  
1093 is not limited to this stage only. Device provisioning may happen at multiple stages in the Device's  
1094 operational lifecycle. However specific security related provisioning of Resource and Property state  
1095 would likely happen at this stage at the end of which, each Device reaches RFNOP. RFNOP is  
1096 consistent and well defined regardless of the specific OTM used or regardless of the variability in  
1097 what gets provisioned. However individual OTM mechanisms and provisioning steps may specify  
1098 additional configuration of Resources and Property states. The minimal mandatory configuration  
1099 required for a Device to be in RFNOP is specified in 8.

#### 1100 **5.3.5 OCF Compliance Management System**

1101 The OCF Compliance Management System (OCMS) is a service maintained by the OCF that  
1102 provides Certification status and information for OCF Devices.

1103 The OCMS shall provide a JSON-formatted Certified Product List (CPL), hosted at the URI:  
1104 <https://www.openconnectivity.org/certification/ocms-cpl.json>

1105 The OBT shall possess the Root Certificate needed to enable https connection to the URI  
1106 <https://www.openconnectivity.org/certification/ocms-cpl.json>.

1107 The OBT should periodically refresh its copy of the CPL via the URI  
1108 <https://www.openconnectivity.org/certification/ocms-cpl.json>, as appropriate to OCF Security  
1109 Domain owner policy requirements.

### 1110 **5.4 Provisioning**

#### 1111 **5.4.1 Provisioning General**

1112 OCF security provisioning includes processes during and after the ownership transfer like  
1113 configuration of credentials for interacting with provisioning services, configuration of any security  
1114 related Resources and credentials for interacting with any services or Devices that the provisioned  
1115 Device needs to contact later on.

1116 The Device needs to engage with the CMS and AMS to be provisioned with:

- 1117 – Security credentials through a CMS, which is currently assumed to be embedded in the same  
1118 OBT as the DOTS.
- 1119 – Access control policies and ACLs through an AMS, which is currently assumed to be embedded  
1120 in the same OBT as the DOTS.

1121 To be able to support the use of distinct device management services, some Device Secure Virtual  
1122 Resources (SVRs) have an associated Resource owner identified in the Resource's rowneruuid  
1123 Property.

1124 The "rowneruuid" Property of the "/oic/sec/doxm" and "/oic/sec/pstat" Resources identifies the  
1125 DOTS.

1126 The "rowneruuid" Property of the "/oic/sec/cred" Resource identifies the CMS.

1127 The "rowneruuid" Property of the "/oic/sec/acl2" Resource identifies the AMS.

1128 The DOTS provisions credentials that enable secure connections between OCF Services and the  
1129 new Device. The DOTS initiates client-directed provisioning by signaling the OCF Service.

#### 1130 **5.4.2 Access Control Provisioning**

1131 ACL provisioning is performed over a secure connection between the AMS and its Devices. The  
1132 AMS provisions the ACL by updating the Device's ACL Resource.

#### 1133 **5.4.3 Credential Provisioning**

1134 The CMS securely provisions credentials for Device-to-Device interactions using the CMS  
1135 credential provisioned by the DOTS during the onboarding procedure. The CMS is also expected  
1136 to proactively monitor the credentials installed on the Device and update them when needed (e.g.  
1137 close to the expiration date).

#### 1138 **5.4.4 Role Provisioning**

1139 The Servers, receiving requests for Resources they host, need to verify the role identifier(s)  
1140 asserted by the Client requesting the Resource and compare that role identifier(s) with the  
1141 constraints described in the Server's ACLs. Thus, a Client may need to be provisioned with one or  
1142 more role credentials. Once provisioned, the Client can assert the role it is using as described in  
1143 10.4.2, if it has a certificate role credential.

1144 Each Device holds the assertable role(s) information as a Property within the Credential Resource.  
1145 Each Device holds the asserted role(s) information as Properties within the Roles Resource.

1146 All asserted roles are used in ACL enforcement. When a server has multiple roles asserted for a  
1147 Client, access to a Resource is granted if it would be granted under any of the roles.

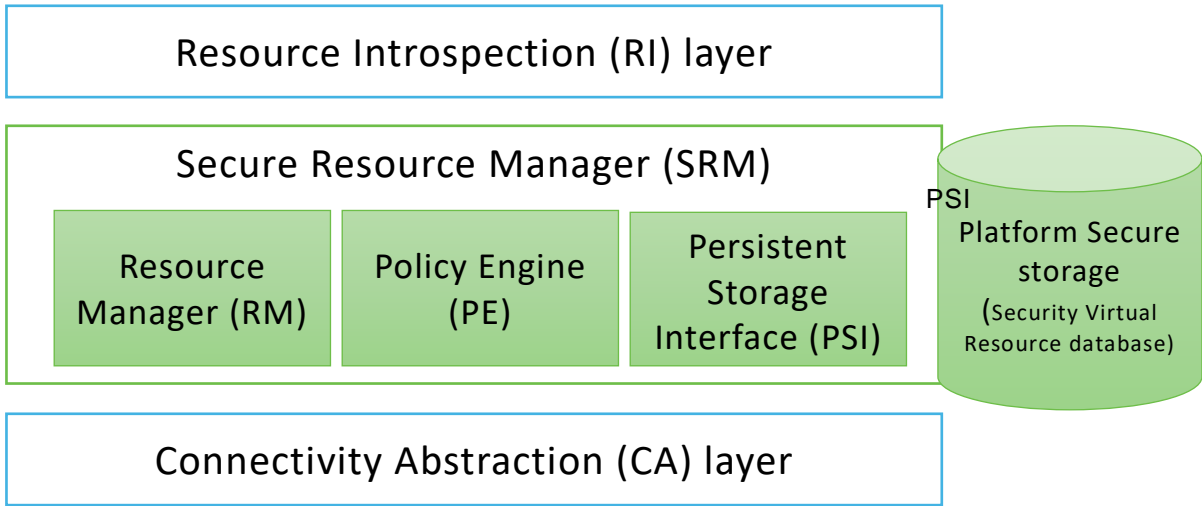
### 1148 **5.5 Secure Resource Manager (SRM)**

1149 SRM plays a key role in the overall security operation. In short, SRM performs both management  
1150 of SVR and access control for requests to access and manipulate Resources. SRM consists of 3  
1151 main functional elements:

- 1152 – A Resource manager (RM): responsible for 1) Loading SVRs from persistent storage (using PSI)  
1153 as needed. 2) Supplying the Policy Engine (PE) with Resources upon request. 3) Responding  
1154 to requests for SVRs. While the SVRs are in SRM memory, the SVRs are in a format that is  
1155 consistent with device-specific data store format. However, the RM will use JSON format to  
1156 marshal SVR data structures before being passed to PSI for storage, or travel off-device.
- 1157 – A Policy Engine (PE) that takes requests for access to SVRs and based on access control  
1158 policies responds to the requests with either "ACCESS\_GRANTED" or "ACCESS\_DENIED". To  
1159 make the access decisions, the PE consults the appropriate ACL and looks for best Access  
1160 Control Entry (ACE) that can serve the request given the subject (Device or role) that was  
1161 authenticated by DTLS.
- 1162 – Persistent Storage Interface (PSI): PSI provides a set of APIs for the RM to manipulate files in  
1163 its own memory and storage. The SRM design is modular such that it may be implemented in  
1164 the Platform's secure execution environment; if available.



Figure 11 depicts OCF's SRM Architecture.



**Figure 11 – OCF's SRM Architecture**

## 5.6 Credential Overview

Devices may use credentials to prove the identity and role(s) of the parties in the Client to Server communication. Credentials may be symmetric or asymmetric. Each Device stores secret and public parts of its own credentials where applicable, as well as credentials for other Devices that have been provisioned by the DOTS or a CMS. These credentials may then be used in the establishment of secure communication sessions (e.g. using DTLS, TLS or OSCORE). Role certificates may be used after an authenticated session is established to assert one or more roles for a Device.

The credential types available within this document include:

- Pairwise symmetric keys
- Certificates
- Raw asymmetric keys

Devices may not support all of these credential types. The set of supported credential types for any Device is contained in its "sct" Property of the "/oic/sec/doxm" Resource.

## 5.7 Event Logging

### 5.7.1 Event Logging General

An OCF Platform can generate various kinds of Auditable Events. These Auditable Events can be used for log analysis or for real-time understanding of a system condition. Usually multiple Auditable Events are stored to backtrack problems that have occurred in the system. The storage capacity of IoT devices is typically very limited, so a specific type of data structure such as a ring buffer is often used.

An OCF Device logs Auditable Event Entries (AEE) for all Auditable Events that satisfy the "categoryfilter" and "priorityfilter" Properties of the "/oic/sec/ael" Resource. The AEEs are stored in local storage (see Figure 1). Due to the limited size of the local storage, OCF Security Domain Owner is expected to adjust the filtering options.



**Figure 12 – Store Events in local storage**

## 5.8 End-to-End Security of Unicast Messages

The Security model for End-to-End Security of Unicast Messages is described in Figure 3 and Figure 4 of clause 5.1 and the accompanying steps.

OCF uses the Object Security for Constrained RESTful Environments (OSCORE) protocol IETF RFC 8613 for End-to-End Security of Unicast Messages. The Origin Client transforms a CoAP-encoded OCF CRUDN request message into an OSCORE request message which can be forwarded towards the Target Server by OCF Proxies; the Target Server then processes the OSCORE request message to extract the OCF CRUDN request message. Likewise, the Target Server then transforms a CoAP-encoded OCF CRUDN response message into an OSCORE response message which can be forwarded towards the Origin Client by OCF Proxies; the Origin Client then processes the OSCORE response message to extract the OCF CRUDN response message. OSCORE preserves the confidentiality, integrity and freshness of the OCF CRUDN messages while in transit between the Origin Client and the Target Server.

OSCORE specification supports transporting OSCORE messages using the CoAP protocol already used in OCF specifications. The payload of the OSCORE message is a CBOR Object Signature and Encryption (COSE) object (see IETF RFC 8152) in which all elements of the CoAP-encoded OCF CRUDN message, other than those parts which are needed for delivering the message to the receiving Device, are encrypted and integrity protected. OSCORE also includes replay protection.

## 5.9 Overview of Simple Secure Multicast

The Security model for SSM is described in Figure 6 of clause 5.1 and the accompanying steps. OCF uses the OSCORE protocol IETF RFC 8613 for the Security of SSM Messages. The Client transforms a CoAP-encoded UPDATE request message into an OSCORE request message which can be forwarded towards the Servers of the SSM Group using network-layer multicast; the Server then processes the OSCORE request message to extract the UPDATE request message.

Note: OSCORE is also used, albeit slightly differently, for End-to-End Security of Unicast Messages.

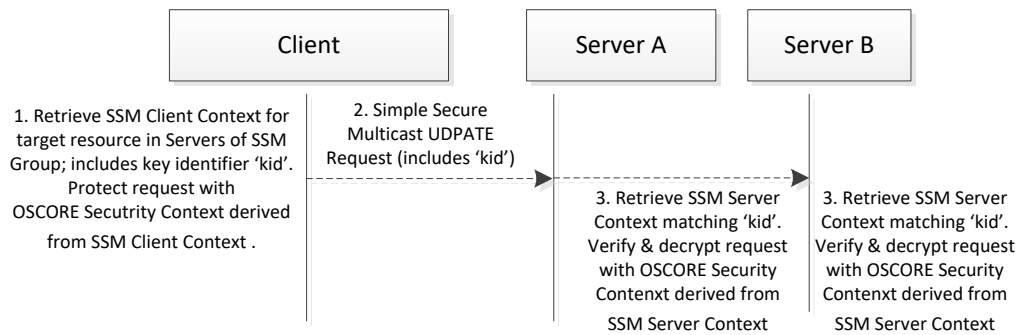
The intended use of the SSM feature is only for updating Resources with one non-confirmable multicast request. Other CRUDN operations (e.g. RETRIEVE, confirmable UPDATE, etc) are not supported because the SSM protocol is not designed to send individual responses back on the request. Hence when sending such operation by means of SSM, the individual Servers will silently ignore the request message and not send a response.

The OSCORE specification supports transporting OSCORE messages using the CoAP protocol already used in OCF specifications. The payload of the OSCORE message is a CBOR Object Signing and Encryption (COSE) object (see IETF RFC 8152) in which all elements of the CoAP-encoded UPDATE request message, other than those parts which are needed for delivering the message to the receiving Device, are encrypted and integrity protected. OSCORE also includes replay protection.

The setup of the OSCORE security context for an SSM Group is a 1-N relationship:

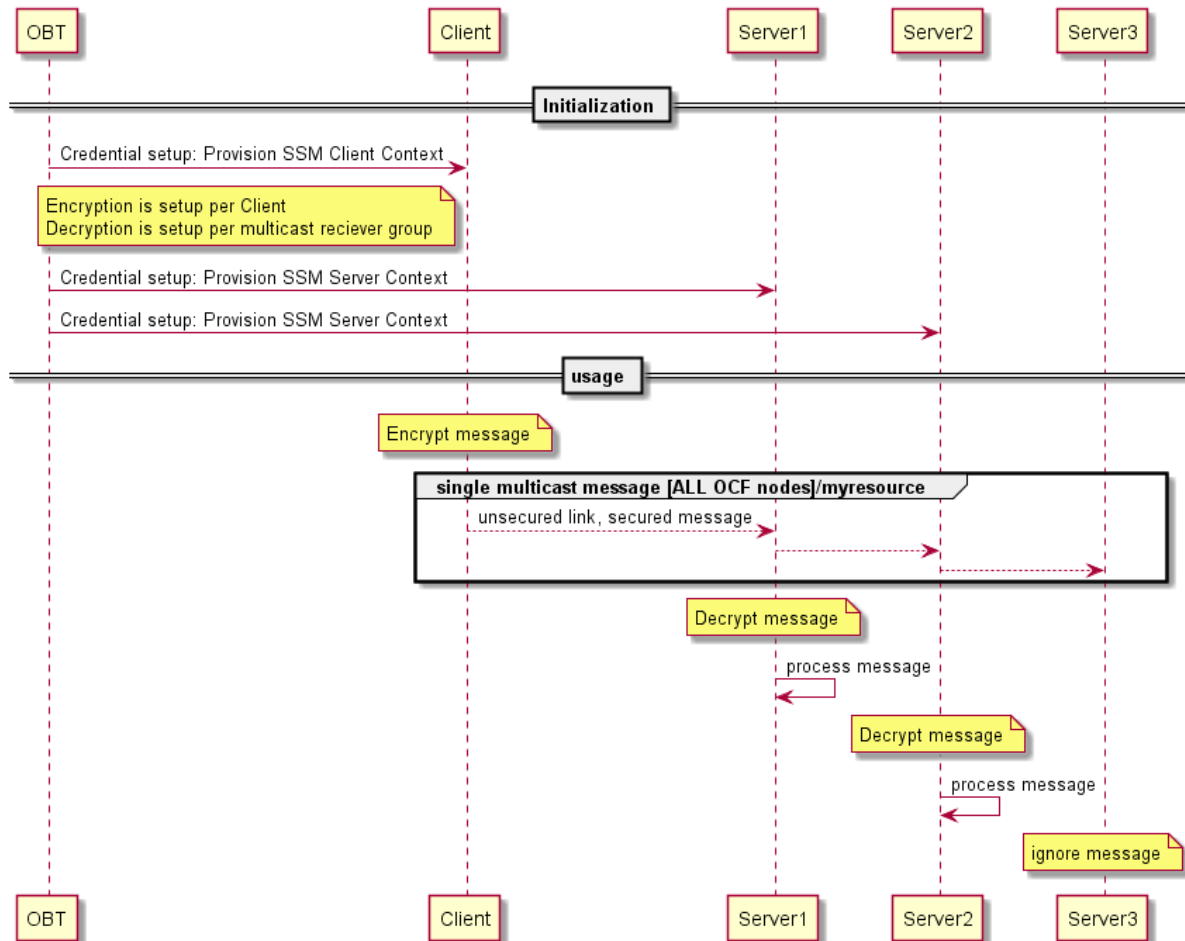
- the SSM Client Context of the SSM Group is only provisioned once in the Client of the SSM Group, and
- copies of the SSM Server Context of the SSM Group are provisioned to one or more Servers in the SSM Group.

Figure 13 depicts the relationship of the SSM Client Context and SSM Server Context.



**Figure 13 – Relationship diagram for Simple Secure Multicast messages**

Figure 14 depicts the full setup and usage.



**Figure 14 – Setup and usage of Secure Simple Multicast**

The first message after onboarding is implicitly trusted by the Server as being a valid message. This is due to the replay window not yet being set up by the Server. The Server stores the received information so that the replay protection is enabled after receiving the first message.

## 6 Security for the Discovery Process

### 6.1 Preamble

The main function of a discovery mechanism is to provide Universal Resource Identifiers (URIs, called links) for the Resources hosted by the Server, complemented by attributes about those Resources and possible further link relations. (in accordance to clause 10 in ISO/IEC 30118-1)

### 6.2 Security Considerations for Discovery

When defining discovery process, care must be taken that only a minimum set of Resources are exposed to the discovering entity without violating security of sensitive information or privacy requirements of the application at hand. This includes both data included in the Resources, as well as the corresponding metadata.

To achieve extensibility and scalability, this document does not provide a mandate on discoverability of each individual Resource. Instead, the Server holding the Resource will rely on ACLs for each Resource to determine if the requester (the Client) is authorized to see/handle any of the Resources.

The `"/oic/sec/acl2"` Resource contains ACL entries governing access to the Server hosted Resources. (See 13.5)

Aside from the privacy and discoverability of Resources from ACL point of view, the discovery process itself needs to be secured. This document sets the following requirements for the discovery process:

- 1) Providing integrity protection for discovered Resources.
- 2) Providing confidentiality protection for discovered Resources that are considered sensitive.

The discovery of Resources is done by doing a RETRIEVE operation (either unicast or multicast) on the known `"/oic/res"` Resource.

The discovery request is sent over a non-secure channel (multicast or unicast without DTLS), a Server cannot determine the identity of the requester. In such cases, a Server that wants to authenticate the Client before responding can list the secure discovery URI (e.g. `coaps://IP:PORT/oic/res`) in the unsecured `"/oic/res"` Resource response. This means the secure discovery URI is by default discoverable by any Client. The Client will then be required to send a separate unicast request using DTLS to the secure discovery URI.

For example, a Client with Device UUID `"d1"` (UUID:`"0685B960-736F-46F7-BEC0-9E6CBD61ADC1"`) makes a RETRIEVE request on the `"/door"` Resource hosted on a Server with Device UUID `"d3"` where d3 has the ACL2s:

```
{
  "aclist2": [
    {
      "subject": {"uuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"},
      "resources": [{"href": "/door"}],
      "permission": 2, // RETRIEVE
      "aceid": 1
    },
    {
      "subject": {"authority": "owner", "role": "owner"},
      "resources": [{"href": "/door"}],
      "permission": 2, // RETRIEVE
      "aceid": 2
    }
  ]
}
```

```

1291     },
1292     {
1293         "subject": {"uuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"},
1294         "resources": [{"href": "/door/lock"}],
1295         "permission": 4, // UPDATE
1296         "aceid": 3
1297     }
1298 ],
1299 "rowneruuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"
1300 }

```

1301 The ACL indicates that Client "d1" has RETRIEVE permissions on the Resource. Hence when  
1302 device "d1" does a discovery on the "/door" Resource of the Server "d3", the response will include  
1303 all the URIs in the "/door" Resource. Client "d2" without a Role ID "owner" will get an error response  
1304 that includes no URI.

1305 Discovery results delivered to d1 regarding d3's "/door" Resource from the secure interface:

```

1306 [
1307     {
1308         "href": "/door",
1309         "rel": "self",
1310         "rt": ["oic.wk.col"],
1311         "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
1312         "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:5555"}]
1313     },
1314     {
1315         "href": "/door/lock",
1316         "rt": ["oic.r.lock.status"],
1317         "if": ["oic.if.a", "oic.if.baseline"],
1318         "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:5555"}]
1319     }
1320 ]

```

## 7 Security Provisioning

### 7.1 Device Identity

#### 7.1.1 General Device Identity

A Device shall be identified by a Device UUID value that is established as part of the device onboarding and contained in the "deviceuuid" Property of the "/oic/sec/doxm" Resource. Device UUIDs shall be unique within the scope of the corresponding OCF Security Domain, and are expected to be randomly generated and provisioned by the OBT. The DOTS is expected to verify that the chosen new Device UUID does not conflict with Device UUIDs previously introduced into the OCF Security Domain.

Devices maintain an association of their Device UUIDs and their own cryptographic credential(s) via "/oic/sec/cred" Resource. The identity is cryptographically bound in case of a certificate credential, or is bound via internal mappings in the "/oic/sec/cred" Resource otherwise. The "/oic/sec/cred" Resource maintains a list of a Device's own and other Device's credentials. Multiple credentials may be associated with the same Device UUID. A Device is expected to only present credentials associated with its own Device UUID for peer authentication purposes. Devices regard the "/oic/sec/cred" Resource as authoritative when verifying authentication credentials of a peer Device.

In case of an authenticated connection, the Device UUID is treated as a Client's identity for purposes of the Access Control check for the target Resource. The Device UUID of a Client is matched against the Subject UUIDs in the pre-provisioned entries of Server's "/oic/sec/acl2" Resource. The Server determines Client's Device UUID based on the credential used for the establishment of the session.

An OCF Platform, which may host multiple Devices, is identified by a Platform ID. The Platform ID is globally unique and inserted in the device in an integrity protected manner (e.g. inside secure storage or signed and verified).

An OCF Platform may have a secure execution environment, used to secure unique identifiers and secrets. If a Platform hosts multiple Devices, some mechanism is needed to provide each Device with the appropriate and separate security context.

#### 7.1.2 Device Identity for Devices with UAID [Deprecated]

This clause is intentionally left blank.

### 7.2 Device Ownership

This is an informative clause. Devices are logical entities that are security endpoints that have an identity that is authenticable using cryptographic credentials. A Device is Unowned when it is first initialized. Establishing device ownership is a process by which the device asserts its identity to the DOTS and the DOTS provisions an owner identity. This exchange results in the device changing its ownership state, thereby preventing a different DOTS from asserting administrative control over the device.

The ownership transfer process starts with the OBT discovering a new device that is in Unowned state through examination of the "Owned" Property of the "/oic/sec/doxm" Resource of the new device. At the end of ownership transfer, the following is accomplished:

- 1) The DOTS establishes a secure session with new device.
- 2) Optionally asserts any of the following:
  - a) Proximity (using PIN) of the OBT to the Platform.
  - b) Manufacturer's certificate asserting Platform vendor, model and other Platform specific attributes.

- 3) Determines the device identifier.
- 4) Determines the device owner.
- 5) Specifies the device owner (e.g. Device UUID of the OBT).
- 6) Provisions the device with owner's credentials.
- 7) Sets the "Owned" state of the new device to TRUE.

### 7.3 Device Ownership Transfer Methods

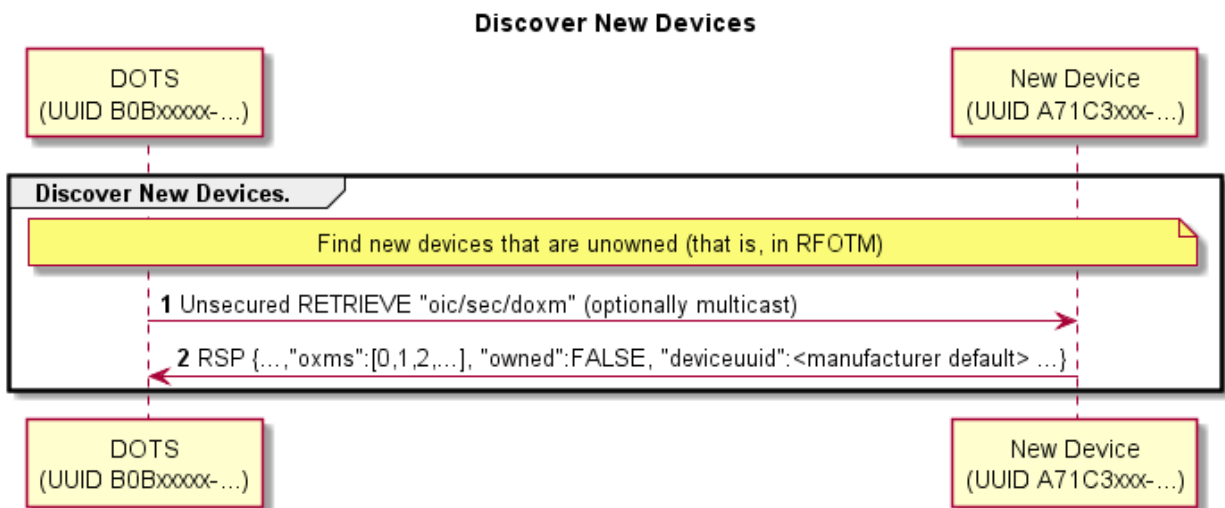
#### 7.3.1 OTM implementation requirements

This document provides specifications for several methods for ownership transfer. Implementation of each individual ownership transfer method is considered optional. However, each device shall implement at least one of the ownership transfer methods not including vendor specific methods.

All OTMs included in this document are considered optional. Each vendor is required to choose and implement at least one of the OTMs specified in this document. The OCF, does however, anticipate vendor-specific approaches will exist. Should the vendor wish to have interoperability between a vendor-specific OTM and OBTs from other vendors, the vendor must work directly with OBT vendors to ensure interoperability. Notwithstanding, standardization of OTMs is the preferred approach. In such cases, a set of guidelines is provided in 7.3.7 to help vendors in designing vendor-specific OTMs.

The "/oic/sec/doxm" Resource is extensible to accommodate vendor-defined owner transfer methods (OTM). The DOTS determines which OTM is most appropriate to onboard the new Device. All OTMs shall represent the onboarding capabilities of the Device using the "oxms" Property of the "/oic/sec/doxm" Resource. The DOTS determines the Device's supported credential types using the Supported Credential Types "sct" Property of the "/oic/sec/doxm" Resource. The DOTS and CMS provision credentials according to the credential types supported.

Figure 15 depicts new Device discovery sequence.



**Figure 15 – Discover New Device Sequence**



**Table 1 – Discover New Device Details**

| Step | Description  |
|------|--|
| 1    | The DOTS queries to see if the new device is not yet owned.  |
| 2    | The new device returns the "/oic/sec/doxm" Resource containing ownership status and supported OTMs. It also contains a temporal Device UUID that may change subsequent to successful owner transfer. The device should supply a temporal ID to facilitate discovery as a guest device.<br>Refer to OCF Onboarding Tool Specification for security considerations regarding selecting an OTM. |

A Device shall support selective use of unsecured multicast to receive RETRIEVE requests to the Device "/oic/sec/doxm" Resource, as shown in Figure 15. Clause 10.4 of the ISO/IEC 30118-1 provides the generic details for using CoAP multicast requests in OCF. Multicast retrieval of the "/oic/sec/doxm" Resource supports filtering using the "owned" query parameter. When a multicast RETRIEVE request omits the "owned" query parameter or includes the "owned" query parameter set to "false", then the Device shall respond only if the Device is in RFOTM and there is no open Device Onboarding Connection. Otherwise the request shall be ignored by the Device, regardless of ACE configuration.

Vendor-specific device OTMs shall adhere to the "/oic/sec/doxm" Resource Specification for OCS that results from vendor-specific device OTM. Vendor-specific OTM should include provisions for establishing trust in the new Device by the DOTS and optionally establishing trust in the OBT by the new Device.

The new device may have to perform some initialization steps at the beginning of an OTM. For example, if the Random PIN Based OTM is initiated, the new device may generate a random PIN value. The DOTS updates the oxmsel property of "/oic/sec/doxm" to the value corresponding to the OTM being used, before performing other OTM steps. This update notifies the new device that ownership transfer is starting.

The end state of a vendor-specific OTM shall allow the new Device to authenticate to the OBT and the OBT to authenticate to the new device.

Additional provisioning steps may be performed subsequent to owner transfer success leveraging the established OTM session.

### 7.3.2 SharedKey Credential Calculation

The SharedKey credential is derived using a PRF that accepts the key\_block value resulting from the DTLS handshake used for onboarding. The new Device shall use the following calculation to ensure interoperability across vendor products (the DOTS performs the same calculation):

SharedKey = PRF(Secret, Message);

Where:

- PRF shall use TLS 1.2 PRF defined by IETF RFC 5246 clause 5.
- Secret is the key\_block resulting from the DTLS handshake
  - See IETF RFC 5246 clause 6.3
  - The length of key\_block depends on cipher suite.
    - (e.g. 96 bytes for TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256  
40 bytes for TLS\_PSK\_WITH\_AES\_128\_CCM\_8)
- Message is a concatenation of the following:
  - DoxmType string for the current onboarding method (e.g. "oic.sec.doxm.jw")
    - See clause 13.2.2 for specific DoxmTypes

- 1431                   ▪   Owner ID is a UUID identifying the device owner identifier and the device that maintains SharedKey.
  - 1432                   •   Use raw bytes as specified in IETF RFC 4122 clause 4.1.2
- 1433                   ▪   Device UUID is new device's UUID
  - 1434                   •   Use raw bytes as specified in IETF RFC 4122 clause 4.1.2
- 1435       -   SharedKey Length will be 32 octets.
  - 1436                   ▪   If subsequent DTLS sessions use 128 bit encryption cipher suites the left most 16 octets will be used.
  - 1437                   ▪   DTLS sessions using 256-bit encryption cipher suites will use all 32 octets.

### 1438   **7.3.3   Certificate Credential Generation**

1439   The Certificate Credential will be used by Devices for secure bidirectional communication. The  
1440   certificates will be issued by a CMS or an external certificate authority (CA). This CA will be used  
1441   to mutually establish the authenticity of the Device.

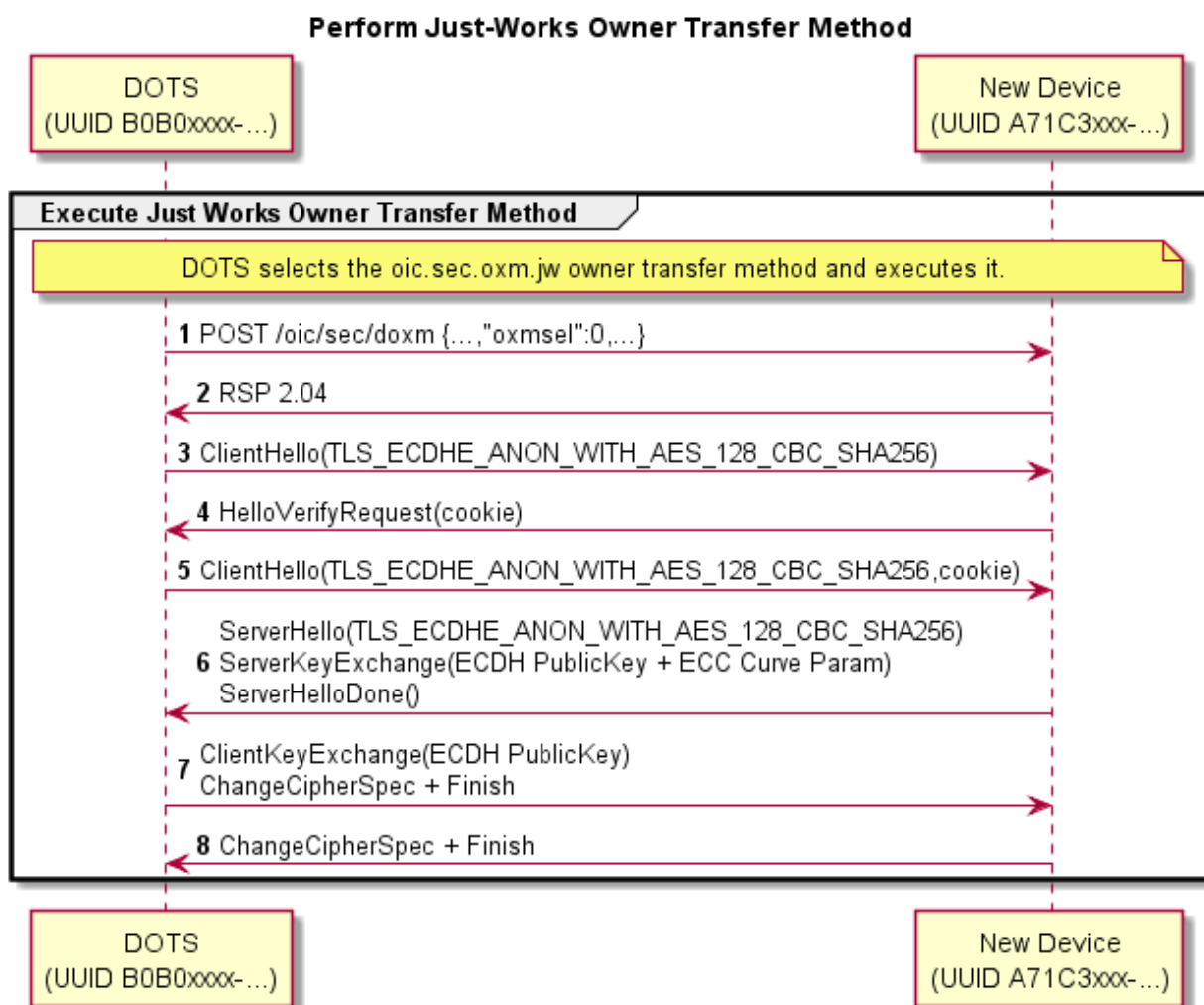
### 1442   **7.3.4   Just-Works OTM**

#### 1443   **7.3.4.1   Just-Works OTM General**

1444   Just-works OTM creates a symmetric key credential that is a pre-shared key used to establish a  
1445   secure connection through which a device should be provisioned for use within the owner's OCF  
1446   Security Domain. Provisioning additional credentials and Resources is a typical step following  
1447   ownership establishment. The pre-shared key is called SharedKey.

1448   The DOTS selects the Just-works OTM using the "oxmsel" Property of the "/oic/sec/doxm"  
1449   Resource and establishes a DTLS session using a cipher suite defined for the Just-works OTM.

1450   Just Works OTM sequence is shown in Figure 16 and steps described in Table 2.



**Figure 16 – A Just Works OTM**

**Table 2 – A Just Works OTM Details**

| Step  | Description  |
|---|--|
| 1, 2  | The DOTS notifies the Device that it selected the "Just Works" method.     |
| 3 - 8   | A DTLS session is established using anonymous Diffie-Hellman. <sup>a</sup> |
| <sup>a</sup> This method assumes the operator is aware of the potential for man-in-the-middle attack and has taken precautions to perform the method in a clean-room network. |  |

#### 7.3.4.2 Security Considerations

Anonymous Diffie-Hellman key agreement is subject to a man-in-the-middle attacker. Use of this method presumes that both the DOTS and the new device perform the "just-works" method assumes onboarding happens in a relatively safe environment absent of an attack device.

This method doesn't have a trustworthy way to prove the Device UUID asserted is reliably bound to the device.

1461 The new device should use a temporal Device UUID prior to transitioning to an owned device while  
1462 it is considered a guest device to prevent privacy sensitive tracking. The device asserts a non-  
1463 temporal Device UUID that could differ from the temporal value during the secure session in which  
1464 owner transfer exchange takes place. The DOTS verifies the asserted Device UUID does not  
1465 conflict with a Device UUID already in use. If it is already in use the existing credentials are used  
1466 to establish a secure session.

1467 An un-owned Device that also has established device credentials might be an indication of a  
1468 corrupted or compromised device.

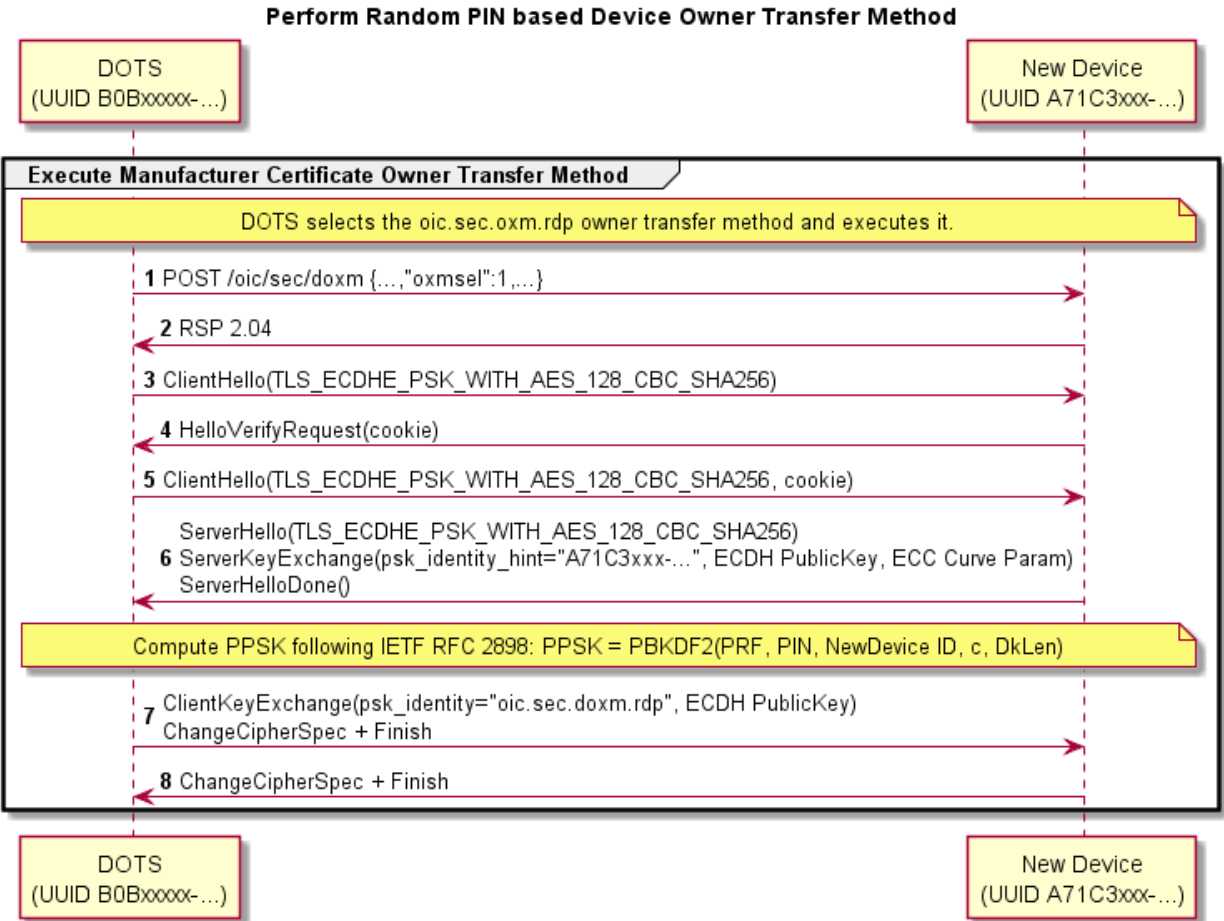
### 1469 **7.3.5 Random PIN based OTM**

#### 1470 **7.3.5.1 Random PIN based OTM General**

1471 The Random PIN method establishes physical proximity between the new device and the OBT can  
1472 prevent man-in-the-middle attacks. The Device generates a random number that is communicated  
1473 to the DOTS over an Out of Band Communication Channel. The definition of an Out of Band  
1474 Communication Channel is outside the scope of the definition of device OTMs. The DOTS and new  
1475 Device use the PIN in a key exchange as evidence that an End User authorized the transfer of  
1476 ownership by having physical access to the new Device via the Out-of-Band Communication  
1477 Channel.

#### 1478 **7.3.5.2 Random PIN based Owner Transfer Sequence**

1479 Random PIN-based OTM sequence is shown in Figure 17 and steps described in Table 3.



**Figure 17 – Random PIN-based OTM**

**Table 3 – Random PIN-based OTM Details**

| Step  | Description  |
|-------|--|
| 1, 2  | The DOTS notifies the Device that it selected the "Random PIN" method.   |
| 3 - 8 | A DTLS session is established using PSK-based Diffie-Hellman cipher suite. The PIN is supplied as the PSK parameter. The PIN is randomly generated by the new device then communicated via an Out of Band Communication Channel that establishes proximal context between the new device and the DOTS. The security principle is the attack device will be unable to intercept the PIN due to a lack of proximity. |

- The following requirements apply to the DTLS handshake messages for this OTM:
- At step 6:
    - The Server shall only use a DTLS ciphersuite supported by the Random PIN Based OTM (see clause 11.3.2.2),

- 1489       – The new Device shall set the "psk\_identity\_hint" field of the ServerKeyExchange message
- 1490       to the concatenation of
- 1491       – the string "oic.sec.doxm.rdp";
- 1492       – the colon character ':';
- 1493       – The "deviceuuid" Property of the "/oic/sec/doxm" Resource being sent in responses when
- 1494       the new Device is in RFOTM and when a Device Onboarding Connection is not currently
- 1495       established.

1496   – At step 7:

- 1497       – If the new Device determines that the "psk\_identity" field of the ClientKeyExchange
- 1498       message does not match the string "oic.sec.doxm.rdp", then the new Device shall reject
- 1499       the DTLS Handshake.

- 1500       – the new Device shall apply the key derivation below.

1501   NOTE The string "oic.sec.doxm.rdp" is the URN defined for the Random PIN-based OTM in Table 18 and is included to

1502   allow future OTMs to re-use the DTLS cipher suites without confusion about which OTM should be applied.

1503   This OTM uses a pseudo-random function (PBKDF2) defined by IETF RFC 2898 and a PIN

1504   exchanged via an Out of Band Communication Channel to generate a pre-shared key. The PIN-

1505   authenticated pre-shared key (PPSK) is supplied to TLS cipher suites that accept a PSK.

- 1506   – PPSK = PBKDF2(PRF, PIN, Device UUID, c, dkLen)

1507   The PBKDF2 function has the following parameters:

- 1508   – PRF – Uses the TLS 1.2 PRF defined by IETF RFC 5246.
- 1509   – PIN – obtained via Out of Band Communication Channel.
- 1510   – Device UUID – the "deviceuuid" Property of the "/oic/sec/doxm" Resource being sent in
- 1511   responses when the new Device is in RFOTM and when a Device Onboarding Connection is
- 1512   not currently established.

1513   Use raw bytes as specified in IETF RFC 4122 clause 4.1.2

- 1514   – c – Iteration count initialized to 1000
- 1515   – dkLen – Desired length of the derived PSK in octets.

### 1516   **7.3.5.3 Security Considerations**

1517   Security of the Random PIN mechanism depends on the entropy of the PIN. Using a PIN with

1518   insufficient entropy may allow a man-in-the-middle attack to recover any long-term credentials

1519   provisioned as a part of onboarding. In particular, learning the provisioned symmetric key

1520   credentials allows an attacker to masquerade as the onboarded device.

1521   It is recommended that the entropy of the PIN be enough to withstand an online brute-force attack,

1522   40 bits or more. For example, a 12-digit numeric PIN, or an 8-character alphanumeric (0-9a-z), or

1523   a 7-character case-sensitive alphanumeric PIN (0-9a-zA-Z). A man-in-the-middle attack is when

1524   the attacker is active on the network and can intercept and modify messages between the DOTS

1525   and device. In the man-in-the-middle attack, the attacker must recover the PIN from the key

1526   exchange messages in "real time", i.e., before the peer's time out and abort the connection attempt.

1527   Having recovered the PIN, he can complete the authentication step of key exchange. The guidance

1528   given here calls for a minimum of 40 bits of entropy, however, the assurance this provides depends

1529   on the resources available to the attacker. Given the parallelizable nature of a brute force guessing

1530   attack, the attack enjoys a linear speedup as more cores/threads are added. A more conservative

1531   amount of entropy would be 64 bits. Since the Random PIN OTM requires using a DTLS cipher

1532   suite that includes an ECDHE key exchange, the security of the Random PIN OTM is always at

1533   least equivalent to the security of the JustWorks OTM.

The Random PIN OTM also has an option to use PBKDF2 to derive key material from the PIN. The rationale is to increase the cost of a brute force attack, by increasing the cost of each guess in the attack by a tuneable amount (the number of PBKDF2 iterations). In theory, this is an effective way to reduce the entropy requirement of the PIN. Unfortunately, it is difficult to quantify the reduction, since an X-fold increase in time spent by the honest peers does not directly translate to an X-fold increase in time by the attacker. This asymmetry is because the attacker may use specialized implementations and hardware not available to honest peers. For this reason, when deciding how much entropy to use for a PIN, it is recommended that implementers assume PBKDF2 provides no security, and ensure the PIN has sufficient entropy.

The Random PIN device OTM security depends on an assumption that a secure Out of Band Communication Channel for communicating a randomly generated PIN from the new device to the OBT exists. If the Out of Band Communication Channel leaks some or the entire PIN to an attacker, this reduces the entropy of the PIN, and the attacks described above apply. The Out of Band Communication Channel should be chosen such that it requires proximity between the DOTS and the new device. The attacker is assumed to not have compromised the Out of Band Communication Channel. As an example Out of Band Communication Channel, the device may display a PIN to be entered into the OBT software. Another example is for the device to encode the PIN as a 2D barcode and display it for a camera on the DOTS device to capture and decode.

### **7.3.6 Manufacturer Certificate Based OTM**

#### **7.3.6.1 Manufacturer Certificate Based OTM General**

The manufacturer certificate-based OTM shall use a certificate embedded into the device by the manufacturer and may use a signed OBT, which determines the Trust Anchor between the device and the DOTS.

Manufacturer embedded certificates do not necessarily need to chain to an OCF Root CA trust anchor.

For some environments, policies or administrators, additional information about device characteristics may be sought. This list of additional attestations that OCF may or may not have tested (understanding that some attestations are incapable of testing or for which testing may be infeasible or economically unviable) can be found under the OCF Security Claims x509.v3 extension described in 9.4.2.2.6.

When utilizing certificate-based ownership transfer, devices shall utilize asymmetric keys with certificate data to authenticate their identities with the DOTS in the process of bringing a new device into operation on an OCF Security Domain. The onboarding process involves several discrete steps:

#### **1) Pre-on-board conditions**

- a) The credential element of the Device's credential Resource ("/oic/sec/cred") containing the manufacturer certificate shall be identified by the "credusage" Property containing the string "oic.sec.cred.mfgcert" to indicate that the credential contains a manufacturer certificate.
- b) The manufacturer certificate chain shall be contained in the identified credential element's "publicdata" Property.
- c) The device shall contain a unique and immutable ECC asymmetric key pair.
- d) If the device requires authentication of the DOTS as part of ownership transfer, it is presumed that the DOTS has been registered and has obtained a certificate for its unique and immutable ECC asymmetric key pair signed by the predetermined Trust Anchor.
- e) An End User has configured the DOTS app with network access info and account info (if any).

#### **2) The DOTS authenticates the Device using ECDSA to verify the signature. Additionally, the Device may authenticate the DOTS to verify the DOTS signature.**

1582 3) If authentication fails, the Device shall indicate the reason for failure and return to the RFOTM.  
1583 If authentication succeeds, the Device shall establish an encrypted link with the DOTS in  
1584 accordance with the negotiated cipher suite.

#### 1585 **7.3.6.2 Certificate Profiles**

1586 See 9.4.2 for details.

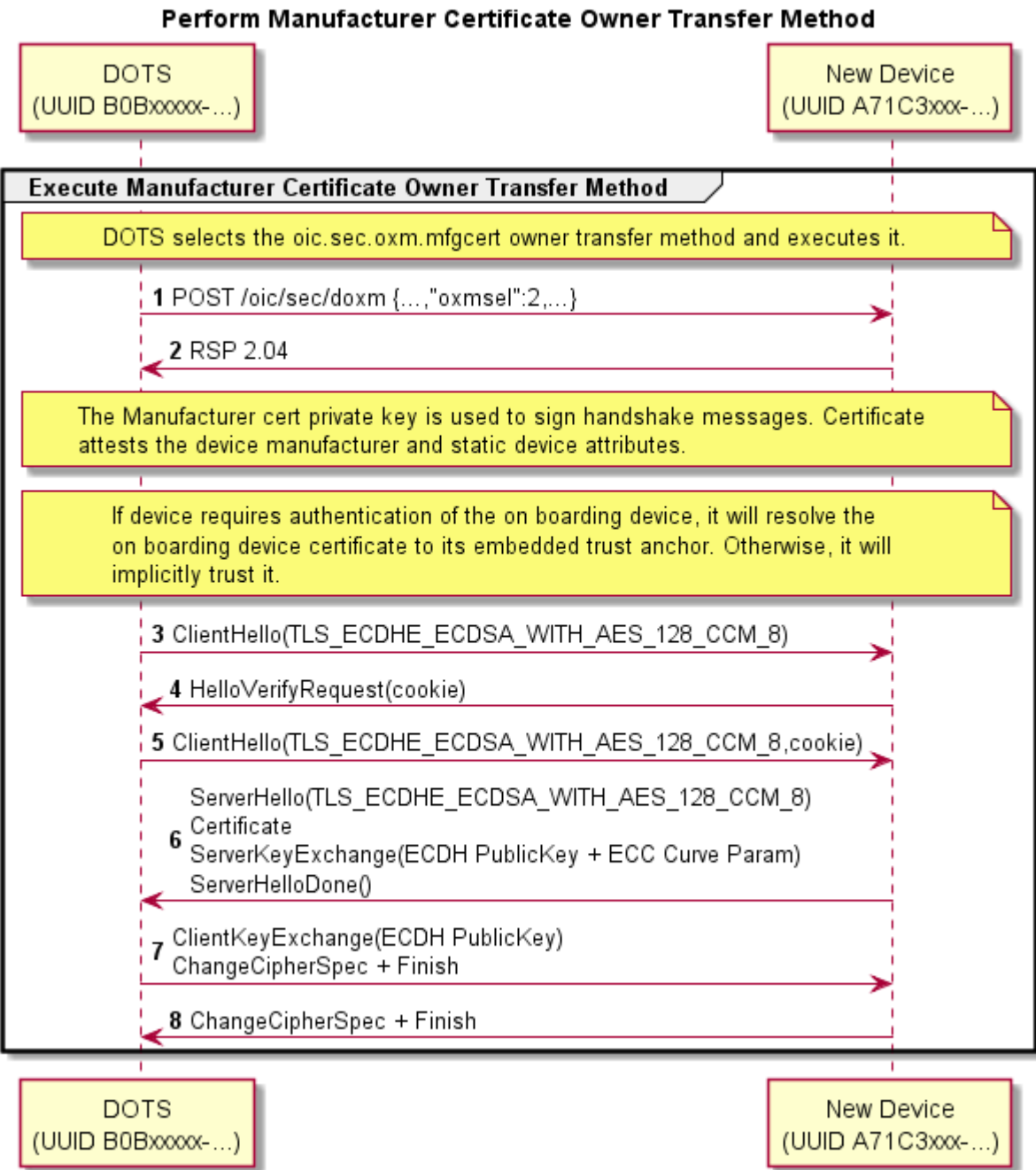
#### 1587 **7.3.6.3 Certificate Owner Transfer Sequence Security Considerations**

1588  
1589 The OBT shall authenticate the device during onboarding. The device will not authenticate the OBT.  
1590 During the DTLS handshake the server shall not send a Certificate Request.

#### 1591 **7.3.6.4 Manufacturer Certificate Based OTM Sequence**

1592 Manufacturer Certificate Based OTM sequence is shown in Figure 18 and steps described in  
1593 Table 4.





1595

1596

1597

1598

**Figure 18 – Manufacturer Certificate Based OTM Sequence**

**Table 4 – Manufacturer Certificate Based OTM Details**

| Step | Description  |
|------|--|
| 1, 2 | The DOTS notifies the Device that it selected the "Manufacturer Certificate" method. |

|       |  |
|-------|--|
| 3 - 8 | A DTLS session is established using the device's manufacturer certificate. The device's manufacturer certificate may contain data attesting to the Device hardening and security properties. |
|-------|--|

1599 If the Manufacturer Certificate Based OTM is selected at step 1, then the following requirements  
1600 apply:

1601 – At step 6:

1602 – The new Device shall use a DTLS ciphersuite supported for use with the Manufacturer  
1603 Certificate Based OTM (see clause 11.3.2.3),

1604 – The new Device shall not send a CertificateRequest message.

1605 NOTE: CertificateRequest message is sent when establishing the DTLS connection for Device authentication using  
1606 certificates (clause 10.4.1).

### 1607 **7.3.6.5 Security Considerations**

1608 The manufacturer certificate private key is embedded in the Platform with a sufficient degree of  
1609 assurance that the private key cannot be compromised.

1610 The Platform manufacturer issues the manufacturer certificate and attests the private key  
1611 protection mechanism.

### 1612 **7.3.7 Vendor Specific OTMs**

#### 1613 **7.3.7.1 Vendor Specific OTM General**

1614 The OCF anticipates situations where a vendor will need to implement an OTM that accommodates  
1615 manufacturing or Device constraints. The Device OTM Resource is extensible for this purpose.  
1616 Vendor-specific OTMs must adhere to a set of conventions that all OTMs follow.

1617 – The OBT must determine which credential types are supported by the Device. This is  
1618 accomplished by querying the Device's "/oic/sec/doxm" Resource to identify supported  
1619 credential types.

1620 – The OBT provisions the Device with OC(s).

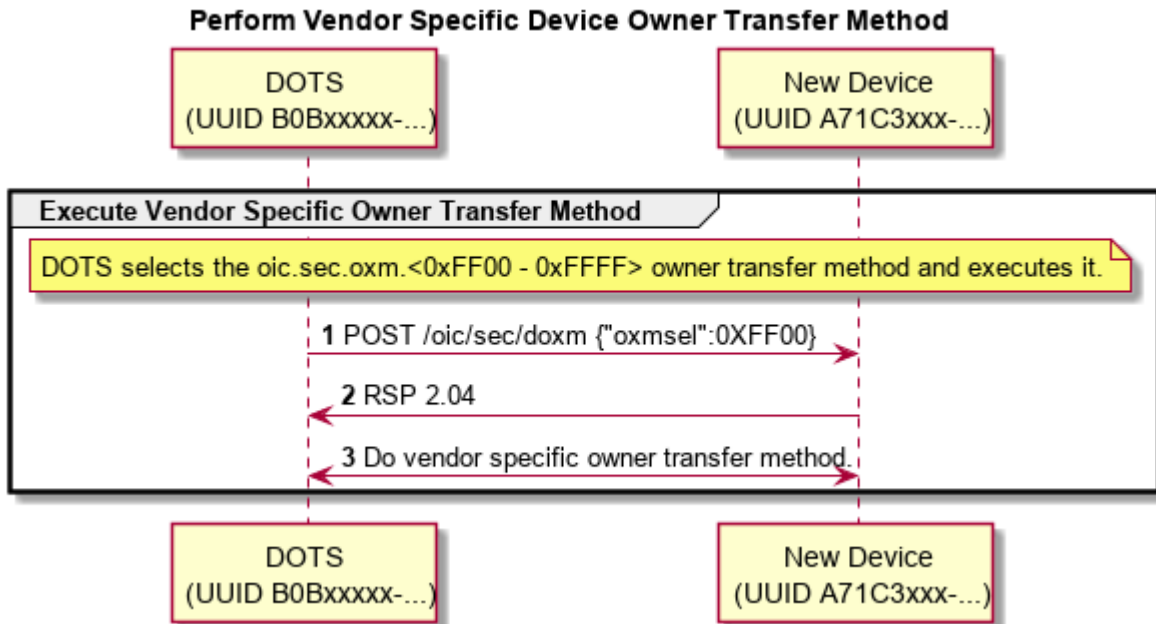
1621 – The OBT supplies the Device UUID and credentials for subsequent access to the OBT.

1622 – The OBT will supply second carrier settings sufficient for accessing the owner's OCF Security  
1623 Domain subsequent to ownership establishment.

1624 – The OBT may perform additional provisioning steps but must not invalidate provisioning tasks  
1625 to be performed by a security service.

#### 1626 **7.3.7.2 Vendor-specific Owner Transfer Sequence Example**

1627 Vendor-specific OTM sequence example is shown in Figure 19 and steps described in Table 5.



**Figure 19 – Vendor-specific Owner Transfer Sequence**

**Table 5 – Vendor-specific Owner Transfer Details**

| Step | Description                             |
|------|---|
| 1, 2 | The DOTS selects a vendor-specific OTM. |
| 3    | The vendor-specific OTM is applied      |

### 7.3.7.3 Security Considerations

The vendor is responsible for considering security threats and mitigation strategies.

### 7.3.8 Establishing Owner Credentials

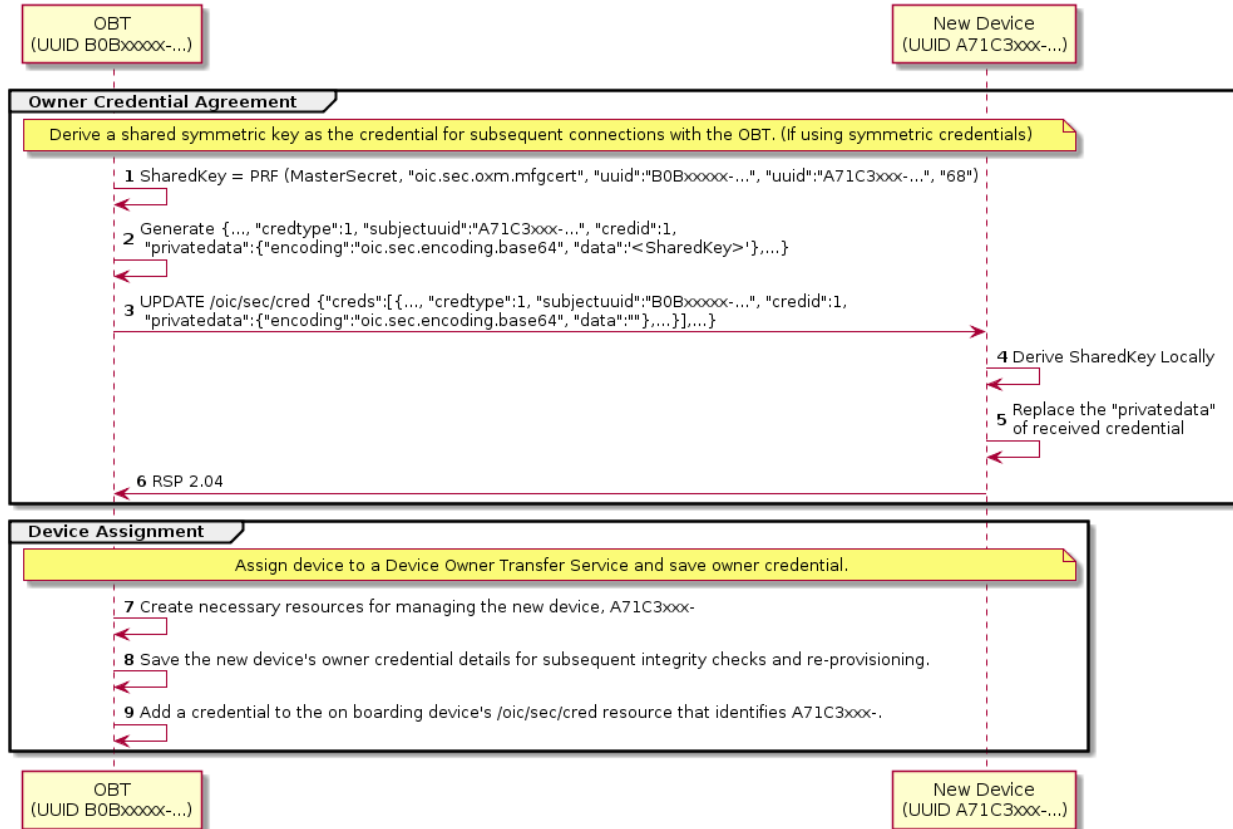
Once the OBT and the new Device have authenticated and established an encrypted connection using one of the defined OTM methods, the Owner Credential(s) can be provisioned.

The Owner Credential is provisioned as part of Ownership Transfer Method, and may be provisioned directly by CMS.

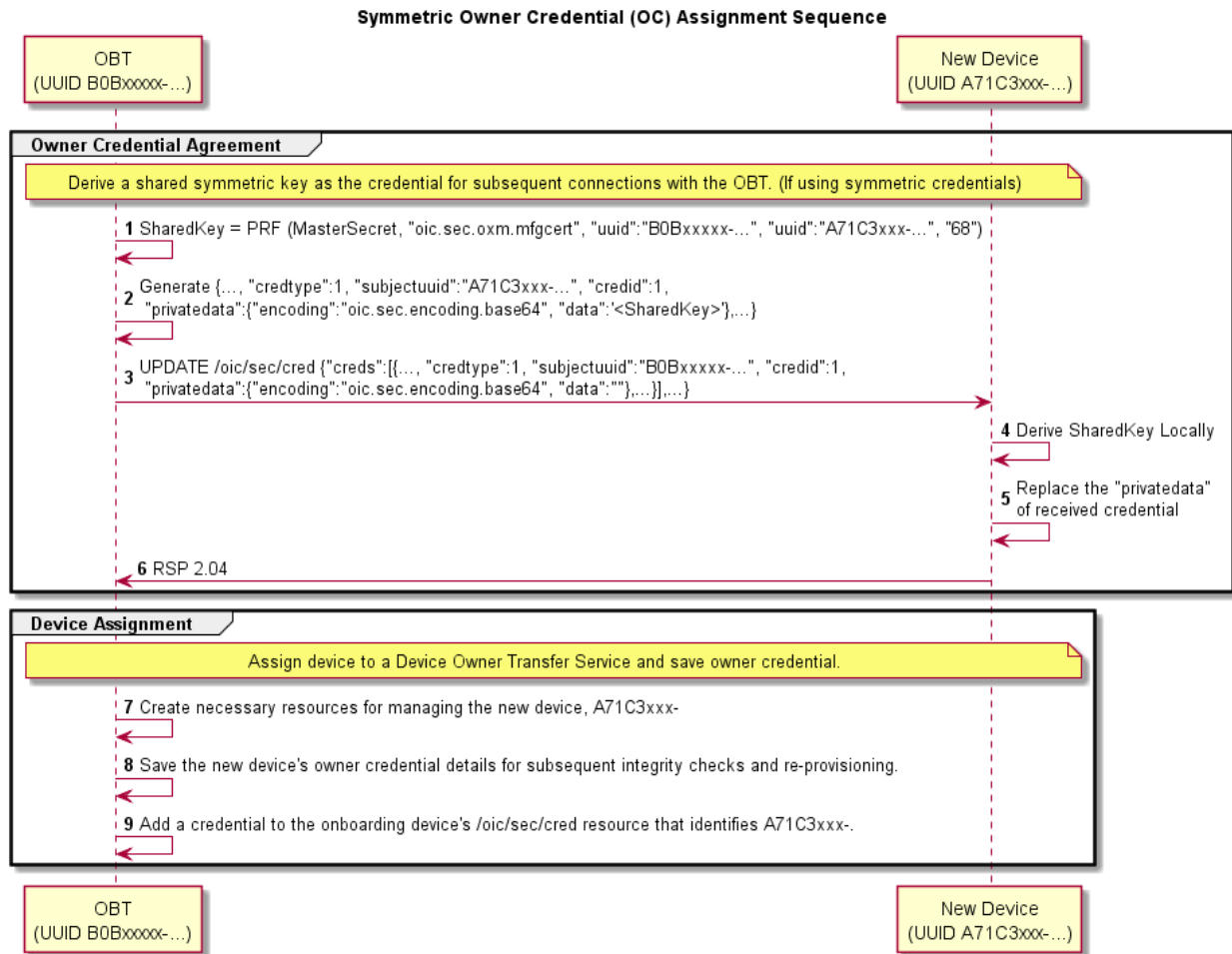
The steps for establishing Device's owner credentials (OC) as part of OTM are:

- 1) The OBT establishes the Device UUID and Device Owner Id.
- 2) The OBT then establishes Device's symmetric OC - See Figure 20 and Table 6.
- 3) Configure Device services.
- 4) Configure Device for peer to peer interaction.

## Symmetric Owner Credential (OC) Assignment Sequence



1645



**Figure 20 – Symmetric Owner Credential Provisioning Sequence**

**Table 6 – Symmetric Owner Credential Assignment Details**

| Step | Description   |
|------|---|
| 1, 2 | The OBT uses a pseudo-random-function (PRF), the master secret resulting from the DTLS handshake, and other information to generate a symmetric key credential Resource Property - SharedKey. |
| 3    | The OBT creates a credential Resource Property set based on SharedKey and then sends the Resource Property set to the new Device with empty "privatedata" Property value.                     |
| 4, 5 | The new Device locally generates the SharedKey and updates it to the "privatedata" Property of the credential Resource Property set.  |
| 6    | The new Device sends a success message.   |
| 7    | The onboarding service creates a subjects Resource for the new device (e.g./A71C3xxx-...)   |
| 8    | The onboarding service provisions its "/oic/svc/dots/subjects/A71C3xxx-/cred" Resource with   |

|   |   |
|---|---|
|   | the owner credential. Credential type is SYMMETRIC KEY.   |
| 9 | (optional) The onboarding service provisions its own "/oic/sec/cred" Resource with the owner credential for new device. Credential type is SYMMETRIC KEY. |

1650 In particular when OBT establishes symmetric owner credentials as part of OTM sequence:

- 1651 – The OBT generates a Shared Key using the SharedKey Credential Calculation method  
1652 described in 7.3.2.
- 1653 – The OBT sends an empty key to the new Device's "/oic/sec/cred" Resource, identified as a  
1654 symmetric pair-wise key. The Subject UUID of the "/oic/sec/cred" entry shall match the Device  
1655 UUID of the OBT.
- 1656 – Upon receipt of the OBT's symmetric owner credential, the new Device shall independently  
1657 generate the Shared Key using the SharedKey Credential Calculation method described in 7.3.2  
1658 and store it with the owner credential.
- 1659 – The new Device shall use the Shared Key owner credential(s) stored via the "/oic/sec/cred"  
1660 Resource to authenticate the owner during subsequent connections.

### 1661 7.3.9 Security Profile Assignment

1662 OCF Devices may have been evaluated according to an OCF Security Profile. Evaluation results  
1663 could be accessed from a manufacturer's certificate, OCF web server or other public repository.  
1664 The DOTS reviews evaluation results to determine which OCF Security Profiles the OCF Device is  
1665 authorized to possess and configures the Device with the subset of evaluated security profiles best  
1666 suited for the OCF Security Domain owner's intended segmentation strategy.

1667 The OCF Device vendor shall set a manufacturer default value for the "supportedprofiles" Property  
1668 of the "/oic/sec/sp" Resource to match those approved by OCF's testing and certification process.  
1669 The "currentprofile" Property of the "/oic/sec/sp" Resource shall be set to one of the values  
1670 contained in the "supportedprofiles". The manufacturer default value shall be re-asserted when the  
1671 Device transitions to RESET.

1672 The OCF Device shall only allow the "/oic/sec/sp" Resource to be updated when the Device is in  
1673 one of the following Device States: RFOTM, RFPRO, SRESET and may not allow any update as  
1674 directed by a Security Profile.

1675 The DOTS may update the "supportedprofiles" Property of the "/oic/sec/sp" Resource with a subset  
1676 of the OCF Security Profiles values the Device achieved as part of OCF Conformance testing. The  
1677 DOTS may locate conformance results by inspecting manufacturer certificates supplied with the  
1678 OCF Device by selecting the "credusage" Property of the "/oic/sec/cred" Resource having the value  
1679 of "oic.sec.cred.mfgcert". The DOTS may further locate conformance results by visiting a well-  
1680 known OCF web site URI corresponding to the ocfCPLAttributes extension fields (clause 9.4.2.2.7).  
1681 The DOTS may select a subset of Security Profiles (from those evaluated by OCF conformance  
1682 testing) based on a local policy.

1683 As part of onboarding (while the OTM session is active) the DOTS should configure ACE entries to  
1684 allow DOTS access subsequent to onboarding.

1685 The DOTS should update the "currentprofile" Property of the "/oic/sec/sp" Resource with the value  
1686 that most correctly depicts the OCF Security Domain owner's intended Device deployment strategy.

1687 The CMS may issue role credentials using the Security Profile value (e.g. the "sp-blue-v0 OID") to  
1688 indicate the OCF Security Domain owner's intention to segment the OCF Security Domain  
1689 according to a Security Profile. The CMS retrieves the supportedprofiles Property of the  
1690 "/oic/sec/sp" Resource to select role names corroborated with the Device's supported Security  
1691 Profiles when issuing role credentials.

If the CMS issues role credentials based on a Security Profile, the AMS supplies access control entries that include the role designation(s).

## 7.4 Provisioning

### 7.4.1 Provisioning Flows

#### 7.4.1.1 Provisioning Flows General

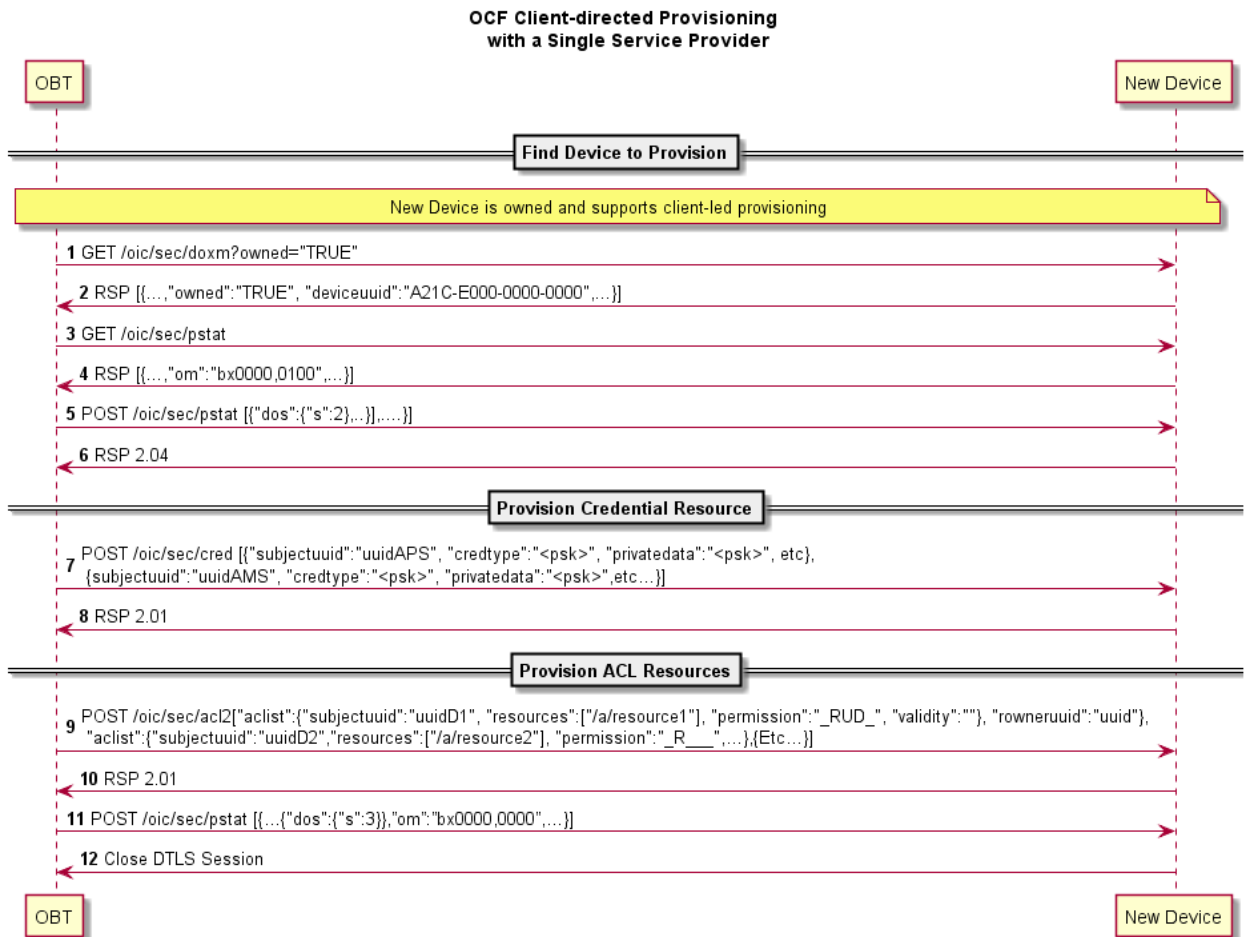
As part of onboarding a new Device a secure channel is formed between the new Device and the OBT. Subsequent to the Device ownership status being changed to "owned", there is an opportunity to begin provisioning. The OBT provisions the support services that should be subsequently used to complete Device provisioning and on-going Device management.

The Device employs a Client-directed provisioning strategy. The "/oic/sec/pstat" Resource identifies the provisioning strategy and current provisioning status. The provisioning service should determine which provisioning strategy is most appropriate for the OCF Security Domain. See 13.8 for additional detail.

#### 7.4.1.2 Client-directed Provisioning

Client-directed provisioning relies on a provisioning service that identifies Servers in need of provisioning then performs all necessary provisioning duties.

An example of Client-directed provisioning is shown in Figure 21 and steps described in Table 7.



**Figure 21 – Example of Client-directed provisioning**

1711

1712

**Table 7 – Steps describing Client -directed provisioning**

| Step   | Description   |
|--------|---|
| 1      | Discover Devices that are owned and support Client-directed provisioning.   |
| 2      | The "/oic/sec/doxm" Resource identifies the Device and it's owned status.   |
| 3      | DOTS (on OBT) obtains the new Device's provisioning status found in "/oic/sec/pstat" Resource   |
| 4      | The "pstat" Resource describes the types of provisioning modes supported and which is currently configured. A Device manufacturer should set a default current operational mode ("om"). If the "om" isn't configured for Client-directed provisioning, its "om" value can be changed. |
| 5 - 6  | Change Device state to RFPRO.   |
| 7 - 8  | CMS (on OBT) instantiates the "/oic/sec/cred" Resource. It contains credentials for the provisioned services and other Devices  |
| 9 - 10 | AMS (on OBT) instantiates "/oic/sec/acl2" Resource.   |
| 11     | The new Device provisioning status mode is updated to reflect that ACLs have been configured. (RFNOP).  |
| 12     | The secure session is closed.   |

### 1713 7.4.1.3 Server-directed Provisioning [DEPRECATED]

1714 This clause is intentionally left blank.

### 1715 7.4.1.4 Server-directed Provisioning Involving Multiple Support Services 1716 [DEPRECATED]

1717 This clause is intentionally left blank.

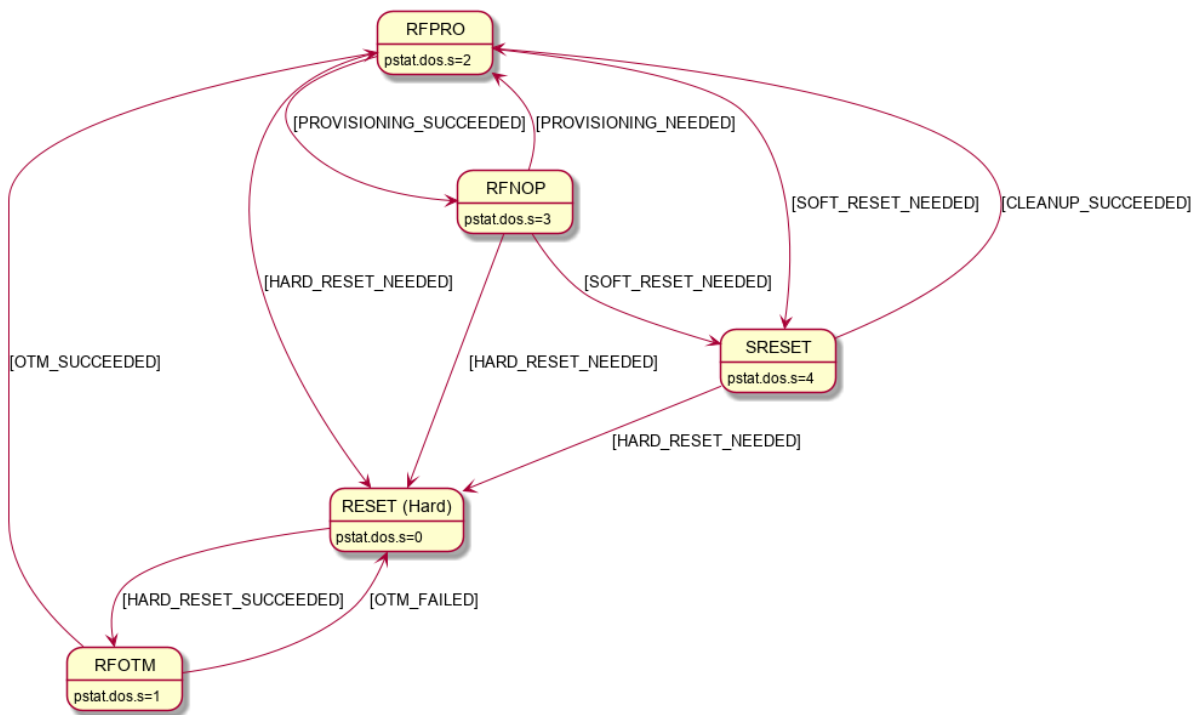
## 1718 8 Device Onboarding State Definitions

### 1719 8.1 Device Onboarding General

1720 As explained in 5.3, the process of onboarding completes after the ownership of the Device has  
1721 been transferred and the Device has been provisioned with relevant configuration/services as  
1722 explained in 5.4. The Figure 22 shows the various states a Device can be in during the Device  
1723 lifecycle. Device shall reject any requests to perform a state transition not shown on Figure 22.

1724 The "/pstat.dos.s" Property is RW by the "/oic/sec/pstat" Resource owner (e.g. "doxs" service) so  
1725 that the Resource owner can remotely update the Device state. When the Device is in RFNOP or  
1726 RFPRO, ACLs can be used to allow remote control of Device state by other Devices. When the  
1727 Device state is SRESET the Device OC may be the only indication of authorization to access the  
1728 Device. The Device owner may perform low-level consistency checks and re-provisioning to get  
1729 the Device suitable for a transition to RFPRO.





**Figure 22 – Device state model**

As shown in the diagram, at the conclusion of the provisioning step, the Device comes in RFNOP where it has all it needs in order to start interoperating with other Devices. Clause 8.5 specifies the minimum mandatory configuration that a Device shall hold in order to be considered as RFNOP.

In the event of power loss or Device failure, the Device should remain in the same state that it was in prior to the power loss / failure

If a Device or Resource owner OBSERVES "/pstat.dos.s", then transitions to SRESET will give early warning notification of Devices that may require SVR consistency checking.

In order for onboarding to function, the Device shall have the following Resources installed:

- 1) "/oic/sec/doxm" Resource
- 2) "/oic/sec/pstat" Resource
- 3) "/oic/sec/cred" Resource

The values contained in these Resources are specified in the state definitions in 8.2, 8.3, 8.4, 8.5 and 8.6. Access policy for these and other SVRs are also described.

## 8.2 Device Reset State Definition

The /pstat.dos.s = RESET is defined as a "hard" reset to manufacturer defaults. Hard reset also defines a state where the Device asset is ready to be transferred to another party.

The Platform manufacturer should provide a physical mechanism (e.g. button) that forces Platform reset. All Devices hosted on the same Platform transition their Device states to RESET when the Platform reset is asserted.

The following Resources and their specific properties shall have the value as specified:

- The "owned" Property of the "/oic/sec/doxm" Resource shall transition to FALSE.
- The "devowneruuid" Property of the "/oic/sec/doxm" Resource shall be nil UUID.

- 1754 – The "deviceuuid" Property of the "/oic/sec/doxm" Resource shall be set to the manufacturer  
1755 default value.
- 1756 – The "sct" Property of the "/oic/sec/doxm" Resource shall be reset to the manufacturer's default  
1757 value.
- 1758 – The "oxmsel" Property of the "/oic/sec/doxm" Resource shall be reset to the manufacturer's  
1759 default value.
- 1760 – The "isop" Property of the "/oic/sec/pstat" Resource shall be FALSE.
- 1761 – The "dos" Property of the "/oic/sec/pstat" Resource shall be updated: dos.s shall equal "RESET".
- 1762 – The "om" (operational modes) Property of the "/oic/sec/pstat" Resource shall be set to the  
1763 manufacturer default value.
- 1764 – The "sm" (supported operational modes) Property of the "/oic/sec/pstat" Resource shall be set  
1765 to the manufacturer default value.
- 1766 – The "creds" Property of the "/oic/sec/cred" Resource shall be set to the manufacturer default  
1767 value.
- 1768 – The "aclist2" Property of the "/oic/sec/acl2" Resource shall be set to the manufacturer default  
1769 value.
- 1770 – The "owneruuid" Property of "/oic/sec/pstat", "/oic/sec/doxm", "/oic/sec/acl2", and  
1771 "/oic/sec/cred" Resources shall be nil UUID.
- 1772 – The "usedspace" Property of the "/oic/sec/ael" Resource shall be set to 0.
- 1773 – The "categoryfilter" Property of the "/oic/sec/ael" Resource shall be set to the manufacturer's  
1774 default value.
- 1775 – The "priorityfilter" Property of the "/oic/sec/ael" Resource shall be set to the manufacturer's  
1776 default value.
- 1777 – The "events" Property of the "/oic/sec/ael" Resource shall be set to an empty array.
- 1778 – The "supportedprofiles" Property of the "/oic/sec/sp" Resource shall be set to the manufacturer  
1779 default value.
- 1780 – The "currentprofile" Property of the "/oic/sec/sp" Resource shall be set to the manufacturer  
1781 default value.
- 1782 – If "/oic/sec/sdi" Resource is exposed by a Device:
- 1783     – The "uuid" Property of the Resource shall be set to nil UUID
- 1784     – The "name" Property of the Resource shall be set to the empty string
- 1785     – The "priv" Property of the Resource shall be set to FALSE
- 1786 – The Device shall not accept DTLS connection attempts nor TLS connection attempts nor any  
1787 other requests, including discovery requests.
- 1788 – Any existing DTLS or TLS Connections shall be closed.

### 1789 **8.3 Device Ready For Owner Transfer Mechanism State Definition**

1790 The following Resources and their specific properties shall have the value as specified when the  
1791 Device enters ready for ownership transfer:

- 1792 – The "owned" Property of the "/oic/sec/doxm" Resource shall be FALSE and will transition to  
1793 TRUE.
- 1794 – The "devowneruuid" Property of the "/oic/sec/doxm" Resource shall be nil UUID.
- 1795 – The "deviceuuid" Property of the "/oic/sec/doxm" Resource shall be set to the manufacturer  
1796 default value.
- 1797 – The "isop" Property of the "/oic/sec/pstat" Resource shall be FALSE.

- 1798 – The "dos" of the "/oic/sec/pstat" Resource shall be updated: "dos.s" shall equal "RFOTM".
- 1799 – The "/oic/sec/cred" Resource shall contain credential(s) if required by the selected OTM
- 1800 – If there is no open Device Onboarding Connection, then
  - 1801 – Anonymous Retrieve and Updates requests (those arriving over unauthenticated channel
  - 1802 such as CoAP) for the "/oic/sec/doxm" Resource shall be granted.
  - 1803 – If an anonymous request to Update the "/oic/sec/doxm" Resource attempts to update
  - 1804 "oxmsel" to a value that is not indicated as supported by the Device in "oxms", then the
  - 1805 Device shall reject the request with an appropriate error message (e.g. bad request).
  - 1806 – All Retrieve requests to the "/oic/sec/pstat" Resource shall be granted.
  - 1807 – All other requests, with the exception of Retrieve requests to the Discovery Resources
  - 1808 ("/oic/res", "/oic/d" and "/oic/p"), shall be rejected with an appropriate error message (e.g.
  - 1809 forbidden).
  - 1810 – Prior to a successful anonymous Update of "oxmsel" in "/oic/sec/doxm", all attempts to
  - 1811 establish new DTLS connections shall be rejected.
  - 1812 – After a successful anonymous Update of "oxmsel" in "/oic/sec/doxm",
  - 1813 – The Device shall allow establishing a Device Onboarding Connection (DOC) matching the
  - 1814 "oxmsel" Property of the "/oic/sec/doxm" Resource (as specified in clause 7.3) , and shall
  - 1815 reject attempts to establish other DTLS connections.
- 1816 – If there is an open DOC, then
  - 1817 – All requests received over the DOC which target DCRs shall be granted, regardless of the
  - 1818 configuration of the ACEs in the "/oic/sec/acl2" Resource.
  - 1819 – All unicast requests which are not received over the open Device DOC shall be rejected
  - 1820 with an appropriate error message (e.g. forbidden), regardless of the configuration of the
  - 1821 ACEs in the "/oic/sec/acl2" Resource.
  - 1822 – All attempts to establish new DTLS connections shall be rejected.
- 1823 – If the DOC is closed in RFOTM, then the Device shall transition to RESET.

#### 1824 **8.4 Device Ready For Provisioning State Definition**

1825 The following Resources and their specific properties shall have the value as specified when the  
 1826 Device enters ready for provisioning:

- 1827 – The "owned" Property of the "/oic/sec/doxm" Resource shall be TRUE.
- 1828 – The "devowneruuid" Property of the "/oic/sec/doxm" Resource shall not be nil UUID.
- 1829 – The "deviceuuid" Property of the "/oic/sec/doxm" Resource shall not be nil UUID and shall be
- 1830 set to the value that was determined during RFOTM processing.
- 1831 – The "oxmsel" Property of the "/oic/sec/doxm" Resource shall have the value of the actual OTM
- 1832 used during ownership transfer.
- 1833 – The "isop" Property of the "/oic/sec/pstat" Resource shall be FALSE.
- 1834 – The "dos" of the "/oic/sec/pstat" Resource shall be updated: "dos.s" shall equal "RFPRO".
- 1835 – The "rowneruuid" Property of every installed Resource shall be set to a valid Resource owner
- 1836 (i.e. an entity that is authorized to instantiate or update the given Resource). Failure to set a
- 1837 "rowneruuid" may result in an orphan Resource.
- 1838 – The "/oic/sec/cred" Resource shall contain credentials for each entity referenced by
- 1839 "rowneruuid" and "devowneruuid" Properties.
- 1840 – All requests to the "/oic/sec/roles" Resource received over a mutually-authenticated connection
- 1841 established using an identity certificate shall be granted, regardless of the configuration of the
- 1842 ACEs in the "/oic/sec/acl2" Resource, subject to the conditions in clause 10.4.2.

- 1843 – If there is an open DOC, then all requests received over the DOC which target a DCR shall be  
1844 granted, regardless of the configuration of the ACEs in the "/oic/sec/acl2" Resource.
- 1845 – The Device shall allow establishing DTLS connections authenticated with locally issued  
1846 credentials (clauses 10.2 and 10.4) and shall reject attempts to establish other DTLS  
1847 connections.

## 1848 **8.5 Device Ready For Normal Operation State Definition**

1849 The following Resources and their specific properties shall have the value as specified when the  
1850 Device enters ready for normal operation:

- 1851 – The "owned" Property of the "/oic/sec/doxm" Resource shall be TRUE.
- 1852 – The "devowneruuid" Property of the "/oic/sec/doxm" Resource shall not be nil UUID.
- 1853 – The "deviceuuid" Property of the "/oic/sec/doxm" Resource shall not be nil UUID and shall be  
1854 set to the ID that was configured during OTM. Also the value of the "di" Property in "/oic/d" shall  
1855 be the same as the deviceuuid.
- 1856 – The "oxmsel" Property of the "/oic/sec/doxm" Resource shall have the value of the actual OTM  
1857 used during ownership transfer.
- 1858 – The "isop" Property of the "/oic/sec/pstat" Resource shall be set to TRUE by the Server once  
1859 transition to RFNOP is otherwise complete.
- 1860 – The "dos" of the "/oic/sec/pstat" Resource shall be updated: "dos.s" shall equal "RFNOP".
- 1861 – The "rowneruuid" Property of every installed Resource shall be set to a valid Resource owner  
1862 (i.e. an entity that is authorized to instantiate or update the given Resource). Failure to set a  
1863 "rowneruuid" results in an orphan Resource.
- 1864 – The "/oic/sec/cred" Resource shall contain credentials for each service referenced by  
1865 "rowneruuid" and "devowneruuid" Properties.
- 1866 – All requests to the "/oic/sec/roles" Resource received over a mutually-authenticated connection  
1867 established using an identity certificate shall be granted, regardless of the configuration of the  
1868 ACEs in the "/oic/sec/acl2" Resource, subject to the conditions in clause 10.4.2.
- 1869 – If there is an open DOC, then requests received over the DOC shall have access decisions  
1870 determined as follows:
  - 1871 – A request which targets a DCR shall be granted, regardless of the configuration of the ACEs  
1872 in the "/oic/sec/acl2" Resource.
  - 1873 – A request which targets an NCR shall be granted by matching an ACE as per normal request  
1874 authorization, with "subject" matching the "anon-clear" connection type.
- 1875 – The Device shall allow establishing DTLS connections authenticated with locally issued  
1876 credentials and shall reject attempts to establish other DTLS connections.

## 1877 **8.6 Device Soft Reset State Definition**

1878 The soft reset state is defined (e.g. "/pstat.dos.s" = SRESET) where entrance into this state means  
1879 the Device is not operational but remains owned by the current owner. The Device may exit  
1880 SRESET by authenticating to a DOTS (e.g. "rt" = "oic.r.doxx") using the OC provided during original  
1881 onboarding (but should not require use of an OTM /doxm.oxms).

1882 If the DOTS credential cannot be found or is determined to be corrupted, the Device state  
1883 transitions to RESET. The Device should remain in SRESET if the DOTS credential fails to validate  
1884 the DOTS. This mitigates denial-of-service attacks that may be attempted by non-DOTS Devices.

1885 When in SRESET, the following Resources and their specific Properties shall have the values as  
1886 specified.

- 1887 – The "owned" Property of the "/oic/sec/doxm" Resource shall be TRUE.
- 1888 – The "devowneruuid" Property of the "/oic/sec/doxm" Resource shall remain non-null.

- 1889 – The "deviceuuid" Property of the "/oic/sec/doxm" Resource shall remain non-null.
- 1890 – The "sct" Property of the "/oic/sec/doxm" Resource shall retain its value.
- 1891 – The "oxmsel" Property of the "/oic/sec/doxm" Resource shall retain its value.
- 1892 – The "isop" Property of the "/oic/sec/pstat" Resource shall be FALSE.
- 1893 – The "/oic/sec/pstat.dos.s" Property shall be SRESET.
- 1894 – The "om" (operational modes) Property of the "/oic/sec/pstat" Resource shall be "client-directed
- 1895 mode".
- 1896 – The "sm" (supported operational modes) Property of "/oic/sec/pstat" Resource may be updated
- 1897 by the Device owner (aka DOTS).
- 1898 – The "rowneruuid" Property of "/oic/sec/pstat", "/oic/sec/doxm", "/oic/sec/acl2", and
- 1899 "/oic/sec/cred" Resources may be reset by the Device owner (aka DOTS) and re-provisioned.
- 1900 – All requests to the "/oic/sec/roles" Resource received over a mutually-authenticated connection
- 1901 established using an identity certificate shall be granted, regardless of the configuration of the
- 1902 ACEs in the "/oic/sec/acl2" Resource, subject to the conditions in clause 10.4.2.
- 1903 – If there is an open DOC, then all requests received over the DOC which target a DCR shall be
- 1904 granted, regardless of the configuration of the ACEs in the "/oic/sec/acl2" Resource.
- 1905 – The Device shall allow establishing DTLS connections authenticated with locally issued
- 1906 credentials and shall reject attempts to establish other DTLS connections.
- 1907

## 1908 **9 Security Credential Management**

### 1909 **9.1 Preamble**

1910 This clause provides an overview of the credential types in OCF, along with details of credential  
1911 use, provisioning and ongoing management.

### 1912 **9.2 Credential Lifecycle**

#### 1913 **9.2.1 Credential Lifecycle General**

1914 OCF credential lifecycle has the following phases: (1) creation, (2) deletion, (3) refresh and (4)  
1915 revocation.

#### 1916 **9.2.2 Creation**

1917 The CMS can provision credentials to the credential Resource onto the Device. The Device shall  
1918 verify the CMS is authorized by matching the rowneruuid Property of the "/oic/sec/cred" Resource  
1919 to the DeviceID of the credential the CMS used to establish the secure connection.

1920 Credential Resources created using a CMS may involve specialized credential issuance protocols  
1921 and messages. These may involve the use of public key infrastructure (PKI) such as a certificate  
1922 authority (CA), symmetric key management such as a key distribution centre (KDC) or as part of a  
1923 provisioning action by a DOTS, CMS or AMS.

#### 1924 **9.2.3 Deletion**

1925 The CMS can delete credentials from the credential Resource. The Device (e.g. the Device where  
1926 the credential Resource is hosted) should delete credential Resources that have expired.

1927 An expired credential Resource may be deleted to manage memory and storage space.

1928 Deletion in OCF key management is equivalent to credential suspension.

#### 1929 **9.2.4 Refresh**

1930 Credential refresh may be performed before it expires. The CMS performs credential refresh.

1931 The "/oic/sec/cred" Resource supports expiry using the Period Property. Credential refresh may be  
1932 applied when a credential is about to expire or is about to exceed a maximum threshold for bytes  
1933 encrypted.

1934 A credential refresh method specifies the options available when performing key refresh. The  
1935 Period Property informs when the credential should expire. The Device may proactively obtain a  
1936 new credential using a credential refresh method using current unexpired credentials to refresh the  
1937 existing credential. If the Device does not have an internal time source, the current time should be  
1938 obtained from a CMS at regular intervals.

1939 If the onboarding established credentials are allowed to expire the DOTS shall re-onboard the  
1940 Device to re-apply device owner transfer steps.

1941 All Devices shall support at least one credential refresh method.

#### 1942 **9.2.5 Revocation**

1943 Credentials issued by a CMS may be equipped with revocation capabilities. In situations where the  
1944 revocation method involves provisioning of a revocation object that identifies a credential that is to  
1945 be revoked prior to its normal expiration period, a credential Resource is created containing the  
1946 revocation information that supersedes the originally issued credential. The revocation object  
1947 expiration should match that of the revoked credential so that the revocation object is cleaned up  
1948 upon expiry.

1949 It is conceptually reasonable to consider revocation applying to a credential or to a Device. Device  
1950 revocation asserts all credentials associated with the revoked Device should be considered for  
1951 revocation. Device revocation is necessary when a Device is lost, stolen or compromised. Deletion  
1952 of credentials on a revoked Device might not be possible or reliable.

## 1953 **9.3 Credential Types**

### 1954 **9.3.1 Preamble**

1955 The "/oic/sec/cred" Resource maintains a credential type Property that supports several  
1956 cryptographic keys and other information used for authentication and data protection. The  
1957 credential types supported include symmetric pair-wise key, group symmetric group key,  
1958 asymmetric signing key, asymmetric signing key with certificate and shared-secret (i.e. PIN or  
1959 password). The Device shall always support symmetric pair-wise key and asymmetric signing key  
1960 with certificate credential types. Other credential types are optional.

### 1961 **9.3.2 Pair-wise Symmetric Key Credentials**

1962 The CMS shall provision exactly one other pair-wise symmetric credential to a peer Device. The  
1963 CMS should not store pair-wise symmetric keys it provisions to managed Devices.

1964 Pair-wise keys could be established through ad-hoc key agreement protocols.

1965 The "PrivateData" Property in the "/oic/sec/cred" Resource contains the symmetric key.

1966 The "PublicData" Property may contain a token encrypted to the peer Device containing the pair-  
1967 wise key.

1968 The "OptionalData" Property may contain revocation status.

1969 The Device implementer should apply hardened key storage techniques that ensure the  
1970 "PrivateData" remains private.

1971 The Device implementer should apply appropriate integrity, confidentiality and access protection  
1972 of the "/oic/sec/cred", "/oic/sec/roles", "/oic/sec/csr" Resources to prevent unauthorized  
1973 modifications.

### 1974 **9.3.3 Group Symmetric Key Credentials**

1975 Group keys are symmetric keys shared among a group of Devices (3 or more). Group keys are  
1976 used for efficient sharing of data among group participants.

1977 Group keys do not provide authentication of Devices but only establish membership in a group.

1978 The CMS shall provision group symmetric key credentials to the group members. The CMS  
1979 maintains the group memberships.

1980 The "PrivateData" Property in the "/oic/sec/cred" Resource contains the symmetric key.

1981 The "PublicData" Property may contain the group name.

1982 The "OptionalData" Property may contain revocation status.

1983 The Device implementer should apply hardened key storage techniques that ensure the  
1984 "PrivateData" remains private.

1985 The Device implementer should apply appropriate integrity, confidentiality and access protection  
1986 of the "/oic/sec/cred", "/oic/sec/roles", "/oic/sec/csr" Resources to prevent unauthorized  
1987 modifications.

1988 **9.3.4 Asymmetric Authentication Key Credentials**

1989 **9.3.4.1 Asymmetric Authentication Key Credentials General**

1990 Asymmetric authentication key credentials contain either a public and private key pair or only a  
1991 public key. The private key is used to sign Device authentication challenges. The public key is used  
1992 to verify a device authentication challenge-response.

1993 The "PrivateData" Property in the "/oic/sec/cred" Resource contains the private key.

1994 The "PublicData" Property contains the public key.

1995 The "OptionalData" Property may contain revocation status.

1996 The Device implementer should apply hardened key storage techniques that ensure the  
1997 "PrivateData" remains private.

1998 Devices should generate asymmetric authentication key pairs internally to ensure the private key  
1999 is only known by the Device. See 9.3.4.2 for when it is necessary to transport private key material  
2000 between Devices.

2001 The Device implementer should apply appropriate integrity, confidentiality and access protection  
2002 of the "/oic/sec/cred", "/oic/sec/roles", "/oic/sec/csr" Resources to prevent unauthorized  
2003 modifications.

2004 **9.3.4.2 External Creation of Asymmetric Authentication Key Credentials**

2005 Devices should employ industry-standard high-assurance techniques when allowing off-device key  
2006 pair creation and provisioning. Use of such key pairs should be minimized, particularly if the key  
2007 pair is immutable and cannot be changed or replaced after provisioning.

2008 When used as part of onboarding, these key pairs can be used to prove the Device possesses the  
2009 manufacturer-asserted properties in a certificate to convince a DOTS or a user to accept  
2010 onboarding the Device. See 7.3.3 for the OTM that uses such a certificate to authenticate the  
2011 Device, and then provisions new OCF Security Domain credentials for use.

2012 **9.3.5 Asymmetric Key Encryption Key Credentials**

2013 The asymmetric key-encryption-key (KEK) credentials are used to wrap symmetric keys when  
2014 distributing or storing the key.

2015 The "PrivateData" Property in the "/oic/sec/cred" Resource contains the private key.

2016 The "PublicData" Property contains the public key.

2017 The "OptionalData" Property may contain revocation status.

2018 The Device implementer should apply hardened key storage techniques that ensure the  
2019 "PrivateData" remains private.

2020 The Device implementer should apply appropriate integrity, confidentiality and access protection  
2021 of the "/oic/sec/cred", "/oic/sec/roles", "/oic/sec/csr" Resources to prevent unauthorized  
2022 modifications.

2023 **9.3.6 Certificate Credentials**

2024 Certificate credentials are asymmetric keys that are accompanied by a certificate issued by a CMS  
2025 or an external certificate authority (CA).

2026 A certificate enrolment protocol is used to obtain a certificate and establish proof-of-possession.

2027 The issued certificate is stored with the asymmetric key credential Resource.



2028 Other objects useful in managing certificate lifecycle such as certificate revocation status are  
2029 associated with the credential Resource.

2030 Either an asymmetric key credential Resource or a self-signed certificate credential is used to  
2031 terminate a path validation.

2032 The "PrivateData" Property in the "/oic/sec/cred" Resource contains the private key.

2033 The "PublicData" Property contains the issued certificate.

2034 The "OptionalData" Property may contain revocation status.

2035 The Device implementer should apply hardened key storage techniques that ensure the  
2036 PrivateData remains private.

2037 The Device implementer should apply appropriate integrity, confidentiality and access protection  
2038 of the "/oic/sec/cred", "/oic/sec/roles", "/oic/sec/csr" Resources to prevent unauthorized  
2039 modifications.

### 2040 **9.3.7 Password Credentials**

2041 The "PrivateData" Property in the "/oic/sec/cred" Resource contains the PIN, password and other  
2042 values useful for changing and verifying the password.

2043 The "PublicData" Property may contain the user or account name if applicable.

2044 The "OptionalData" Property may contain revocation status.

2045 The Device implementer should apply hardened key storage techniques that ensure the  
2046 "PrivateData" remains private.

2047 The Device implementer should apply appropriate integrity, confidentiality and access protection  
2048 of the "/oic/sec/cred", "/oic/sec/roles", "/oic/sec/csr" Resources to prevent unauthorized  
2049 modifications.

### 2050 **9.3.8 Credentials for direct provisioning an OSCORE Security Context**

2051 A credential entry with the credential type 64 is used for direct provisioning of OSCORE Security  
2052 Context parameters for use in End-to-End Security of Unicast Messages.

2053 The "privatedata" Property of the credential entry with the credential type 64 in the "/oic/sec/cred"  
2054 Resource contains the OSCORE Master Key.

2055 A credential entry with the credential type 64 shall expose the OSCORE Configuration ("oscore")  
2056 Property, which includes:

- 2057 – The "senderid" Property containing the OSCORE Sender ID parameter.
- 2058 – The "recipientid" Property containing the OSCORE Recipient ID parameter.
- 2059 – The "ssn" Property contains a read-only value used to store the OSCORE Sender Sequence  
2060 Number.

2061 NOTE: values of "senderid" and "recipientid" are expected to be lowercase hexadecimal encoded with "0x" encoding  
2062 prefix omitted.

2063 See clause 16.2 for description of the OSCORE parameters.

### 2064 **9.3.9 Credentials for Simple Secure Multicast**

2065 There are two distinct credential types used for provisioning OSCORE Security Context parameters  
2066 used in Simple Secure Multicast (SSM): one for the SSM Client Context identified using  
2067 "credtype" : "128"; and one for the SSM Server Context identified using "credtype" : "256". In a

2068 Client of an SSM Group, the Client's OSCORE Security Context (Sender context) is derived from  
2069 a provisioned SSM Client Context. In the Servers of an SSM Group, the Server's OSCORE Security  
2070 Context (Recipient Context) is derived from a provisioned SSM Server Context.

2071 For both of these credential types, the "privatedata" Property of the credential entry in the  
2072 "/oic/sec/cred" Resource contains the value of the OSCORE Master Secret of the SSM Group,  
2073 which is generated by the OBT.

2074 A SSM Client Context credential entry shall expose the OSCORE Configuration ("oscore") Property,  
2075 which for this credential type shall include:

- 2076 – The "senderid" Property containing the OSCORE Sender ID parameter.
  - 2077 – This value is selected and provisioned by the OBT.
- 2078 – The "desc" Property containing a description of the usage of the security context
  - 2079 – This Property contains a human-readable description intended for identifying the
  - 2080 corresponding SSM Group when a Security Domain contains multiple SSM Groups.
  - 2081 – This value is selected and provisioned by the OBT
- 2082 – The "ssn" Property contains a read-only value used to store the OSCORE Sender Sequence  
2083 Number.

2084 NOTE 1: The value of "senderid" is expected to be lowercase hexadecimal encoded with "0x" encoding prefix omitted.

2085 An SSM Server Context credential entry shall include the OSCORE Configuration ("oscore")  
2086 Property, which shall include:

- 2087 – The "recipientid" Property containing the OSCORE Group Recipient ID parameter.
  - 2088 – This value is equal for all Servers in the SSM Group, and is the same as the value of the
  - 2089 "senderid" of the Client Context for the SSM Group
  - 2090 – This value is selected and provisioned by the OBT
- 2091 – The "desc" Property containing a description of the usage of the security context
  - 2092 – This Property contains a human-readable description intended for identifying the
  - 2093 corresponding SSM Group when a Security Domain contains multiple SSM Groups.
  - 2094 – This value is selected and provisioned by the OBT

2095 NOTE 2: The value of "recipientid" is expected to be lowercase hexadecimal encoded with "0x" encoding prefix omitted.

2096 See clause 16.3.3 for description of the OSCORE parameters used in SSM.

## 2097 **9.4 Certificate Based Key Management**

### 2098 **9.4.1 Overview**

2099 To achieve authentication and transport security during communications in OCF Security Domain,  
2100 certificates containing public keys of communicating parties and private keys can be used.

2101 The certificate and private key may be issued by a local or remote certificate authority (CA).

2102 The OCF certificate format is a subset of X.509 format, only elliptic curve algorithm and PEM  
2103 encoding format are allowed, most of optional fields in X.509 are not supported so that the format  
2104 intends to meet the constrained Device's requirement.

2105 The CMS manages the certificate lifecycle for certificates it issues. The DOTS assigns a CMS to a  
2106 Device when it is newly onboarded.

## 9.4.2 X.509 Digital Certificate Profiles

### 9.4.2.1 Digital Certificate Profile General

An OCF certificate format is a subset of X.509 format (version 3 or above) as defined in IETF RFC 5280.

This clause develops a profile to facilitate the use of X.509 certificates within OCF applications for those communities wishing to make use of X.509 technology. The X.509 v3 certificate format is described in detail, with additional information regarding the format and semantics of OCF specific extension(s). The supported standard certificate extensions are also listed.

**Certificate Format:** The OCF certificate profile is derived from IETF RFC 5280. However, this document does not support the "issuerUniqueID" and "subjectUniqueID" fields which are deprecated and shall not be used in the context of OCF. If these fields are present in a certificate, compliant entities shall ignore their contents.

**Certificate Encoding:** Conforming entities shall use the Privacy-Enhanced Mail (PEM) to encode certificates.

**Certificates Hierarchy and Crypto Parameters.** OCF supports a three-tier hierarchy for its Public Key Infrastructure (i.e., a Root CA, an Intermediate CA, and EE certificates). OCF accredited CAs SHALL use Elliptic Curve Cryptography (ECC) keys (secp256r1 – OID:1.2.840.10045.3.1.7) and use the ecdsaWithSHA256 (OID:1.2.840.10045.4.3.2) algorithm for certificate signatures. Elliptic Curve Cryptography public keys shall be encoded using uncompressed Elliptic Curve points.

The following clauses specify the supported standard and custom extensions for the OCF certificates profile.

### 9.4.2.2 Certificate Profile and Fields

#### 9.4.2.2.1 Root CA Certificate Profile

Table 8 describes X.509 v1 fields required for Root CA Certificates.

**Table 8 – X.509 v1 fields for Root CA Certificates**

| V1 Field                | Value / Remarks  |
|-------------------------|--|
| signatureAlgorithm      | ecdsa-with-SHA256 (OID: 1.2.840.10045.4.3.2)   |
| Version                 | v3 (value is 2)  |
| SerialNumber            | SHALL be a positive integer, unique among all certificates issued by a given CA  |
| Issuer                  | SHALL match the Subject field  |
| Subject                 | SHALL match the Issuer field   |
| notBefore               | The time at which the Root CA Certificate was generated.<br>See 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.   |
| notAfter                | No stipulation for expiry date.<br>See 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.  |
| Subject Public Key Info | id-ecPublicKey (OID: 1.2.840.10045.2.1)<br>secp256r1 (OID:1.2.840.10045.3.1.7)<br>Elliptic Curve Cryptography public keys shall be encoded using uncompressed Elliptic Curve points. |

Table 9 describes X.509 v3 extensions required for Root CA Certificates.

2133

**Table 9 - X.509 v3 extensions for Root CA Certificates**

| Extension              | Required/Optional | Criticality  | Value / Remarks   |
|------------------------|-------------------|--------------|---|
| authorityKeyIdentifier | OPTIONAL          | Non-critical | N/A   |
| subjectKeyIdentifier   | OPTIONAL          | Non-critical | N/A   |
| keyUsage               | REQUIRED          | Critical     | keyCertSign (5) & cRLSign (6) bits shall be enabled.<br>digitalSignature(0) bit may be enabled.<br>All other bits shall not be enabled. |
| basicConstraints       | REQUIRED          | Critical     | cA = TRUE<br>pathLenConstraint = not present (unlimited)  |

2134 **9.4.2.2.2 Intermediate CA Certificate Profile**

2135 Table 10 describes X.509 v1 fields required for Intermediate CA Certificates.

2136 **Table 10 - X.509 v1 fields for Intermediate CA Certificates**

| V1 Field                | Value / Remarks   |
|-------------------------|---|
| signatureAlgorithm      | ecdsa-with-SHA256 (OID: 1.2.840.10045.4.3.2)  |
| Version                 | v3 (value is 2)   |
| SerialNumber            | SHALL be a positive integer, unique among all certificates issued by Root CA  |
| Issuer                  | SHALL match the Subject field of the issuing Root CA  |
| Subject                 | (no stipulation)  |
| notBefore               | The time at which the Intermediate CA Certificate was generated.<br>See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.                           |
| notAfter                | No stipulation for expiry date.<br>See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.  |
| Subject Public Key Info | id-ecPublicKey (OID: 1.2.840.10045.2.1)<br>secp256r1 (OID: 1.2.840.10045.3.1.7)<br>Elliptic Curve Cryptography public keys shall be encoded using uncompressed Elliptic Curve points. |

2137 Table 11 **describes** X.509 v3 extensions required for Intermediate CA Certificates.2138 **Table 11 – X.509 v3 extensions for Intermediate CA Certificates**

| Extension              | Required/Optional | Criticality  | Value / Remarks   |
|------------------------|-------------------|--------------|---|
| authorityKeyIdentifier | OPTIONAL          | Non-critical | N/A   |
| subjectKeyIdentifier   | OPTIONAL          | Non-critical | N/A   |
| keyUsage               | REQUIRED          | Critical     | keyCertSign (5) & cRLSign (6) bits shall be enabled.<br>digitalSignature (0) bit may be enabled<br>All other bits shall not be enabled. |
| basicConstraints       | REQUIRED          | Critical     | cA = TRUE   |

|                            |          |              |   |
|----------------------------|----------|--------------|---|
|                            |          |              | pathLenConstraint = 0<br>(can only sign End-Entity certs)                                 |
| certificatePolicies        | OPTIONAL | Non-critical | (no stipulation)  |
| cRLDistributionPoints      | OPTIONAL | Non-critical | 1 or more URIs where the Certificate Revocation List (CRL) from the Root can be obtained. |
| authorityInformationAccess | OPTIONAL | Non-critical | OCSP URI – the URI of the Root CA's OCSP Responder  |

#### 9.4.2.2.3 End-Entity Black Certificate Profile

Table 12 describes X.509 v1 fields required for End-Entity Certificates used for Black security profile.

**Table 12 – X.509 v1 fields for End-Entity Certificates**

| V1 Field                | Value / Remarks  |
|-------------------------|--|
| signatureAlgorithm      | ecdsa-with-SHA256 (OID: 1.2.840.10045.4.3.2)   |
| Version                 | v3 (value is 2)  |
| SerialNumber            | SHALL be a positive integer, unique among all certificates issued by the Intermediate CA   |
| Issuer                  | SHALL match the Subject field of the issuing Intermediate CA   |
| Subject                 | Subject DN shall include:<br>o=OCF-verified device manufacturer organization name.<br><br>The Subject DN may include other attributes (e.g. cn, c, ou, etc.) with no stipulation by OCF. |
| notBefore               | The time at which the End-Entity Certificate was generated.<br>See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.                                   |
| notAfter                | No stipulation.<br>See clause 10.4.5 for details around IETF RFC 5280-compliant validity field formatting.   |
| Subject Public Key Info | id-ecPublicKey (OID: 1.2.840.10045.2.1)<br>secp256r1 (OID:1.2.840.10045.3.1.7)<br>Elliptic Curve Cryptography public keys shall be encoded using uncompressed Elliptic Curve points.     |

Table 13 describes X.509 v3 extensions required for End-Entity Certificates.

**Table 13 – X.509 v3 extensions for End-Entity Certificates**

| Extension              | Required/ Optional | Criticality  | Value / Remarks  |
|------------------------|--------------------|--------------|--|
| authorityKeyIdentifier | OPTIONAL           | Non-critical | N/A  |
| subjectKeyIdentifier   | OPTIONAL           | Non-critical | N/A  |
| keyUsage               | REQUIRED           | Critical     | digitalSignature (0) and keyAgreement(4) bits SHALL be the only bits enabled |
| basicConstraints       | OPTIONAL           | Non-Critical | cA = FALSE   |

|                        |                                   |              |   |
|------------------------|-----------------------------------|--------------|---|
|                        |                                   |              | pathLenConstraint = not present   |
| certificatePolicies    | OPTIONAL                          | Non-critical | <p>End-Entity certificates chaining to an OCF Root CA SHOULD contain at least one PolicyIdentifierId set to the OCF Certificate Policy OID – (1.3.6.1.4.1.51414.0.1.2) corresponding to the version of the OCF Certificate Policy under which it was issued. Additional manufacturer-specific CP OIDs may also be populated.</p>  |
| extendedKeyUsage       | REQUIRED                          | Non-critical | <p>The following extendedKeyUsage (EKU) OIDs SHALL both be present:</p> <ul style="list-style-type: none"> <li>• serverAuthentication - 1.3.6.1.5.5.7.3.1</li> <li>• clientAuthentication - 1.3.6.1.5.5.7.3.2</li> </ul> <p>Exactly ONE of the following OIDs SHALL be present:</p> <ul style="list-style-type: none"> <li>• Identity certificate - 1.3.6.1.4.1.44924.1.6</li> <li>• Role certificate - 1.3.6.1.4.1.44924.1.7</li> </ul> <p>End-Entity certificates SHALL NOT contain the anyExtendedKeyUsage OID (2.5.29.37.0)</p>   |
| subjectAlternativeName | REQUIRED UNDER CERTAIN CONDITIONS | Non-critical | <p>The subjectAltName extension is used to encode one or more Role ID values in role certificates, binding the roles to the subject public key.</p> <p>When the extendedKeyUsage (EKU) extension contains the Identity Certificate OID (1.3.6.1.4.1.44924.1.6), the subjectAltName extension SHOULD NOT be present.</p> <p>If the EKU extension contains the Role Certificate OID (1.3.6.1.4.1.44924.1.7), the subjectAltName extension SHALL be present and populated as follows:</p> <p>Each GeneralName in the GeneralNames SEQUENCE which encodes a role shall be a directoryName, which is of type Name. Name is an X.501 Distinguished Name. Each Name shall contain exactly one CN (Common Name) component, and zero or one OU (Organizational Unit) components. The OU component, if present, shall</p> |

|                                      |          |              |   |
|--------------------------------------|----------|--------------|---|
|                                      |          |              | specify the authority that defined the semantics of the role. If the OU component is absent, the certificate issuer has defined the role. The CN component shall encode the role ID. Other GeneralName types in the SEQUENCE may be present, but shall not be interpreted as roles. The role, and authority shall be encoded as ASN.1 PrintableString type, the restricted character set [0-9a-z-A-z '()+,./:=?]. |
| cRLDistributionPoints                | OPTIONAL | Non-critical | 1 or more URIs where the Certificate Revocation List (CRL) from the Intermediate CA can be obtained.  |
| authorityInformationAccess           | OPTIONAL | Non-critical | OCSP URI – the URI of the Intermediate CA's OCSP Responder  |
| OCF Compliance                       | OPTIONAL | Non-critical | See 9.4.2.2.4   |
| Manufacturer Usage Description (MUD) | OPTIONAL | Non-critical | Contains a single Uniform Resource Locator (URL) that points to an on-line Manufacturer Usage Description concerning the certificate subject. See 9.4.2.2.5   |
| OCF Security Claims                  | OPTIONAL | Non-critical | Contains a list of security claims above those required by this OCF Compliance version or Security Profile. See 9.4.2.2.6   |
| OCF CPL Attributes                   | OPTIONAL | Non-critical | Contains the list of OCF Attributes used to perform OCF Certified Product List lookups  |

#### 9.4.2.2.4 OCF Compliance X.509v3 Extension

The OCF Compliance Extension defines required parameters to correctly identify the type of Device, its manufacturer, its OCF Version, and the Security Profile compliance of the device.

The extension carries an "ocfVersion" field which provides the specific base version of the OCF documents the device implements. The "ocfVersion" field shall contain a sequence of three integers ("major", "minor", and "build"). For example, if an entity is certified to be compliant with OCF specifications 1.3.2, then the "major", "minor", and "build" fields of the "ocfVersion" will be set to "1", "3", and "2" respectively. The "ocfVersion" may be used by Security Profiles to denote compliance to a specified base version of the OCF documents.

The "securityProfile" field shall carry the ocfSecurityProfile OID(s) (clause 14.8.3) of one or more supported Security Profiles associated with the certificate in string form (UTF-8). All Security Profiles associated with the certificate should be identified by this field.

The extension shall also carry two string fields (UTF-8): "DeviceName" and "deviceManufacturer". The fields carry human-readable descriptions of the Device's name and manufacturer, respectively.

The ASN.1 definition of the OCFCCompliance extension (OID – 1.3.6.1.4.1.51414.1.0) is defined as follows:

```
id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)
```

```

2162         private(4) enterprise(1) OCF(51414) }
2163
2164     id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }
2165
2166     id-ocfCompliance OBJECT IDENTIFIER ::= { id-ocfX509Extensions 0 }
2167
2168     ocfVersion ::= SEQUENCE {
2169         major    INTEGER,
2170             --Major version number
2171         minor    INTEGER,
2172             --Minor version number
2173         build    INTEGER,
2174             --Build/Micro version number
2175     }
2176
2177     ocfCompliance ::= SEQUENCE {
2178         version          ocfVersion,
2179             --Device/OCF version
2180         securityProfile  SEQUENCE SIZE (1..MAX) OF ocfSecurityProfileOID,
2181             --Sequence of OCF Security Profile OID strings
2182             --Clause 14.8.2 defines valid ocfSecurityProfileOIDs
2183         deviceName       UTF8String,
2184             --Name of the device
2185         deviceManufacturer UTF8String,
2186             --Human-Readable Manufacturer
2187             --of the device
2188     }

```

#### 2189 **9.4.2.2.5 Manufacturer Usage Description (MUD) X.509v3 Extension**

2190 The goal of the Manufacturer Usage Description (MUD) extension is to provide a means for devices  
2191 to signal to the network the access and network functionality they require to properly function.  
2192 Access controls can be more easily achieved and deployed at scale when the MUD extension is  
2193 used.

2194 The MUD X.509 v3 extension is specified in IETF RFC 8520 with the full ASN.1 definition in clause  
2195 11.

#### 2196 **9.4.2.2.6 OCF Security Claims X.509v3 Extension**

2197 The OCF Security Claims Extension defines a list of OIDs representing security claims that the  
2198 manufacturer/integrator is making as to the security posture of the device above those required by  
2199 the OCF Compliance version or that of the OCF Security Profile being indicated by the device.

2200 The purpose of this extension is to allow for programmatic evaluation of assertions made about  
2201 security to enable some platforms/policies/administrators to better understand what is being  
2202 onboarded or challenged.

2203 The ASN.1 definition of the OCF Security Claims extension (OID – 1.3.6.1.4.1.51414.1.1) is defined  
2204 as follows:

```

2205     id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)
2206                                     private(4) enterprise(1) OCF(51414) }
2207
2208     id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }
2209
2210     id-ocfSecurityClaims OBJECT IDENTIFIER ::= { id-ocfX509Extensions 1 }
2211
2212         claim-secure-boot          ::= ocfSecurityClaimsOID { id-ocfSecurityClaims 0 }
2213         --Device claims that the boot process follows a procedure trusted
2214         --by the firmware and the BIOS
2215
2216         claim-hw-backed-cred-storage ::= ocfSecurityClaimsOID { id-ocfSecurityClaims 1 }

```



```

2217         --Device claims that credentials are stored in a specialized hardware
2218         --protection environment such as a Trusted Platform Module (TPM) or
2219         --similar mechanism.
2220
2221         ocfSecurityClaimsOID ::= OBJECT IDENTIFIER
2222
2223         ocfSecurityClaims ::= SEQUENCE SIZE (1..MAX) of ocfSecurityClaimsOID
2224
2225 9.4.2.2.7 OCF Certified Product List Attributes X.509v3 Extension
2226 The OCF Certified Product List Extension defines required parameters to utilize the OCF
2227 Compliance Management System Certified Product List (OCMS-CPL). This clause is only
2228 applicable if you plan to utilize the OCMS-CPL. The OBT may make use of these attributes to verify
the compliance level of a device.
2229
2230 The extension carries the OCF CPL Attributes: IANA Private Enterprise Number (PEN), Model and
Version.
2231
2232 The 'cpl-at-IANAPen' IANA Private Enterprise Number (PEN) provides the manufacturer's unique
2233 PEN established in the IANA PEN list located at: https://www.iana.org/assignments/enterprise-
2234 numbers. The 'cpl-at-IANAPen' field found in end-products shall be the same information as
reported during OCF Certification.
2235
2236 The 'cpl-at-model' represents an OCF-Certified product's model name. The 'cpl-at-model' field
found in end-products shall be the same information as reported during OCF Certification.
2237
2238 The 'cpl-at-version' represents an OCF-Certified product's version. The 'cpl-at-version' field found
in end-products shall be the same information as reported during OCF Certification.
2239
2240 The ASN.1 definition of the OCF CPL Attributes extension (OID – 1.3.6.1.4.1.51414.1.2) is defined
as follows:
2241
2242 id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)
2243                                private(4) enterprise(1) OCF(51414) }
2244
2245 id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }
2246
2247 id-ocfCPLAttributes OBJECT IDENTIFIER ::= { id-ocfX509Extensions 2 }
2248
2249 cpl-at-IANAPen ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 0 }
2250 cpl-at-model ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 1 }
2251 cpl-at-version ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 2 }
2252
2253 ocfCPLAttributes ::= SEQUENCE {
2254     cpl-at-IANAPen UTF8String,
2255     --Manufacturer's registered IANA Private Enterprise Number
2256     cpl-at-model UTF8String,
2257     --Device OCF Security Profile
2258     cpl-at-version UTF8String
2259     --Name of the device
2260 }
2261
9.4.2.3 Supported Certificate Extensions
2262 As these certificate extensions are a standard part of IETF RFC 5280, this document includes the
2263 clause number from that RFC to include it by reference. Each extension is summarized here, and
2264 any modifications to the RFC definition are listed. Devices MUST implement and understand the
2265 extensions listed here; other extensions from the RFC are not included in this document and
2266 therefore are not required. 10.4 describes what Devices must implement when validating certificate
2267 chains, including processing of extensions, and actions to take when certain extensions are absent.

```

2268 – Authority Key Identifier (4.2.1.1)

2269 The Authority Key Identifier (AKI) extension provides a means of identifying the public key  
2270 corresponding to the private key used to sign a certificate. This document makes the following  
2271 modifications to the referenced definition of this extension:

2272 The "authorityCertIssuer" or "authorityCertSerialNumber" fields of the "AuthorityKeyIdentifier"  
2273 sequence are not permitted; only "keyIdentifier" is allowed. This results in the following  
2274 grammar definition:

2275 id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 35 }

2276  
2277 AuthorityKeyIdentifier ::= SEQUENCE {  
2278     keyIdentifier                     [0] KeyIdentifier                     }  
2279

2280 KeyIdentifier ::= OCTET STRING

2281 – Subject Key Identifier (4.2.1.2)

2282 The Subject Key Identifier (SKI) extension provides a means of identifying certificates that  
2283 contain a particular public key.

2284 This document makes the following modification to the referenced definition of this extension:

2285 Subject Key Identifiers SHOULD be derived from the public key contained in the certificate's  
2286 "SubjectPublicKeyInfo" field or a method that generates unique values. This document  
2287 RECOMMENDS the 256-bit SHA-2 hash of the value of the BIT STRING "subjectPublicKey"  
2288 (excluding the tag, length, and number of unused bits). Devices verifying certificate chains must  
2289 not assume any particular method of computing key identifiers, however, and must only base  
2290 matching AKI's and SKI's in certification path constructions on key identifiers seen in certificates.

2291 – Subject Alternative Name

2292 If the EKU extension is present, and has the value XXXXXX, indicating that this is a role  
2293 certificate, the Subject Alternative Name (subjectAltName) extension shall be present and  
2294 interpreted as described below. When no EKU is present, or has another value, the  
2295 "subjectAltName" extension SHOULD be absent. The "subjectAltName" extension is used to  
2296 encode one or more Role ID values in role certificates, binding the roles to the subject public  
2297 key. The "subjectAltName" extension is defined in IETF RFC 5280 (See 4.2.1.6):

2298 id-ce-subjectAltName OBJECT IDENTIFIER ::= { id-ce 17 }

2299  
2300 SubjectAltName ::= GeneralNames

2301  
2302 GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

2303  
2304 GeneralName ::= CHOICE {  
2305     otherName                             [0]     OtherName,  
2306     rfc5322Name                         [1]     IA5String,  
2307     dNSName                              [2]     IA5String,  
2308     x400Address                         [3]     ORAddress,  
2309     directoryName                        [4]     Name,  
2310     ediPartyName                         [5]     EDIPartyName,  
2311     uniformResourceIdentifier           [6]     IA5String,  
2312     iPAddress                            [7]     OCTET STRING,  
2313     registeredID                         [8]     OBJECT IDENTIFIER }  
2314

2315     EDIPartyName ::= SEQUENCE {  
2316         nameAssigner                     [0]     DirectoryString OPTIONAL,  
2317         partyName                        [1]     DirectoryString }  
2318

2319 Each "GeneralName" in the "GeneralNames" SEQUENCE which encodes a role shall be a  
2320 "directoryName", which is of type Name. Name is an X.501 Distinguished Name. Each Name  
2321 shall contain exactly one CN (Common Name) component, and zero or one OU (Organizational  
2322 Unit) components. The OU component, if present, shall specify the authority that defined the

2323 semantics of the role. If the OU component is absent, the certificate issuer has defined the role.  
 2324 The CN component shall encode the role ID. Other "GeneralName" types in the SEQUENCE  
 2325 may be present, but shall not be interpreted as roles. Therefore, if the certificate issuer includes  
 2326 non-role names in the "subjectAltName" extension, the extension should not be marked critical.

2327 The role, and authority need to be encoded as ASN.1 "PrintableString" type, the restricted  
 2328 character set [0-9a-z-A-z '()+, -./:=?].

2329 – Key Usage (4.2.1.3)

2330 The key usage extension defines the purpose (e.g., encipherment, signature, certificate signing)  
 2331 of the key contained in the certificate. The usage restriction might be employed when a key that  
 2332 could be used for more than one operation is to be restricted.

2333 This document does not modify the referenced definition of this extension.

2334 – Basic Constraints (4.2.1.9)

2335 The basic constraints extension identifies whether the subject of the certificate is a CA and the  
 2336 maximum depth of valid certification paths that include this certificate. Without this extension,  
 2337 a certificate cannot be an issuer of other certificates.

2338 This document does not modify the referenced definition of this extension.

2339 – Extended Key Usage (4.2.1.12)

2340

2341 Extended Key Usage describes allowed purposes for which the certified public key may can be  
 2342 used. When a Device receives a certificate, it determines the purpose based on the context of  
 2343 the interaction in which the certificate is presented, and verifies the certificate can be used for  
 2344 that purpose.

2345 This document makes the following modifications to the referenced definition of this extension:

2346 CAs SHOULD mark this extension as critical.

2347 CAs MUST NOT issue certificates with the anyExtendedKeyUsage OID (2.5.29.37.0).

2348

2349 The list of OCF-specific purposes and the assigned OIDs to represent them are:

2350 – Identity certificate 1.3.6.1.4.1.44924.1.6

2351 – Role certificate 1.3.6.1.4.1.44924.1.7

2352 **9.4.2.4 Cipher Suite for Authentication, Confidentiality and Integrity**

2353 OCF compliant entities shall support TLS version 1.2. Compliant entities shall support  
 2354 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8 cipher suite as defined in IETF RFC 7251 and may  
 2355 support additional ciphers as defined in the TLS v1.2 specifications.

2356 **9.4.2.5 Encoding of Certificate**

2357 See 9.4.2 for details.

2358 **9.4.3 Certificate Revocation List (CRL) Profile [Deprecated]**

2359 This clause is intentionally left blank.

2360 **9.4.4 Resource Model**

2361 Device certificates and private keys are kept in "cred" Resource.

2362 The "cred" Resource contains the certificate information pertaining to the Device. The "PublicData"  
 2363 Property holds the device certificate and CA certificate chain. "PrivateData" Property holds the  
 2364 Device private key paired to the certificate. (See 13.3 for additional detail regarding the  
 2365 "/oic/sec/cred" Resource).

### 9.4.5 Certificate Provisioning

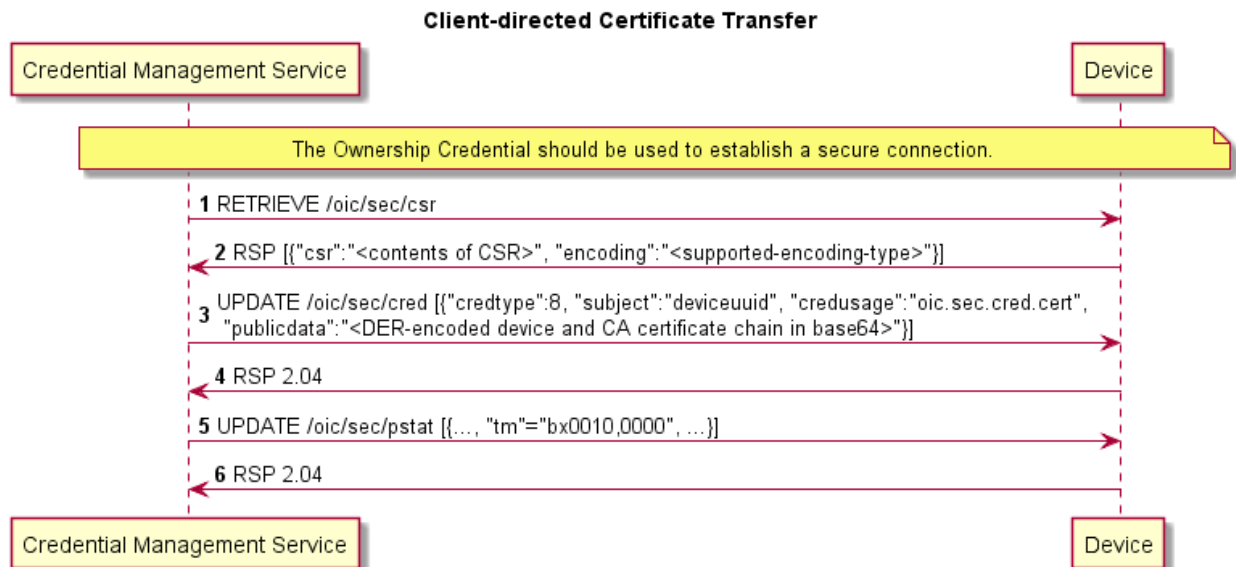
The CMS (e.g. a hub or a smart phone) issues certificates for new Devices.

The CA in the CMS retrieves a Device's public key and proof of possession of the private key, generates a Device's certificate signed by this CA certificate, and then the CMS transfers them to the Device including its CA certificate chain. Optionally, the CMS can also transfer one or more role certificates, which shall have the format described in clause 9.4.2. The "subjectPublicKey" of each role certificate shall match the "subjectPublicKey" in the Device certificate.

In the sequence in Figure 23, the Certificate Signing Request (CSR) is defined by PKCS#10 in IETF RFC 2986, and is included here by reference.

The sequence flow of a certificate transfer for a Client-directed model is described in Figure 23.

- 1) The CMS retrieves a CSR from the Device that requests a certificate. In this CSR, the Device shall place its requested UUID into the subject and its public key in the "SubjectPublicKeyInfo". The Device determines the public key to present; this may be an already-provisioned key it has selected for use with authentication, or if none is present, it may generate a new key pair internally and provide the public part. The key pair shall be compatible with the allowed cipher suites listed in 9.4.2.4 and 11.3.4, since the certificate will be restricted for use in OCF authentication.
- 2) Alternatively, the CMS generates and provisions a private key and corresponding certificate directly to the Device.
- 3) The CMS transfers the issued certificate and CA chain to the designated Device using the same credid, to maintain the association with the private key. The credential type ("oic.sec.cred") used to transfer certificates in Figure 23 is also used to transfer role certificates, by including multiple credentials in the POST from CMS to Device. Identity certificates shall be stored with the credusage Property set to "oic.sec.cred.cert" and role certificates shall be stored with the credusage Property set to "oic.sec.cred.rolecert".



**Figure 23 – Client-directed Certificate Transfer**

### 9.4.6 CRL Provisioning [Deprecated]

This clause is intentionally left blank.

## 10 Device Authentication

### 10.1 Device Authentication General

When a Client is accessing a restricted Resource on a Server, the Server shall authenticate the Client. Clients shall authenticate Servers while requesting access. Clients may also assert one or more roles that the server can use in access control decisions. Roles may be asserted when the Device authentication is done with certificates.

### 10.2 Device Authentication with Symmetric Key Credentials

When using symmetric keys to authenticate, the Server Device shall include the "ServerKeyExchange" message and set "psk\_identity\_hint" to the Server's Device UUID. The Client shall validate that it has a credential with the Subject UUID set to the Server's Device UUID, and a credential type of PSK. If it does not, the Client shall respond with an unknown\_psk\_identity error or other suitable error.

If the Client finds a suitable PSK credential, it shall reply with a "ClientKeyExchange" message that includes a "psk\_identity" set to the Client's Device UUID. The Server shall verify that it has a credential with the matching Subject UUID and type. If it does not, the Server shall respond with an "unknown\_psk\_identity" or other suitable error code. If it does, then it shall continue with the DTLS protocol, and both Client and Server shall compute the resulting premaster secret.

### 10.3 Device Authentication with Raw Asymmetric Key Credentials

When using raw asymmetric keys to authenticate, the Client and the Server shall include a suitable public key from a credential that is bound to their Device. Each Device shall verify that the provided public key matches the Public Data field of a credential they have, and use the corresponding Subject UUID of the credential to identify the peer Device.

### 10.4 Device Authentication with Certificates

#### 10.4.1 Device Authentication with Certificates General

When using certificates to authenticate, the Client and Server shall each include their certificate chain, as stored in the appropriate credential, as part of the selected authentication cipher suite. Each Device shall validate the certificate chain presented by the peer Device. Each certificate signature shall be verified until a public key is found within the "/oic/sec/cred" Resource with the "oic.sec.cred.trustca" credusage.

Devices shall follow the certificate path validation algorithm in clause 6 of IETF RFC 5280. In addition:

- For both End-Entity certificates and non-End-Entity certificates, Devices shall verify that "notBefore" and "notAfter" fields in the certificates conform to IETF RFC 5280 clauses 4.1.2.5, 4.1.2.5.1, and 4.1.2.5.2.
- For non-End-Entity certificates, Devices shall verify that the Basic Constraints extension is present, and that the "cA" boolean in the extension is TRUE. If any of these are false, the certificate chain shall be rejected. If the pathLenConstraint field is present, Devices shall verify that the number of certificates between this certificate and the End-Entity certificate is less than or equal to "pathLenConstraint". In particular, if "pathLenConstraint" is zero, only an End-Entity certificate can be issued by this certificate. If the "pathLenConstraint" field is absent, there is no limit to the chain length.
- For End-Entity certificates, Devices shall verify that the Basic Constraints extension (if present) has a "cA" boolean value of FALSE, and does not contain a "pathLenConstraint" ASN.1 sequence.
- For non-End-Entity certificates, Devices shall verify that the Key Usage extension is present, and that the "keyCertSign" (5) bit is asserted.

- 2442 – For End-Entity certificates, Devices shall verify that the Key Usage extension is present and  
2443 that "digitalSignature" (0) and "keyAgreement" (4) bits are asserted.
- 2444 – For End-Entity certificates, Devices shall verify that the Extended Key Usage (EKU) extension  
2445 is present and suitable to the purpose for which it is being presented: Identity  
2446 ("1.3.6.1.4.1.44924.1.6") or Role ("1.3.6.1.4.1.44924.1.7"). An End-Entity certificate which  
2447 contains no EKU extension, or presents both identity and role OIDs is not valid and shall be  
2448 rejected. Any certificate which contains the "anyExtendedKeyUsage" purpose ("2.5.29.37.0")  
2449 shall be rejected, even if other valid EKUs are also present. For End-Entity certificates, Devices  
2450 shall verify that the EKU extension also contains OIDs for "serverAuthentication"  
2451 ("1.3.6.1.5.5.7.3.1") and "clientAuthentication" ("1.3.6.1.5.5.7.3.2") for compatibility with  
2452 various TLS implementations.
- 2453 – For End-Entity certificates which chain to an OCF Root CA, the Devices should verify that they  
2454 contain at least one "PolicyIdentifierId" set to the OCF Certificate Policy OID –  
2455 ("1.3.6.1.4.1.51414.0.1.2") corresponding to the version of the OCF Certificate Policy under  
2456 which it was issued. Additional manufacturer-specific CP OIDs may also be populated.

2457 If the Device does not recognize an extension, it shall examine the "critical" field. If the field is  
2458 TRUE, the Device shall reject the certificate. If the field is FALSE, the Device shall treat the  
2459 certificate as if the extension were absent and proceed accordingly. This applies to all certificates  
2460 in a chain.

2461 A Device retrieves the Subject UUID from the "Common Name" component of the "Subject Name"  
2462 property of the End-Entity certificate which has the following format: "uuid: X",, where X is  
2463 provisioned by the CMS to match the "deviceuuid" Property of the "/oic/sec/doxm" Resource. The  
2464 Device treats all requests arriving over a connection authenticated by this End-Entity certificate as  
2465 having originated from the Device with this Subject UUID. The Device shall use this Subject UUID  
2466 to match against the "subjectuuid" Property of the provisioned ACL entries to perform access  
2467 control checks.

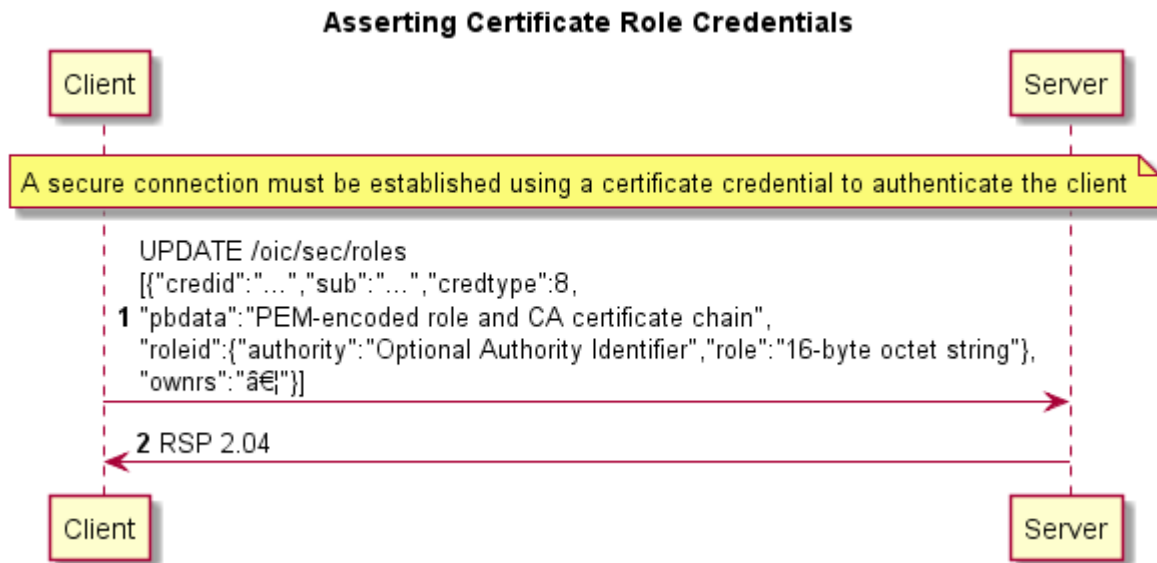
#### 2468 **10.4.2 Role Assertion with Certificates**

2469 This clause describes role assertion by a client to a server using a certificate role credential.

2470 Following authentication with a certificate, an OCF Client shall assert Roles by updating the  
2471 Server's "/oic/sec/roles" Resource with all the Role certificates it possesses, unless the device  
2472 manufacturer provides a vendor-specific mechanism for End User to select which roles to assert.  
2473 The Role credentials shall be certificate credentials and shall include a certificate chain. The Server  
2474 shall validate each certificate chain as specified in clause 10.3. Additionally, the public key in the  
2475 End-Entity certificate used for Device authentication shall be identical to the public key in all Role  
2476 (End-Entity) certificates. Also, the common name component of the subject name for both Role  
2477 certificates and identity certificates shall include a string of format "uuid:X" where X matches the  
2478 "deviceuuid" Property of the "oic.sec.doxm" Resource.

2479 Furthermore, a Client is prohibited from adding Role certificates for other Clients. The Server shall  
2480 reject Clients' request to add Role certificates if either (1) the request was received over an un-  
2481 secured connection or (2) the request was received over a secured connection but the public key  
2482 in the Role certificate does not match the public key in the identity certificate, which was used to  
2483 establish the secured connection.

2484 The Roles asserted are encoded in the "subjectAltName" extension in the certificate. The  
2485 "subjectAltName" field can have multiple values, allowing a single certificate to encode multiple  
2486 Roles that apply to the Client. The Server shall also check that the EKU extension of the Role  
2487 certificate(s) contains the value "1.3.6.1.4.1.44924.1.7" (see clause 9.4.2.2) indicating the  
2488 certificate may be used to assert Roles. Figure 24 describes how a Client Device asserts Roles to  
2489 a Server.



**Figure 24 – Asserting a role with a certificate role credential.**

Additional comments for Figure 24

- 1) The response shall contain "204 No Content" to indicate success or 4xx to indicate an error. If the server does not support certificate credentials, it should return "501 Not Implemented"
- 2) Roles asserted by the client may be kept for a duration chosen by the server. The duration shall not exceed the validity period of the role certificate.
- 3) Servers should choose a nonzero duration to avoid the cost of frequent re-assertion of a role by a client. It is recommended that servers use the validity period of the certificate as a duration, effectively allowing the CMS to decide the duration.
- 4) The format of the data sent in the create call shall be a list of credentials ("oic.sec.cred", see Table 19). They shall have "credtype" 8 (indicating certificates) and "PrivateData" field shall not be present. For fields that are duplicated in the "oic.sec.cred" object and the certificate, the value in the certificate shall be used for validation. For example, if the "Period" field is set in the credential, the server shall treat the validity period in the certificate as authoritative. Similar for the roleid data (authority, role).
- 5) Certificates shall be encoded as in Figure 23 (PEM-encoded certificate chain).
- 6) Clients may GET the "/oic/sec/roles" Resource to determine the roles that have been previously asserted. An array of credential objects shall be returned. If there are no valid certificates corresponding to the currently connected and authenticated Client's identity, then an empty array (i.e. []) shall be returned.

#### 10.4.3 OCF PKI Roots

This clause intentionally left empty.

#### 10.4.4 PKI Trust Store

Each Device using a certificate chained to an OCF Root CA trust anchor SHALL securely store the OCF Root CA certificates in the "oic/sec/cred" Resource and SHOULD physically store this Resource in a hardened memory location where the certificates cannot be tampered with.

#### 10.4.5 Path Validation and extension processing

See clause 10.3.

## 2520 **11 Message Integrity and Confidentiality**

### 2521 **11.1 Preamble**

2522 Secured communications between Clients and Servers are protected against eavesdropping,  
2523 tampering, or message replay, using security mechanisms that provide message confidentiality and  
2524 integrity.

### 2525 **11.2 Session Protection with DTLS**

#### 2526 **11.2.1 DTLS Protection General**

2527 Devices shall support DTLS for secured communications as defined in IETF RFC 6347. Devices  
2528 using TCP shall support TLS v1.2 for secured communications as defined in IETF RFC 5246. See  
2529 11.3 for a list of required and optional cipher suites for message communication.

2530 OCF Devices MUST support (D)TLS version 1.2 or greater and MUST NOT support versions 1.1  
2531 or lower.

2532 Multicast session semantics are not yet defined in this version of the security document.

#### 2533 **11.2.2 Unicast Session Semantics**

2534 For unicast messages between a Client and a Server, both Devices shall authenticate each other.  
2535 See clause 10 for details on Device Authentication.

2536 Secured unicast messages between a Client and a Server shall employ a cipher suite from 11.3.  
2537 The sending Device shall encrypt and authenticate messages as defined by the selected cipher  
2538 suite and the receiving Device shall verify and decrypt the messages before processing them.

### 2539 **11.3 Cipher Suites**

#### 2540 **11.3.1 Cipher Suites General**

2541 The cipher suites allowed for use can vary depending on the context. This clause lists the cipher  
2542 suites allowed during ownership transfer and normal operation. The following RFCs provide  
2543 additional information about the cipher suites used in OCF.

2544 IETF RFC 4279: Specifies use of pre-shared keys (PSK) in (D)TLS

2545 IETF RFC 4492: Specifies use of elliptic curve cryptography in (D)TLS

2546 IETF RFC 5489: Specifies use of cipher suites that use elliptic curve Diffie-Hellman (ECDHE) and  
2547 PSKs

2548 IETF RFC 6655 and IETF RFC 7251: Specifies AES-CCM mode cipher suites, with ECDHE

#### 2549 **11.3.2 Cipher Suites for Device Ownership Transfer**

##### 2550 **11.3.2.1 Just Works Method Cipher Suites**

2551 The Just Works OTM may use the following (D)TLS cipher suites.

2552 TLS\_ECDH\_ANON\_WITH\_AES\_128\_CBC\_SHA256

2553 All Devices supporting Just Works OTM shall implement:

2554 TLS\_ECDH\_ANON\_WITH\_AES\_128\_CBC\_SHA256 (with the value 0xFF00)

##### 2555 **11.3.2.2 Random PIN Method Cipher Suites**

2556 The Random PIN Based OTM may use the following (D)TLS cipher suites.

2557 TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256



2558 All Devices supporting Random Pin Based OTM shall implement:

2559 TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256

2560 **11.3.2.3 Certificate Method Cipher Suites**

2561 The Manufacturer Certificate Based OTM may use the following (D)TLS cipher suites.

2562 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8,

2563 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8,

2564 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM,

2565 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM

2566 Using the following curve:

2567 secp256r1 (See IETF RFC 4492)

2568 All Devices supporting Manufacturer Certificate Based OTM shall implement:

2569 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8

2570 Devices supporting Manufacturer Certificate Based OTM should implement:

2571 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8,

2572 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM,

2573 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM

2574 **11.3.3 Cipher Suites for Symmetric Keys**

2575 The following cipher suites are defined for (D)TLS communication using PSKs:

2576 TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256,

2577 TLS\_PSK\_WITH\_AES\_128\_CCM\_8, (\* 8 OCTET Authentication tag \*)

2578 TLS\_PSK\_WITH\_AES\_256\_CCM\_8,

2579 TLS\_PSK\_WITH\_AES\_128\_CCM, (\* 16 OCTET Authentication tag \*)

2580 TLS\_PSK\_WITH\_AES\_256\_CCM,

2581 All CCM based cipher suites also use HMAC-SHA-256 for authentication.

2582 All Devices shall implement the following:

2583 TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256,

2584

2585 Devices should implement the following:

2586 TLS\_ECDHE\_PSK\_WITH\_AES\_128\_CBC\_SHA256,

2587 TLS\_PSK\_WITH\_AES\_128\_CCM\_8,

2588 TLS\_PSK\_WITH\_AES\_256\_CCM\_8,

2589 TLS\_PSK\_WITH\_AES\_128\_CCM,

2590 TLS\_PSK\_WITH\_AES\_256\_CCM

2591 **11.3.4 Cipher Suites for Asymmetric Credentials**

2592 The following cipher suites are defined for (D)TLS communication with asymmetric keys or  
2593 certificates:

2594 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8,

2595 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8,  
2596 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM,  
2597 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM  
2598 Using the following curve:  
2599 secp256r1 (See IETF RFC 4492)  
2600 All Devices supporting Asymmetric Credentials shall implement:  
2601 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8  
2602 All Devices supporting Asymmetric Credentials should implement:  
2603 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8,  
2604 TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM,  
2605 TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM  
2606

## 12 Access Control

### 12.1 ACL Generation and Management

This clause intentionally left empty.

### 12.2 ACL Evaluation and Enforcement

#### 12.2.1 ACL Evaluation and Enforcement General

The Server enforces access control over application Resources before exposing them to the requestor. The Security Layer in the Server authenticates the requestor when access is received via the secure port. Authenticated requestors, known as the "subject" can be used to match ACL entries that specify the requestor's identity, role or may match authenticated requestors using a subject wildcard.

If the request arrives over the unsecured port, the only ACL policies allowed are those that use a subject wildcard match of anonymous requestors.

Access is denied if a requested Resource is not matched by an ACL entry.

NOTE There are documented exceptions pertaining to Device onboarding where access to Security Virtual Resources may be granted prior to provisioning of ACL Resources.

The second generation ACL (i.e. `/oic/sec/acl2`) contains an array of Access Control Entries (ACE2) that employ a Resource matching algorithm that uses an array of Resource references to match Resources to which the ACE2 access policy applies. Matching consists of comparing the values of the ACE2 "resources" Property (see clause 13) to the requested Resource. Resources are matched in two ways:

- 1) host reference (`"href"`)
- 2) Resource wildcard (`"wc"`).

#### 12.2.2 Host Reference Matching

When present in an ACE2 matching element, the Host Reference (`href`) Property shall be used for Resource matching.

- The `href` Property shall be used to find an exact match of the Resource name if present.

#### 12.2.3 Resource Wildcard Matching

When present, a wildcard (`"wc"`) expression shall be used to match multiple Resources using a wildcard Property contained in the `"oic.sec.ace2.resource-ref"` structure.

A wildcard expression may be used to match multiple Resources using a wildcard Property contained in the `"oic.sec.ace2.resource-ref"` structure. The wildcard matching strings are defined in Table 14.

**Table 14 – ACE2 Wildcard Matching Strings Description**

| String           | Description   |
|------------------|---|
| <code>"+"</code> | Shall match all Discoverable Non-Configuration Resources which expose at least one Secure OCF Endpoint.   |
| <code>"_"</code> | Shall match all Discoverable Non-Configuration Resources which expose at least one Unsecure OCF Endpoint. |
| <code>""</code>  | Shall match all Non-Configuration Resources.  |

NOTE Discoverable Resources appear in the `/oic/res` Resource, while non-discoverable Resources may appear in other collection Resources but do not appear in the `/res` collection.

#### 12.2.4 Multiple Criteria Matching

If the ACE2 "resources" Property contains multiple entries, then a logical OR shall be applied for each array element. For example, if a first array element of the "resources" Property contains "href"="/a/light" and the second array element of the "resources" Property contains "href"="/a/led", then Resources that match either of the two "href" criteria shall be included in the set of matched Resources.

##### Example 1 JSON for Resource matching

```
{
  //Matches Resources named "/x/door1" or "/x/door2"
  "resources":[
    {
      "href":"/x/door1"
    },
    {
      "href":"/x/door2"
    },
  ]
}
```

##### Example 2 JSON for Resource matching

```
{
  // Matches all Resources
  "resources":[
    {
      "wc":"*"
    }
  ]
}
```

#### 12.2.5 Subject Matching using Wildcards

When the ACE subject is specified as the wildcard string "\*" any requestor is matched. The OCF server may authenticate the OCF client, but is not required to.

##### Examples: JSON for subject wildcard matching

```
//matches all subjects that have authenticated and confidentiality protections in place.
"subject" : {
  "conntype" : "auth-crypt"
}

//matches all subjects that have NOT authenticated and have NO confidentiality protections in place.
"subject" : {
  "conntype" : "anon-clear"
}
```

#### 12.2.6 Subject Matching using Roles

When the ACE subject is specified as a role, a requestor shall be matched if either:

- 1) The requestor authenticated with a symmetric key credential, and the role is present in the "roleid" Property of the credential's entry in the "credential" Resource, or

2685 2) The requestor authenticated with a certificate, and a valid role certificate is present in the roles  
2686 Resource with the requestor's certificate's public key at the time of evaluation. Validating role  
2687 certificates is defined in 10.3.1.

## 2688 **12.2.7 ACL Evaluation**

### 2689 **12.2.7.1 ACE2 matching algorithm**

2690 The OCF Server shall apply an ACE2 matching algorithm that matches in the following sequence:

- 2691 1) The local "/oic/sec/acl2" Resource contributes its ACE2 entries for matching.
- 2692 2) Access shall be granted when all these criteria are met:
  - 2693 a) The requestor is matched by the ACE2 "subject" Property.
  - 2694 b) The requested Resource is matched by the ACE2 "resources" Property and the requested  
2695 Resource shall exist on the local Server.
  - 2696 c) The "period" Property constraint shall be satisfied.
  - 2697 d) The "permission" Property constraint shall be applied.

2698 If multiple ACE2 entries match the Resource request, the union of permissions, for all matching  
2699 ACEs, defines the effective permission granted. E.g. If Perm1=CR---; Perm2=--UDN; Then UNION  
2700 (Perm1, Perm2)=CRUDN.

2701 The Server shall enforce access based on the effective permissions granted.

2702 Batch requests to Resource containing Links require additional considerations when accessing the  
2703 linked Resources. ACL considerations for batch request to the Atomic Measurement Resource  
2704 Type are provided in clause 12.2.7.2. ACL considerations for batch request to the Collection  
2705 Resource Type are provided in clause 12.2.7.3.

2706 Clause 12.2.7.4 provides ACL considerations when a new Resource is created on a Server in  
2707 response to a CREATE request.

### 2708 **12.2.7.2 ACL considerations for batch request to the Atomic Measurement Resource Type**

2709 The present clause shall apply to any Resource Type based on the Atomic Measurement Resource  
2710 Type.

2711 If an OCF Server receives a batch OCF Interface request to an Atomic Measurement Resource and  
2712 there is an ACE matching the Atomic Measurement Resource which permits the request, then the  
2713 corresponding requests to the linked Resources of the Atomic Measurement Resource shall be  
2714 permitted by the OCF Server. That is, the request to each linked Resource is permitted regardless  
2715 of whether there is an ACE configured on the OCF Server which would permit a corresponding  
2716 request from the OCF Client (which sent the batch OCF Interface request to the Atomic  
2717 Measurement Resource) addressing the linked Resource.

2718 NOTE As specified in ISO/IEC 30118-1, the linked Resources of an Atomic Measurement Resource are hosted on the  
2719 same Device as the Atomic Measurement Resource.

### 2720 **12.2.7.3 ACL considerations for a batch OCF Interface request to a Collection**

2721 This clause addresses the additional authorization processes which take place when a Server  
2722 receives a batch OCF Interface request from a Client to a Collection hosted on that Server,  
2723 assuming there is an ACE matching the Collection which permits the original Client request. For  
2724 the purposes of this clause, the Server hosting this Collection is called the "Collection host". The  
2725 additional authorization process is dependent on whether the linked Resource is hosted on the  
2726 Collection host or the linked Resource is hosted on another Server:

- 2727 – For each generated request to a linked Resource hosted on the Collection host, the Collection  
2728 host shall apply the ACE2 matching algorithm in clause 12.2.7.1 to determine whether the linked  
2729 Resource is permitted to process the generated request, with the following clarifications:

- 2730       – The requestor in clause 12.2.7.1 shall be the Client which sent the original Client request.
- 2731       – The requested Resource in clause 12.2.7.1 shall be the linked Resource, which shall be
- 2732       matched using at least one of:
- 2733       – a Resource Wildcard matching the linked Resource, or
- 2734       – an exact match of the local path of the linked Resource with a "href" Property in the
- 2735       "resources" array in the ACE2.
- 2736       – an exact match of the full URI of the linked Resource with a "href" Property in the
- 2737       "resources" array in the ACE2.

2738       NOTE The full URI of a linked Resource is obtained by concatenating the "anchor" Property of the Link, if present, and

2739       the "href" Property of the Link. The local path can then be determined from the full URI.

2740       If the linked Resource is not permitted to process the generated request, then the Collection host

2741       shall treat such cases as a linked Resource which cannot process the request when composing the

2742       aggregated response to the original Client Request, as specified for the batch OCF Interface in the

2743       ISO/IEC 30118-1.

#### 2744       **12.2.7.4    ACL Considerations on creation of a new Resource**

2745       When a new Resource is created on a Server in response to a CREATE request, there might be

2746       no ACEs permitting access to the newly created Resource. The present clause describes how the

2747       Server autonomously modifies the "/oic/sec/acl2" Resource to provide some initial authorizations

2748       for accessing the newly created Resource. The purpose of this autonomous modification is to avoid

2749       relying on the AMS update the "/oic/sec/acl2" Resource after every new Resource is created.

2750       Subsequent to a Server creating a Collection inside another Collection in response to a CREATE

2751       request from a Client, and prior to sending a response to the Client:

- 2752       – If there is an ACE with "subject" containing the UUID of the Client, and "permissions" exactly
- 2753       matching the CREATE, RETRIEVE, UPDATE and DELETE operations, then the Server shall
- 2754       autonomously add an "href" entry to "resources" with the URI of the newly created Collection.
- 2755       – Otherwise, the Server shall autonomously add an ACE with "subject" containing the UUID
- 2756       of the Client, "resources" containing an "href" entry with the URI of the newly created
- 2757       Collection, and "permissions" exactly matching the CREATE, RETRIEVE, UPDATE and
- 2758       DELETE operations.

2759       Subsequent to a Server creating a non-Collection Resource inside another Collection in response

2760       to a CREATE request from a Client, and prior to sending a response to the Client:

- 2761       – If there is an ACE with "subject" containing the UUID of the Client, and "permissions" exactly
- 2762       matching the RETRIEVE, UPDATE and DELETE operations, then the Server shall
- 2763       autonomously add an "href" entry to "resources" with the URI of the newly created Resource.
- 2764       – Otherwise, the Server shall autonomously add an ACE with "subject" containing the UUID
- 2765       of the Client, "resources" containing an "href" entry with the URI of the newly created, and
- 2766       "permissions" exactly matching the RETRIEVE, UPDATE and DELETE operations.

2767

## 13 Security Resources

### 13.1 Security Resources General

OCF Security Resources are shown in Figure 25.

"/oic/sec/cred" Resource and Properties are shown in Figure 26.

"/oic/sec/acl2" Resource and Properties are shown in Figure 27.

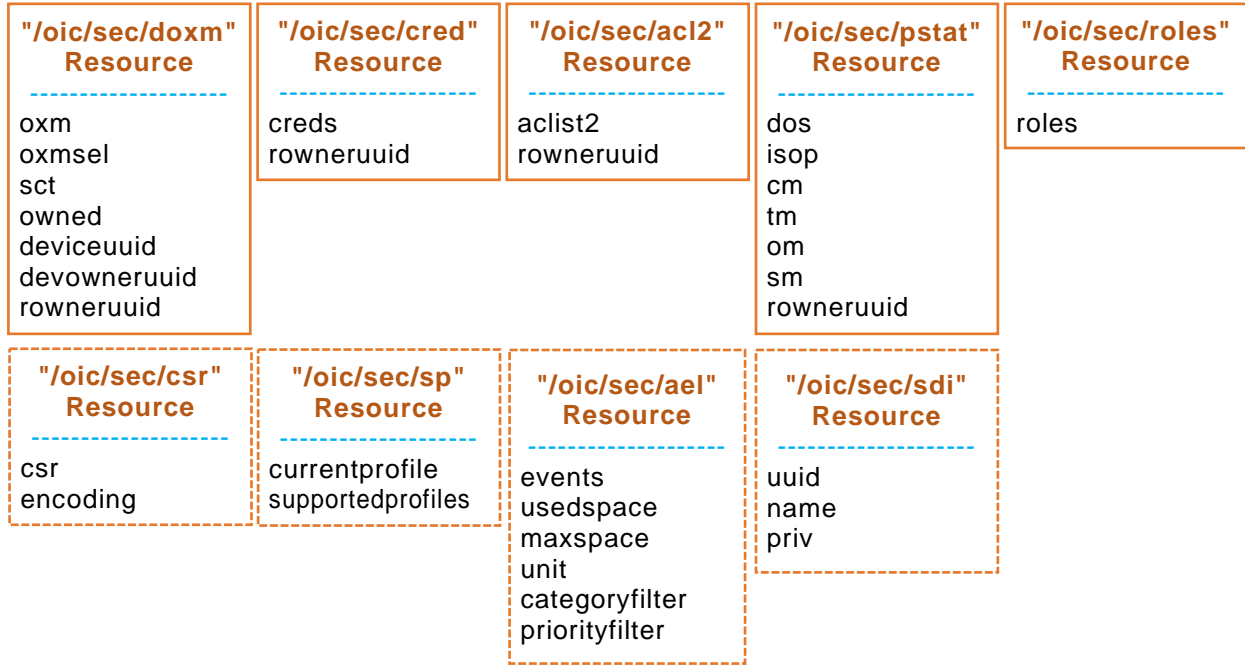
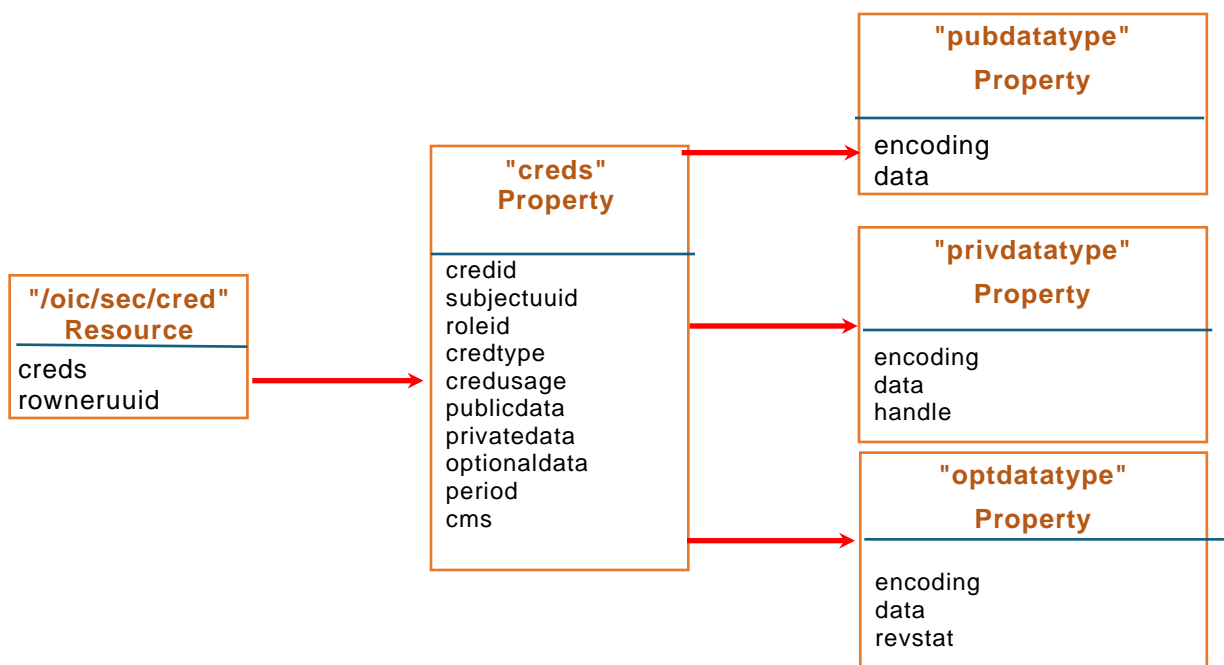
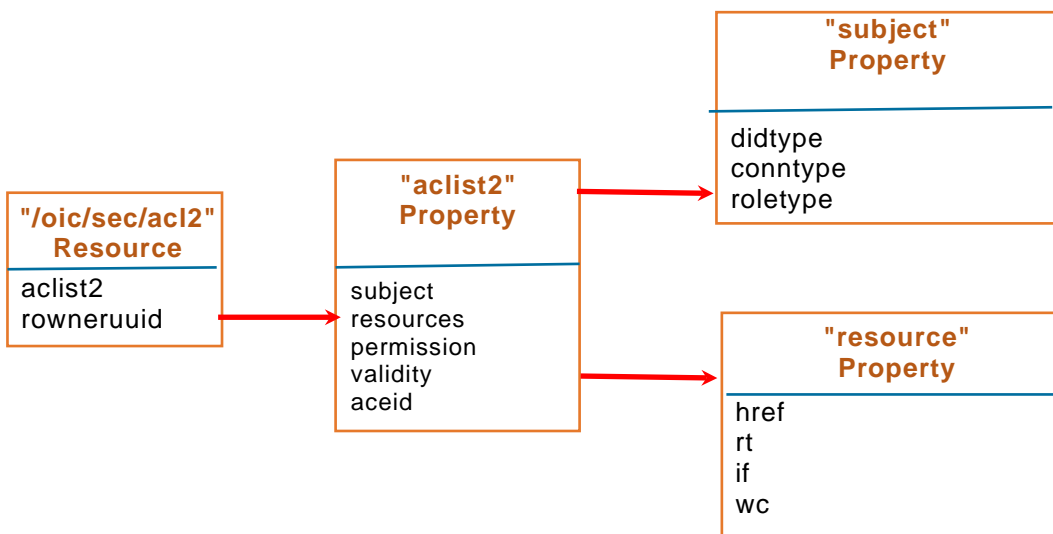


Figure 25 – OCF Security Resources

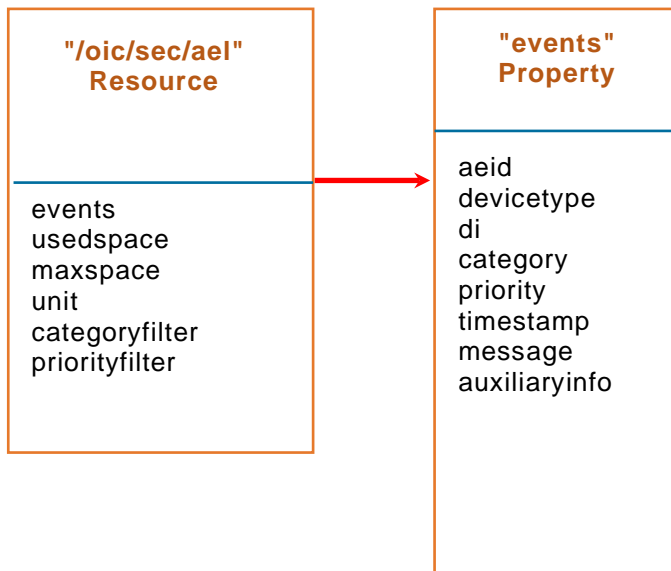


**Figure 26 – "/oic/sec/cred" Resource and Properties**



**Figure 27 – "/oic/sec/acl2" Resource and Properties**





**Figure 28 – "/oic/sec/ael" Resource and Properties**

## 13.2 Device Owner Transfer Resource

### 13.2.1 Device Owner Transfer Resource General

The "/oic/sec/doxm" Resource contains the set of supported Device OTMs.

Resource discovery processing respects the CRUDN constraints supplied as part of the security Resource definitions contained in this document.

"/oic/sec/doxm" Resource is defined in Table 15.

**Table 15 – Definition of the "/oic/sec/doxm" Resource**

| Fixed URI     | Resource Type Title | Resource Type ID ("rt" value) | OCF Interfaces                 | Description                                   | Related Functional Interaction |
|---------------|---------------------|-------------------------------|--------------------------------|---|--------------------------------|
| /oic/sec/doxm | Device OTMs         | oic.r.doxm                    | oic.if.baselin<br>e, oic.if.rw | Resource for supporting Device owner transfer | Configuration                  |

Table 16 defines the Properties of the "/oic/sec/doxm" Resource.

**Table 16 – Properties of the "/oic/sec/doxm" Resource**

| Property Title | Property Name | Value Type           | Value Rule | Mandat<br>ory | Device State        | Access Mode | Description  |
|----------------|---------------|----------------------|------------|---------------|---------------------|-------------|--|
| OTM            | oxms          | oic.sec.doxm<br>type | array      | Yes           |                     | R           | Value identifying the owner-transfer-method and the organization that defined the method.  |
| OTM Selection  | oxmsel        | oic.sec.doxm<br>type | UINT16     | Yes           | RESET               | R           | Server shall set to (4) "oic.sec.oxm.self"   |
|                |               |                      |            |               | RFOTM (no open DOC) | RW          | DOTS shall set to its selected DOTS and both parties execute the DOTS. After secure owner transfer session is established DOTS shall update the oxmsel again making it permanent. If the |

|                            |             |                  |                 |     |                     |    |  |
|----------------------------|-------------|------------------|-----------------|-----|---------------------|----|--|
|                            |             |                  |                 |     |                     |    | DOTS fails the Server shall transition device state to RESET.  |
|                            |             |                  |                 |     | RFOTM (open DOC)    | R  | n/a  |
|                            |             |                  |                 |     | RFPRO               | R  | n/a  |
|                            |             |                  |                 |     | RFNOP               | R  | n/a  |
|                            |             |                  |                 |     | SRESET              | R  | n/a  |
| Supported Credential Types | sct         | oic.sec.credtype | bitmask         | Yes |                     | R  | Identifies the types of credentials the Device supports. The Server sets this value at framework initialization after determining security capabilities.<br><br>The Device always supports symmetric pair-wise key and asymmetric signing key with certificate (bit positions 0x1 and 0x8 respectively). Other credential types are optional as per clause 9.3 |
| Device Ownership Status    | owned       | Boolean          | T F             | Yes | RESET               | R  | Server shall set to FALSE.   |
|                            |             |                  |                 |     | RFOTM (no open DOC) | R  | FALSE  |
|                            |             |                  |                 |     | RFOTM (open DOC)    | RW | DOTS (Device communicating over DOC) shall set to TRUE after secure owner transfer session is established.   |
|                            |             |                  |                 |     | RFPRO               | R  | TRUE   |
|                            |             |                  |                 |     | RFNOP               | R  | TRUE   |
|                            |             |                  |                 |     | SRESET              | R  | TRUE   |
| Device UUID                | deviceuuid  | String           | oic.sec.didtype | Yes | RESET               | R  | No stipulation.  |
|                            |             |                  |                 |     | RFOTM (no open DOC) | R  | n/a  |
|                            |             |                  |                 |     | RFOTM (open DOC)    | RW | DOTS (Device communicating over DOC) updates to a value it has selected after secure owner transfer session is established.  |
|                            |             |                  |                 |     | RFPRO               | R  | n/a  |
|                            |             |                  |                 |     | RFNOP               | R  | n/a  |
|                            |             |                  |                 |     | SRESET              | R  | n/a  |
| Device Owner Id            | devowneruid | String           | uid             | Yes | RESET               | R  | Server shall set to the nil uid value (e.g. "00000000-0000-0000-0000-000000000000" )   |
|                            |             |                  |                 |     | RFOTM (no open DOC) | R  | n/a  |
|                            |             |                  |                 |     | RFOTM (open DOC)    | RW | DOTS (Device communicating over DOC) shall set value after secure owner transfer session is established.   |
|                            |             |                  |                 |     | RFPRO               | R  | n/a  |
|                            |             |                  |                 |     | RFNOP               | R  | n/a  |
|                            |             |                  |                 |     | SRESET              | R  | n/a  |

|                   |            |        |      |     |                     |    |   |
|-------------------|------------|--------|------|-----|---------------------|----|---|
| Resource Owner Id | rowneruuid | String | uuid | Yes | RESET               | R  | Server shall set to the nil uuid value (e.g. "00000000-0000-0000-0000-000000000000" )   |
|                   |            |        |      |     | RFOTM (no open DOC) | R  | n/a   |
|                   |            |        |      |     | RFOTM (open DOC)    | RW | The DOTS (Device communicating over DOC) shall configure the rowneruuid Property when a successful owner transfer session is established.   |
|                   |            |        |      |     | RFPRO               | R  | n/a   |
|                   |            |        |      |     | RFNOP               | R  | n/a   |
|                   |            |        |      |     | SRESET              | RW | The DOTS (referenced via devowneruuid Property) should verify and if needed, update the Resource owner Property when a mutually authenticated secure session is established. If the rowneruuid does not refer to a valid DOTS device identifier the Server shall transition to RESET. |

2788 Table 17 defines the Properties of the "oic.sec.didtype".

2789 **Table 17 – Properties of the "oic.sec.didtype" type**

| Property Title | Property Name | Value Type | Value Rule | Mandatory | Device State | Access Mode | Description  |
|----------------|---------------|------------|------------|-----------|--------------|-------------|--------------|
| Device UUID    | uuid          | String     | uuid       | Yes       | RW           | -           | A uuid value |

2790 The "oxms" Property contains a list of OTM where the entries appear in the order of preference.  
2791 This Property contains the higher priority methods appearing before the lower priority methods.  
2792 The DOTS queries this list at the time of onboarding and selects the most appropriate method.

2793 OTMs consist of two parts, a URI identifying the vendor or organization and the specific method.

```

2794 <DoxmType> ::= <NSS>
2795 <NSS> ::= <Identifier> | { {<NID> "."} <NameSpaceQualifier> "." } <Method>
2796 <NID> ::= <Vendor-or-Organization>
2797 <Identifier> ::= INTEGER
2798 <NameSpaceQualifier> ::= String
2799 <Method> ::= String
2800 <Vendor-Organization> ::= String

```

2801 When an OTM successfully completes, the "owned" Property is set to "1" (TRUE). Consequently,  
2802 subsequent attempts to take ownership of the Device will fail.

2803 There are four device identifiers:

- 2804 1) "deviceuuid" Property of "/oic/sec/doxm" Resource - random DOTS-provisioned value unique  
2805 for a given security domain, used as a device identity for access control, mapped internally to  
2806 a device-owned credential.
- 2807 2) "di" Property of "/oic/d" Resource - mirroring the value of "deviceuuid" Property of  
2808 "/oic/sec/doxm" Resource.
- 2809 3) "piid" Property of "/oic/d" Resource - defined in ISO/IEC 30118-1.
- 2810 4) "pi" Property of "/oic/p" Resource - defined in ISO/IEC 30118-1.

2811 The "/oic/sec/doxm" Resource supports CoAP multicast requests in certain cases. For details see  
2812 clause 7.3.1

### 13.2.2 OCF defined OTMs

Table 18 defines the Properties of the "oic.sec.doxmtype".

**Table 18 – Properties of the "oic.sec.doxmtype" type**

| Value Type Name   | Value Type URN (optional) | Enumeration Value (mandatory) | Description   |
|---|---------------------------|-------------------------------|---|
| OCFJustWorks  | oic.sec.doxm.jw           | 0                             | The just-works method relies on anonymous Diffie-Hellman key agreement protocol to allow a DOTS to assert ownership of the new Device. The first DOTS to make the assertion is accepted as the Device owner. The just-works method results in a shared secret that is used to authenticate the Device to the DOTS and likewise authenticates the DOTS to the Device. The Device permits the DOTS to take ownership of the Device, after which a second attempt to take ownership by a different DOTS will fail <sup>a</sup> . |
| OCFSharedPin  | oic.sec.doxm.rdp          | 1                             | The new Device randomly generates a PIN that is communicated via an Out Of Band Communication Channel to a DOTS. An in-band Diffie-Hellman key agreement protocol establishes that both endpoints possess the PIN. Possession of the PIN by the DOTS signals the new Device that device ownership can be asserted.  |
| OCFMfgCert  | oic.sec.doxm.mfgcert      | 2                             | The new Device is presumed to have been manufactured with an embedded asymmetric private key that is used to sign a Diffie-Hellman exchange at Device onboarding. The manufacturer certificate should contain Platform hardening information and other security assurances assertions.  |
| OCF Reserved  | <Reserved>                | 3                             | Reserved  |
| OCFSelf   | oic.sec.oxm.self          | 4                             | The manufacturer shall set the "/doxm.oxmsel" value to (4). The Server shall reset this value to (4) upon entering RESET.   |
| OCF Reserved  | <Reserved>                | 5~0xFEFF                      | Reserved for OCF use  |
| Vendor-defined Value Type Name  | <Reserved>                | 0xFF00~0xFFFF                 | Reserved for vendor-specific OTM use  |
| <sup>a</sup> The just-works method is subject to a man-in-the-middle attacker. Precautions should be taken to provide physical security when this method is used. |                           |                               |   |

## 13.3 Credential Resource

### 13.3.1 Credential Resource General

The "/oic/sec/cred" Resource maintains credentials used to authenticate the Server to Clients and support services as well as credentials used to verify Clients and support services.

Multiple credential types are anticipated by the OCF framework, including pair-wise pre-shared keys, asymmetric keys, certificates and others. The credential Resource uses a Subject UUID to distinguish the Clients and support services it recognizes by verifying an authentication challenge.

In order to provide an interface which allows management of the "creds" Array Property, the RETRIEVE, UPDATE and DELETE operations on the "/oic/sec/cred" Resource shall behave as follows:

- 1) A RETRIEVE shall return the full Resource representation, except that any write-only Properties shall be omitted (e.g. private key data).
- 2) An UPDATE shall replace or add to the Properties included in the representation sent with the UPDATE request, as follows:

- 2830 a) If an UPDATE representation includes the "creds" array Property, then:
- 2831 i) Supplied "creds" with a "credid" that matches an existing "credid" shall replace
- 2832 completely the corresponding "cred" in the existing "creds" array.
- 2833 ii) Supplied "creds" without a "credid" shall be appended to the existing "creds" array, and
- 2834 a unique (to the "cred" Resource) "credid" shall be created and assigned to the new
- 2835 "cred" by the Server. The "credid" of a deleted "cred" should not be reused, to improve
- 2836 the determinism of the interface and reduce opportunity for race conditions.
- 2837 iii) Supplied "creds" with a "credid" that does not match an existing "credid" shall be
- 2838 appended to the existing "creds" array, using the supplied "credid".
- 2839 iv) The rows in Table 20 corresponding to the "creds" array Property dictate the Device
- 2840 States in which an UPDATE of the "creds" array Property is always rejected. If OCF
- 2841 Device is in a Device State where the Access Mode in this row contains "R", then the
- 2842 OCF Device shall reject all UPDATES of the "creds" array Property.
- 2843 3) A DELETE without query parameters shall set the "creds" array to the empty array, but shall
- 2844 not remove the "/oic/sec/cred" Resource.
- 2845 4) A DELETE with one or more "credid" query parameters shall remove the "cred"(s) with the
- 2846 corresponding "credid"(s) from the "creds" array.
- 2847 5) The rows in Table 20 corresponding to the "creds" array Property dictate the Device States in
- 2848 which a DELETE is always rejected. If OCF Device is in a Device State where the Access Mode
- 2849 in this row contains "R", then the OCF Device shall reject all DELETES.
- 2850 NOTE The "/oic/sec/cred" Resource's use of the DELETE operation is not in accordance with the OCF Interfaces defined
- 2851 in ISO/IEC 30118-1.
- 2852 "/oic/sec/cred" Resource is defined in Table 19.

2853 **Table 19 – Definition of the "/oic /sec/cred" Resource**

| Fixed URI     | Resource Type Title | Resource Type ID ("rt" value) | OCF Interfaces                | Description   | Related Functional Interaction |
|---------------|---------------------|-------------------------------|-------------------------------|---|--------------------------------|
| /oic/sec/cred | Credentials         | oic.r.cred                    | oic.if.baseline,<br>oic.if.rw | Resource containing credentials for Device authentication, verification and data protection | Security                       |

2854 Table 20 defines the Properties of the "/oic/sec/cred" Resource.

**Table 20 – Properties of the "/oic/sec/cred" Resource**

| Property Title    | Property Name | Value Type   | Value Rule | Mandatory | Device State | Access Mode | Description   |
|-------------------|---------------|--------------|------------|-----------|--------------|-------------|---|
| Credentials       | creds         | oic.sec.cred | array      | Yes       | RESET        | R           | Server shall set to manufacturer defaults.  |
|                   |               |              |            |           | RFOTM        | RW          | Set by DOTS after successful OTM  |
|                   |               |              |            |           | RFPRO        | RW          | Set by the CMS (referenced via the rowneruuid Property of "/oic/sec/cred" Resource) after successful authentication. Access to NCRs is prohibited.  |
|                   |               |              |            |           | RFNOP        | R           | Access to NCRs is permitted after a matching ACE is found.  |
|                   |               |              |            |           | SRESET       | RW          | The DOTS (referenced via devowneruuid Property of "/oic/sec/doxm" Resource or the rowneruuid Property of "/oic/sec/doxm" Resource) should evaluate the integrity of and may update creds entries when a secure session is established and the Server and DOTS are authenticated.  |
| Resource Owner ID | rowneruuid    | String       | uuid       | Yes       | RESET        | R           | Server shall set to the nil uuid value (e.g. "00000000-0000-0000-0000-000000000000" )   |
|                   |               |              |            |           | RFOTM        | RW          | The DOTS shall configure the rowneruuid Property of "/oic/sec/cred" Resource when a successful owner transfer session is established.   |
|                   |               |              |            |           | RFPRO        | R           | n/a   |
|                   |               |              |            |           | RFNOP        | R           | n/a   |
|                   |               |              |            |           | SRESET       | RW          | The DOTS (referenced via devowneruuid Property of "/oic/sec/doxm" Resource or the rowneruuid Property of "/oic/sec/doxm" Resource) should verify and if needed, update the Resource owner Property when a mutually authenticated secure session is established. If the "rowneruuid" Property does not refer to a valid DOTS the Server shall transition to RESET. |

2856 All secure Device accesses shall have a "/oic/sec/cred" Resource that protects the end-to-end  
 2857 interaction.

2858 The "/oic/sec/cred" Resource shall be updateable by the service named in its rowneruuid Property.

2859 ACLs naming "/oic/sec/cred" Resource should further restrict access beyond CRUDN access  
 2860 modes.

2861 Table 21 defines the Properties of "oic.sec.creds".

**Table 21 – Properties of the "oic.sec.creds" Property**

| Property Title   | Property Name | Value Type           | Value Rule | Mandatory | Access Mode | Device State | Description  |
|------------------|---------------|----------------------|------------|-----------|-------------|--------------|--|
| Credential ID    | credid        | UINT16               | 0 – 64K-1  | Yes       | RW          |              | Short credential ID for local references from other Resource   |
| Subject UUID     | subjectuuid   | String               | uuid       | Yes       | RW          |              | A uuid that identifies the subject to which this credential applies or "" if any identity is acceptable  |
| Role ID          | roleid        | oic.sec.roletype     | -          | No        | RW          |              | Identifies the role(s) the subject is authorized to assert.  |
| Credential Type  | credtype      | oic.sec.credtype     | bitmask    | Yes       | RW          |              | Represents this credential's type.<br>0 – Used for testing<br>1 – Symmetric pair-wise key<br>2 – Symmetric group key<br>4 – Asymmetric signing key<br>8 – Asymmetric signing key with certificate<br>16 – PIN or password<br>32 – Asymmetric encryption key<br>64 – Directly Provisioned OSCORE Security Context<br>128 – Simple Secure Multicast Client Context<br>256 – Simple Secure Multicast Server Context |
| Credential Usage | credusage     | oic.sec.credusage    | String     | No        | RW          |              | Used to resolve undecidability of the credential. Provides indication for how/where the cred is used<br>"oic.sec.cred.trustca": certificate trust anchor<br>"oic.sec.cred.cert": identity certificate<br>"oic.sec.cred.rolecert": role certificate<br>"oic.sec.cred.mfgtrustca": manufacturer certificate trust anchor<br>"oic.sec.cred.mfgcert": manufacturer certificate                                       |
| Public Data      | publicdata    | oic.sec.pubdatatype  | -          | No        | RW          |              | Credential Type dependent. Public credential information<br>1:2: ticket, public SKDC values<br>4, 32: Public key value<br>8: A chain of one or more certificate  |
| Private Data     | privatedata   | oic.sec.privdatatype | -          | No        | -           | RESET        | Server shall set to manufacturer default   |
|                  |               |                      |            |           | RW          | RFOTM        | Set by DOTS after successful OTM   |
|                  |               |                      |            |           | W           | RFPRO        | Set by authenticated DOTS or CMS   |
|                  |               |                      |            |           | -           | RFNOP        | Not writable during normal operation.  |
|                  |               |                      |            |           | W           | SRESET       | DOTS may modify to enable transition to RFPRO.   |
| Optional Data    | optionaldata  | oic.sec.optdatatype  | -          | No        | RW          |              | Credential Type dependent. Credential revocation status information<br>1, 2, 4, 32, 64: revocation status information<br>8: Revocation information   |



|                           |        |                    |       |    |    |  |   |
|---------------------------|--------|--------------------|-------|----|----|--|---|
| Period                    | period | String             | -     | No | RW |  | Period as defined by IETF RFC 5545. The credential should not be used if the current time is outside the Period window.                         |
| Credential Refresh Method | crms   | oic.sec.crmtype    | array | No | RW |  | Credentials with a Period Property are refreshed using the credential refresh method (crm) according to the type definitions for "oic.sec.crm". |
| OSCORE Configuration      | oscore | oic.sec.oscoretype |       | No | RW |  | Contains parameters for use with credentials intended for use with OSCORE. See type definition for "oic.sec.oscoretype"                         |

2863 Table 22 defines the Properties of "oic.sec.credusagetype".

2864 **Table 22: Properties of the "oic.sec.credusagetype" Property**

| Value Type Name       | Value Type URN<br>(mandatory) |
|-----------------------|-------------------------------|
| Trust Anchor          | oic.sec.cred.trustca          |
| Certificate           | oic.sec.cred.cert             |
| Role Certificate      | oic.sec.cred.rolecert         |
| Manufacturer Trust CA | oic.sec.cred.mfgtrustca       |
| Manufacturer CA       | oic.sec.cred.mfgcert          |

2865 Table 23 defines the Properties of "oic.sec.pubdatatype".

2866 **Table 23 – Properties of the "oic.sec.pubdatatype" Property**

| Property Title  | Property Name | Value Type | Value Rule | Access Mode | Mandatory | Description  |
|-----------------|---------------|------------|------------|-------------|-----------|--|
| Encoding format | encoding      | String     | N/A        | RW          | No        | A string specifying the encoding format of the data contained in the pubdata<br>"oic.sec.encoding.pem" – Encoding for PEM-encoded certificate or chain |
| Data            | data          | String     | N/A        | RW          | No        | The encoded value  |

2867 Table 24 defines the Properties of "oic.sec.privdatatype".

2868 **Table 24 – Properties of the "oic.sec.privdatatype" Property**

| Property Title  | Property Name | Value Type | Value Rule | Access Mode | Mandatory | Description  |
|-----------------|---------------|------------|------------|-------------|-----------|--|
| Encoding format | encoding      | String     | N/A        | RW          | Yes       | A string specifying the encoding format of the data contained in the privdata<br>"oic.sec.encoding.pem" – Encoding for PEM-encoded private key<br>"oic.sec.encoding.base64" – Encoding of Base64 encoded PSK<br>"oic.sec.encoding.handle" – Data is contained in a storage sub-system referenced using a handle<br>"oic.sec.encoding.raw" – Raw hex encoded data |
| Data            | data          | String     | N/A        | W           | No        | The encoded value<br>This value shall not be RETRIEVE-able.  |
| Handle          | handle        | UINT16     | N/A        | RW          | No        | Handle to a key storage Resource   |

2869 Table 25 defines the Properties of "oic.sec.optdatatype".

2870

**Table 25 – Properties of the "oic.sec.optdatatype" Property**

| Property Title    | Property Name | Value Type | Value Rule | Access Mode | Mandatory | Description  |
|-------------------|---------------|------------|------------|-------------|-----------|--|
| Revocation status | revstat       | Boolean    | T   F      | RW          | Yes       | Revocation status flag<br>True – revoked<br>False – not revoked  |
| Encoding format   | encoding      | String     | N/A        | RW          | No        | A string specifying the encoding format of the data contained in the optdata<br>"oic.sec.encoding.pem" – Encoding for PEM-encoded certificate or chain |
| Data              | data          | String     | N/A        | RW          | No        | The encoded structure  |

2871 Table 26 defines the Properties of "oic.sec.roletype".

2872

**Table 26 – Definition of the "oic.sec.roletype" type.**

| Property Title | Property Name | Value Type | Value Rule | Access Mode | Mandatory | Description  |
|----------------|---------------|------------|------------|-------------|-----------|--|
| Authority      | authority     | String     | N/A        | R           | No        | A name for the authority that defined the role. If not present, the credential issuer defined the role. If present, must be expressible as an ASN.1 PrintableString. |
| Role           | role          | String     | N/A -      | R           | Yes       | An identifier for the role. Must be expressible as an ASN.1 PrintableString.   |

2873 Table 27 defines the Properties of "oic.sec.oscoretype".

2874

**Table 27 – Definition of the "oic.sec.oscoretype" type.**

| Property Title                      | Property Name | Value Type | Value Rule           | Access Mode | Mandatory | Description   |
|-------------------------------------|---------------|------------|----------------------|-------------|-----------|---|
| OSCORE Sender ID                    | senderid      | String     | Hexadecimal encoding | RW          | No        | OSCORE Sender ID for this OSCORE Security Context.  |
| OSCORE Recipient ID                 | recipientid   | String     |                      | RW          | No        | OSCORE Recipient ID for this OSCORE Security Context.   |
| OSCORE Sender Sequence Number 1     | ssn           | Integer    |                      | R           | No        | OSCORE Sender Sequence Number being stored in non volatile memory to handle the loss of mutable security context parameters. See clause 16.2.4. |
| OSCORE Security Context Description | desc          | String     |                      | RW          | No        | Description of the usage of this OSCOE Security Context.  |

2875 **13.3.2 Properties of the Credential Resource**2876 **13.3.2.1 Credential ID**

2877 Credential ID ("credid") is a local reference to an entry in a "creds" Property array of the  
 2878 "/oic/sec/cred" Resource. The SRM generates it. The "credid" Property shall be used to  
 2879 disambiguate array elements of the "creds" Property.

#### 2880 **13.3.2.2 Subject UUID**

2881 The "subjectuuid" Property identifies the Device to which an entry in a "creds" Property array of the  
2882 "/oic/sec/cred" Resource shall be used to establish a secure session, verify an authentication  
2883 challenge-response or to authenticate an authentication challenge.

2884 A "subjectuuid" Property that matches the Server's own "deviceuuid" Property, distinguishes the  
2885 array entries in the "creds" Property that pertain to this Device.

2886 The "subjectuuid" Property shall be used to identify a group to which a group key is used to protect  
2887 shared data.

2888 When certificate chain is used during secure connection establishment, the "subjectuuid" Property  
2889 shall also be used to verify the identity of the responder. The presented certificate chain shall be  
2890 accepted, if there is a matching Credential entry on the Device that satisfies all of the following:

- 2891 – Public Data of the entry contains trust anchor (root) of the presented chain.
- 2892 – Subject UUID of the entry matches UUID in the Common Name field of the End-Entity certificate  
2893 in the presented chain. If Subject UUID of the entry is set as a wildcard "\*", this condition is  
2894 automatically satisfied.
- 2895 – Credential Usage of the entry is "oic.sec.cred.trustca".

#### 2896 **13.3.2.3 Role ID**

2897 The "roleid" Property identifies a role that has been granted to the credential.

#### 2898 **13.3.2.4 Credential Type**

2899 The "credtype" Property is used to interpret several of the other Property values whose contents  
2900 can differ depending on credential type. These Properties include "publicdata", "privatedata" and  
2901 "optionaldata". The "credtype" Property value of "0" ("no security mode") is reserved for testing and  
2902 debugging circumstances. Production deployments shall not allow provisioning of credentials of  
2903 type "0". The SRM should introduce checking code that prevents its use in production deployments.

#### 2904 **13.3.2.5 Public Data**

2905 The "publicdata" Property contains information that provides additional context surrounding the  
2906 issuance of the credential. For example, it might contain information included in a certificate or  
2907 response data from a CMS. It might contain wrapped data.

#### 2908 **13.3.2.6 Private Data**

2909 The "privatedata" Property contains secret information that is used to authenticate a Device, protect  
2910 data or verify an authentication challenge-response.

2911 The "privatedata" Property shall not be disclosed outside of the SRM's trusted computing perimeter.  
2912 A secure element (SE) or trusted execution environment (TEE) should be used to implement the  
2913 SRM's trusted computing perimeter. The privatedata contents may be referenced using a handle;  
2914 for example, if used with a secure storage sub-system.

#### 2915 **13.3.2.7 Optional Data**

2916 The "optionaldata" Property contains information that is optionally supplied, but facilitates key  
2917 management, scalability or performance optimization.

#### 2918 **13.3.2.8 Period**

2919 The "period" Property identifies the validity period for the credential. If no validity period is specified,  
2920 the credential lifetime is undetermined. Constrained devices that do not implement a date-time  
2921 capability shall obtain current date-time information from its CMS.

### 13.3.2.9 Credential Refresh Method Type Definition [Deprecated]

This clause is intentionally left blank.

### 13.3.2.10 Credential Usage

Credential Usage indicates to the Device the circumstances in which a credential should be used. Five values are defined:

- "oic.sec.cred.trustca": This certificate is a trust anchor for the purposes of certificate chain validation, as defined in 10.4. OCF Server SHALL remove any "/oic/sec/cred" entries with an "oic.sec.cred.trustca" credusage upon transitioning to RFOTM. OCF Servers SHALL use "/oic/sec/cred" entries that have an "oic.sec.cred.trustca" Value of "credusage" Property only as trust anchors for post-onboarding (D)TLS session establishment in RFNOP; these entries are not to be used for onboarding (D)TLS sessions.
- "oic.sec.cred.cert": This "credusage" is used for certificates for which the Device possesses the private key and uses it for identity authentication in a secure session, as defined in clause 10.4.
- "oic.sec.cred.rolecert": This "credusage" is used for certificates for which the Device possesses the private key and uses to assert one or more roles, as defined in clause 10.4.2.
- "oic.sec.cred.mfgtrustca": This certificate is a trust anchor for the purposes of the Manufacturer Certificate Based OTM as defined in clause 7.3.6. OCF Servers SHALL use "/oic/sec/cred" entries that have an "oic.sec.cred.mfgtrustca" Value of "credusage" Property only as trust anchors for onboarding (D)TLS session establishment; these entries are not to be used for post-onboarding (D)TLS sessions.
- "oic.sec.cred.mfgcert": This certificate is used for certificates for which the Device possesses the private key and uses it for authentication in the Manufacturer Certificate Based OTM as defined in clause 7.3.6.

### 13.3.2.11 Resource Owner

The Resource Owner Property allows credential provisioning to occur soon after Device onboarding before access to support services has been established. It identifies the entity authorized to manage the "/oic/sec/cred" Resource in response to Device recovery situations.

## 13.3.3 Key Formatting

### 13.3.3.1 Symmetric Key Formatting

Symmetric keys shall have the format described in Table 28 and Table 29.

**Table 28 – 128-bit symmetric key**

| Name   | Value  | Type        | Description  |
|--------|--------|-------------|--|
| Length | 16     | OCTET       | Specifies the number of 8-bit octets following Length                            |
| Key    | opaque | OCTET Array | 16-byte array of octets. When used as input to a PSK function Length is omitted. |

**Table 29 – 256-bit symmetric key**

| Name   | Value  | Type        | Description  |
|--------|--------|-------------|--|
| Length | 32     | OCTET       | Specifies the number of 8-bit octets following Length                            |
| Key    | opaque | OCTET Array | 32-byte array of octets. When used as input to a PSK function Length is omitted. |

### 13.3.3.2 Asymmetric Keys

Asymmetric key formatting is not available in this revision of the document.

### 13.3.3.3 Asymmetric Keys with Certificate

Key formatting is defined by certificate definition.

### 13.3.3.4 Passwords

Password formatting is not available in this revision of the document.

### 13.3.4 Credential Refresh Method Details [Deprecated]

This clause is intentionally left blank.

## 13.4 Certificate Revocation List

### 13.4.1 CRL Resource Definition [Deprecated]

This clause is intentionally left blank.

## 13.5 ACL Resources

### 13.5.1 ACL Resources General

All Resource hosted by a Server are required to match an ACL policy. ACL policies can be expressed using "/oic/sec/acl2". The subject (e.g. "deviceuuid" of the Client) requesting access to a Resource shall be authenticated prior to applying the ACL check. Resources that are available to multiple Clients can be matched using a wildcard subject. All Resources accessible via the unsecured communication endpoint shall be matched using a wildcard subject.

### 13.5.2 OCF Access Control List (ACL) BNF defines ACL structures.

ACL structure in Backus-Naur Form (BNF) notation is defined in Table 30:

**Table 30 – BNF Definition of OCF ACL**

|               |  |
|---------------|--|
| <ACL>         | <ACE> {<ACE>}  |
| <ACE>         | <SubjectId> <ResourceRef> <Permission> {<Validity>}              |
| <SubjectId>   | <DeviceId>   <Wildcard>   <RoleId>                               |
| <DeviceId>    | <UUID>   |
| <RoleId>      | <Character>   <RoleName><Character>                              |
| <RoleName>    | " "   <Authority><Character>                                     |
| <Authority>   | <UUID>   |
| <ResourceRef> | ' (' <OIC_LINK> {',' {OIC_LINK}> } ')'                           |
| <Permission>  | ('C'   '-' ) ('R'   '-' ) ('U'   '-' ) ('D'   '-' ) ('N'   '-' ) |
| <Validity>    | <Period> {<Recurrence>}  |
| <Wildcard>    | '*'  |
| <URI>         | IETF RFC 3986  |
| <UUID>        | IETF RFC 4122  |
| <Period>      | IETF RFC 5545 Period   |
| <Recurrence>  | IETF RFC 5545 Recurrence   |
| <OIC_LINK>    | ISO/IEC 30118-1 defined in JSON Schema                           |
| <Character>   | <Any UTF8 printable character, excluding NUL>                    |

The <DeviceId> token means the requestor must possess a credential that uses <UUID> as its identity in order to match the requestor to the <ACE> policy.

The <RoleId> token means the requestor must possess a role credential with <Character> as its role in order to match the requestor to the <ACE> policy.

2980 The <Wildcard> token "\*" means any requestor is matched to the <ACE> policy, with or without  
 2981 authentication.

2982 When a <SubjectId> is matched to an <ACE> policy the <ResourceRef> is used to match the <ACE>  
 2983 policy to Resources.

2984 The <OIC\_LINK> token contains values used to query existence of hosted Resources.

2985 The <Permission> token specifies the privilege granted by the <ACE> policy given the <SubjectId>  
 2986 and <ResourceRef> matching does not produce the empty set match.

2987 Permissions are defined in terms of CREATE ("C"), RETRIEVE ("R"), UPDATE ("U"), DELETE ("D"),  
 2988 NOTIFY ("N") and NIL ("-"). NIL is substituted for a permissions character that signifies the  
 2989 respective permission is not granted.

2990 The empty set match result defaults to a condition where no access rights are granted.

2991 If the <Validity> token exists, the <Permission> granted is constrained to the time <Period>.  
 2992 <Validity> may further be segmented into a <Recurrence> pattern where access may alternatively  
 2993 be granted and rescinded according to the pattern.

2994 **13.5.3 ACL Resource**

2995 An "acl2" is a list of type "ace2".

2996 In order to provide an interface which allows management of array elements of the "aclist2"  
 2997 Property associated with a "/oic/sec/acl2" Resource, the RETRIEVE, UPDATE and DELETE  
 2998 operations on the "/oic/sec/acl2" Resource SHALL behave as follows:

2999 1) A RETRIEVE shall return the full Resource representation.

3000 2) An UPDATE shall replace or add to the Properties included in the representation sent with the  
 3001 UPDATE request, as follows:

3002 a) If an UPDATE representation includes the "aclist2" array Property, then:

3003 i) Supplied ACEs with an "aceid" that matches an existing "aceid" shall replace completely  
 3004 the corresponding ACE in the existing "aclist2" array.

3005 ii) Supplied ACEs without an "aceid" shall be appended to the existing "aclist2" array, and  
 3006 a unique (to the "/oic/sec/acl2" Resource) "aceid" shall be created and assigned to the  
 3007 new ACE by the Server. The "aceid" of a deleted ACE should not be reused, to improve  
 3008 the determinism of the interface and reduce opportunity for race conditions.

3009 iii) Supplied ACEs with an "aceid" that does not match an existing "aceid" shall be  
 3010 appended to the existing "aclist2" array, using the supplied "aceid".

3011 iv) The rows in Table 33 corresponding to the "aclist2" array Property dictate the Device  
 3012 States in which an UPDATE of the "aclist2" array Property is always rejected. If OCF  
 3013 Device is in a Device State where the Access Mode in this row contains "R", then the  
 3014 OCF Device shall reject all UPDATES of the "aclist2" array Property.

3015 3) A DELETE without query parameters shall set the "aclist2" array to the empty array, but shall  
 3016 not remove the "oic/sec/ace2" Resource.

3017 4) A DELETE with one or more "aceid" query parameters shall remove the ACE(s) with the  
 3018 corresponding "aceid"(s) from the "aclist2" array.

3019 5) The rows in Table 33 corresponding to the "aclist2" array Property dictate the Device States in  
 3020 which a DELETE is always rejected. If OCF Device is in a Device State where the Access Mode  
 3021 in this row contains "R", then the OCF Device shall reject all DELETES.

3022 NOTE The "/oic/sec/acl2" Resource's use of the DELETE operation is not in accordance with the OCF Interfaces  
 3023 defined in ISO/IEC 30118-1.

3024 Evaluation of local ACL Resource completes when all ACL Resource have been queried and no  
 3025 entry can be found for the requested Resource for the requestor – e.g. "/oic/sec/acl2" does not  
 3026 match the subject and the requested Resource.

3027 Table 31 defines the values of "oic.sec.crudntype".

3028 **Table 31 – Value Definition of the "oic.sec.crudntype" Property**

| Value            | Access Policy  | Description                    | RemarksNotes  |
|------------------|----------------|--------------------------------|---|
| bx0000,0000 (0)  | No permissions | No permissions                 | N/A   |
| bx0000,0001 (1)  | C              | CREATE                         | N/A   |
| bx0000,0010 (2)  | R              | RETRIEVE, OBSERVE,<br>DISCOVER | The "R" permission bit covers both the Read permission and the Observe permission.  |
| bx0000,0100 (4)  | U              | WRITE, UPDATE                  | N/A   |
| bx0000,1000 (8)  | D              | DELETE                         | N/A   |
| bx0001,0000 (16) | N              | NOTIFY                         | The "N" permission bit is ignored in OCF 1.0, since "R" covers the Observe permission. It is documented for future versions |

3029 "/oic/sec/acl2" Resource is defined in Table 19.

3030 **Table 32 – Definition of the "oic/sec/acl2" Resource**

| Fixed URI     | Resource Type Title | Resource Type ID ("rt" value) | OCF Interfaces                 | Description                  | Related Functional Interaction |
|---------------|---------------------|-------------------------------|--------------------------------|------------------------------|--------------------------------|
| /oic/sec/acl2 | ACL2                | oic.r.acl2                    | oic.if.baseli<br>ne, oic.if.rw | Resource for managing access | Security                       |

3031 Table 33 defines the Properties of "oic.sec.acl2".

**Table 33 – Properties of the "/oic/sec/acl2" Resource**

| Property Name | Value Type            | Mandatory | Device State | Access Mode | Description   |
|---------------|-----------------------|-----------|--------------|-------------|---|
| aclist2       | array of oic.sec.ace2 | Yes       | N/A          |             | The aclist2 Property is an array of ACE records of type "oic.sec.ace2". The Server uses this list to apply access control to its local Resources.   |
| N/A           | N/A                   | N/A       | RESET        | R           | Server shall set to manufacturer defaults.  |
|               |                       |           | RFOTM        | RW          | Set by DOTS after successful OTM  |
|               |                       |           | RFPRO        | RW          | The AMS (referenced via rowneruuid property) shall update the aclist entries after mutually authenticated secure session is established. Access to NCRs is prohibited.  |
|               |                       |           | RFNOP        | R           | Access to NCRs is permitted after a matching ACE2 is found.   |
|               |                       |           | SRESET       | RW          | The DOTS (referenced via devowneruuid Property of "/oic/sec/doxm Resource") should evaluate the integrity of and may update aclist entries when a secure session is established and the Server and DOTS are authenticated.  |
| rowneruuid    | uuid                  | Yes       | N/A          |             | The Resource owner Property (rowneruuid) is used by the Server to reference a service provider trusted by the Server. Server shall verify the service provider is authorized to perform the requested action  |
|               |                       |           | RESET        | R           | Server shall set to the nil uuid value (e.g. "00000000-0000-0000-0000-000000000000" )   |
|               |                       |           | RFOTM        | RW          | The DOTS should configure the rowneruuid Property of "/oic/sec/acl2" Resource when a successful owner transfer session is established.  |
|               |                       |           | RFPRO        | R           | n/a   |
|               |                       |           | RFNOP        | R           | n/a   |
|               |                       |           | SRESET       | RW          | The DOTS (referenced via devowneruuid Property or rowneruuid Property of "/oic/sec/doxm" Resource) should verify and if needed, update the Resource owner Property when a mutually authenticated secure session is established. If the rowneruuid Property does not refer to a valid DOTS the Server shall transition to RESET. |



3035

**Table 34 – "oic.sec.ace2" data type definition.**

| Property Name | Value Type  | Mandatory | Description   |
|---------------|---|-----------|---|
| subject       | oic.sec.roletype,<br>oic.sec.didtype,<br>oic.sec.conntype | Yes       | The Client is the subject of the ACE when the roles, Device UUID, or connection type matches.   |
| resources     | array of<br>oic.sec.ace2.resource-ref                     | Yes       | The application's Resources to which a security policy applies  |
| permission    | oic.sec.crudntype.bitmask                                 | Yes       | Bitmask encoding of CRUDN permission  |
| validity      | array of oic.sec.time-pattern                             | No        | An array of a tuple of period and recurrence. Each item in this array contains a string representing a period using the IETF RFC 5545 Period, and a string array representing a recurrence rule using the IETF RFC 5545 Recurrence. |
| aceid         | integer   | Yes       | An aceid is unique with respect to the array entries in the aclist2 Property.   |

3036 Table 35 defines the Properties of "oic.sec.ace2.resource-ref".

3037

**Table 35 – "oic.sec.ace2.resource-ref" data type definition.**

| Property Name | Value Type | Mandatory | Description   |
|---------------|------------|-----------|---|
| href          | uri        | No        | A URI referring to a Resource to which the containing ACE applies |
| wc            | string     | No        | Refer to Table 14.  |

3038 Table 36 defines the values of "oic.sec.ace2.resource-ref".

3039

**Table 36 – Value definition "oic.sec.conntype" Property**

| Property Name | Value Type | Value Rule                                | Description  |
|---------------|------------|---|--|
| conntype      | string     | enum<br>[ "auth-crypt",<br>"anon-clear" ] | This Property allows an ACE to be matched based on the connection or message protection type                                   |
|               |            | auth-crypt                                | ACE applies if the Client is authenticated and the data channel or message is encrypted and integrity protected                |
|               |            | anon-clear                                | ACE applies if the Client is not authenticated and the data channel or message is not encrypted but may be integrity protected |

3040 Local ACL Resources supply policy to a Resource access enforcement point within an OCF stack  
 3041 instance. The OCF framework gates Client access to Server Resources. It evaluates the subject's  
 3042 request using policies contained in ACL Resources.

3043 Resources named in the ACL policy can be fully qualified or partially qualified. Fully qualified  
 3044 Resource references include the device identifier in the href Property that identifies the remote  
 3045 Resource Server that hosts the Resource. Partially qualified references mean that the local  
 3046 Resource Server hosts the Resource. If a fully qualified Resource reference is given, the  
 3047 Intermediary enforcing access shall have a secure channel to the Resource Server and the  
 3048 Resource Server shall verify the Intermediary is authorized to act on its behalf as a Resource  
 3049 access enforcement point.

3050 Resource Servers should include references to Device and ACL Resources where access  
3051 enforcement is to be applied. However, access enforcement logic shall not depend on these  
3052 references for access control processing as access to Server Resources will have already been  
3053 granted.

3054 Local ACL Resources identify a Resource Owner service that is authorized to instantiate and modify  
3055 this Resource. This prevents non-terminating dependency on some other ACL Resource.  
3056 Nevertheless, it should be desirable to grant access rights to ACL Resources using an ACL  
3057 Resource.

3058 An ACE2 entry is considered "currently valid" if the validity period of the ACE2 entry includes the  
3059 time of the request. The validity period in the ACE2 may be a recurring time period (e.g., daily from  
3060 1:00-2:00). Matching the Resource(s) specified in a request to the "resource" Property of the ACE2  
3061 is defined in clause 12.2. For example, one way they can match is if the Resource URI in the  
3062 request exactly matches one of the Resource references in the ACE2 entries.

3063 A request will match an ACE2 if any of the following are true:

- 3064 1) The ACE2 "subject" Property is of type "oic.sec.didtype" has a UUID value that matches the  
3065 "deviceuuid" Property associated with the secure session;  
3066 AND the Resource of the request matches one of the "resources" Property of the ACE2  
3067 "oic.sec.ace2.resource-ref";  
3068 AND the ACE2 is currently valid.
- 3069 2) The ACE2 "subject" Property is of type "oic.sec.conntype" and has the wildcard value that  
3070 matches the currently established connection type;  
3071 AND the Resource of the request matches one of the "resources" Property of the ACE2  
3072 "oic.sec.ace2.resource-ref";  
3073 AND the ACE2 is currently valid.
- 3074 3) When Client authentication uses a certificate credential;  
3075 AND one of the "roleid" values contained in the role certificate matches the "roleid" Property of  
3076 the ACE2 "oic.sec.roletype";  
3077 AND the role certificate public key matches the public key of the certificate used to establish  
3078 the current secure session;  
3079 AND the Resource of the request matches one of the array elements of the "resources"  
3080 Property of the ACE2 "oic.sec.ace2.resource-ref";  
3081 AND the ACE2 is currently valid.
- 3082 4) When Client authentication uses a certificate credential;  
3083 AND the CoAP payload query string of the request specifies a role, which is member of the set  
3084 of roles contained in the role certificate;  
3085 AND the roleid values contained in the role certificate matches the "roleid" Property of the ACE2  
3086 "oic.sec.roletype";  
3087 AND the role certificate public key matches the public key of the certificate used to establish  
3088 the current secure session;  
3089 AND the Resource of the request matches one of the "resources" Property of the ACE2  
3090 "oic.sec.ace2.resource-ref";  
3091 AND the ACE2 is currently valid.
- 3092 5) When Client authentication uses a symmetric key credential;  
3093 AND one of the "roleid" values associated with the symmetric key credential used in the secure  
3094 session, matches the "roleid" Property of the ACE2 "oic.sec.roletype";

3095 AND the Resource of the request matches one of the array elements of the "resources"  
 3096 Property of the ACE2 "oic.sec.ace2.resource-ref";  
 3097 AND the ACE2 is currently valid.  
 3098 6) When Client authentication uses a symmetric key credential;  
 3099 AND the CoAP payload query string of the request specifies a role, which is contained in the  
 3100 "oic.r.cred.creds.roleid" Property of the current secure session;  
 3101 AND CoAP payload query string of the request specifies a role that matches the "roleid"  
 3102 Property of the ACE2 "oic.sec.roletype";  
 3103 AND the Resource of the request matches one of the array elements of the "resources"  
 3104 Property of the ACE2 "oic.sec.ace2.resource-ref";  
 3105 AND the ACE2 is currently valid.

3106 A request is granted if ANY of the 'matching' ACE2 entries contain the permission to allow the  
 3107 request. Otherwise, the request is denied.

3108 There is no way for an ACE2 entry to explicitly deny permission to a Resource. Therefore, if one  
 3109 Device with a given role should have slightly different permissions than another Device with the  
 3110 same role, they must be provisioned with different roles.

3111 The Server is required to verify that any hosted Resource has authorized access by the Client  
 3112 requesting access. The "/oic/sec/acl2" Resource is co-located on the Resource host so that the  
 3113 Resource request processing should be applied securely and efficiently. See Annex A for example.

### 3114 13.6 Access Manager ACL Resource [Deprecated]

3115 This clause is intentionally left blank.

### 3116 13.7 Signed ACL Resource [Deprecated]

3117 This clause is intentionally left blank.

### 3118 13.8 Provisioning Status Resource

3119 The "/oic/sec/pstat" Resource maintains the Device provisioning status. Device provisioning should  
 3120 be Client-directed or Server-directed. Client-directed provisioning relies on a Client device to  
 3121 determine what, how and when Server Resources should be instantiated and updated. Server-  
 3122 directed provisioning relies on the Server to seek provisioning when conditions dictate. Furthermore,  
 3123 the "/oic/sec/cred" Resource should be provisioned at ownership transfer with credentials  
 3124 necessary to open a secure connection with appropriate support service.

3125 "/oic/sec/pstat" Resource is defined in Table 37.

3126 **Table 37 – Definition of the "/oic/sec/pstat" Resource**

| Fixed URI      | Resource Type Title | Resource Type ID ("rt" value) | OCF Interfaces                | Description                                      | Related Functional Interaction |
|----------------|---------------------|-------------------------------|-------------------------------|--|--------------------------------|
| /oic/sec/pstat | Provisioning Status | oic.r.pstat                   | oic.if.baseline,<br>oic.if.rw | Resource for managing Device provisioning status | Configuration                  |

3127 Table 38 defines the Properties of "/oic/sec/pstat".

**Table 38 – Properties of the "/oic/sec/pstat" Resource**

| Property Title          | Property Name | Value Type       | Value Rule | Mandatory | Access Mode | Device State | Description   |
|-------------------------|---------------|------------------|------------|-----------|-------------|--------------|---|
| Device Onboarding State | dos           | oic.sec.dostype  | N/A        | Yes       | RW          |              | Device Onboarding State   |
| Is Device Operational   | isop          | Boolean          | T F        | Yes       | R           | RESET        | Server shall set to FALSE   |
|                         |               |                  |            |           | R           | RFOTM        | Server shall set to FALSE   |
|                         |               |                  |            |           | R           | RFPRO        | Server shall set to FALSE   |
|                         |               |                  |            |           | R           | RFNOP        | Server shall set to TRUE  |
|                         |               |                  |            |           | R           | SRESET       | Server shall set to FALSE   |
| Current Mode            | cm            | oic.sec.dpmttype | bitmask    | Yes       | R           |              | Current Mode  |
| Target Mode             | tm            | oic.sec.dpmttype | bitmask    | Yes       | RW          |              | Target Mode   |
| Operational Mode        | om            | oic.sec.pomtype  | bitmask    | Yes       | R           | RESET        | Server shall set to manufacturer default.   |
|                         |               |                  |            |           | RW          | RFOTM        | Set by DOTS after successful OTM  |
|                         |               |                  |            |           | RW          | RFPRO        | Set by CMS, AMS, DOTS after successful authentication   |
|                         |               |                  |            |           | RW          | RFNOP        | Set by CMS, AMS, DOTS after successful authentication   |
|                         |               |                  |            |           | RW          | SRESET       | Set by DOTS.  |
| Supported Mode          | sm            | oic.sec.pomtype  | bitmask    | Yes       | R           | All states   | Supported provisioning services operation modes   |
| Device UUID             | deviceuuid    | String           | uuid       | Yes       | RW          | All states   | [DEPRECATED] A uuid that identifies the Device to which the status applies  |
| Resource Owner ID       | rowneruuid    | String           | uuid       | Yes       | R           | RESET        | Server shall set to the nil uuid value (e.g. "00000000-0000-0000-0000-000000000000" )   |
|                         |               |                  |            |           | RW          | RFOTM        | The DOTS should configure the rowneruuid Property when a successful owner transfer session is established.  |
|                         |               |                  |            |           | R           | RFPRO        | n/a   |
|                         |               |                  |            |           | R           | RFNOP        | n/a   |
|                         |               |                  |            |           | RW          | SRESET       | The DOTS (referenced via devowneruuid Property of "/oic/sec/doxm" Resource) should verify and if needed, update the Resource owner Property when a mutually authenticated secure session is established. If the rowneruuid does not refer to a valid DOTS the Server shall transition to RESET. |

**Table 39 – Properties of the ".oic.sec.dostype" Property**

| Property Title          | Property Name | Value Type | Value Rule  | Mandatory | Access Mode | Device State | Description   |
|-------------------------|---------------|------------|---|-----------|-------------|--------------|---|
| Device Onboarding State | s             | UINT16     | enum<br>(0=RESET,<br>1=RFOTM,<br>2=RFPRO,<br>3=RFNOP,<br>4=SRESET | Y         | R           | RESET        | The Device is in a hard reset state.  |
|                         |               |            |   |           | RW          | RFOTM        | Set by DOTS after successful OTM to RFPRO.  |
|                         |               |            |   |           | RW          | RFPRO        | Set by CMS, AMS, DOTS after successful authentication   |
|                         |               |            |   |           | RW          | RFNOP        | Set by CMS, AMS, DOTS after successful authentication   |
|                         |               |            |   |           | RW          | SRESET       | Set by CMS, AMS, DOTS after successful authentication   |
| Pending state           | p             | Boolean    | T   F   | Y         | R           | All States   | FALSE (0) – "s" state changes are complete. Since Device is not able to respond when the value is TRUE, other values of this property are DEPRECATED. |

3132 In all Device states:

- 3133 – The Device permits an authenticated and authorised Client to change the Device state of a  
3134 Device by updating the "s" Property of the "dos" Property of the "/oic/sec/pstat" Resource to  
3135 the desired value. The allowed Device state transitions are defined in Figure 22.
- 3136 – Prior to updating the "s" Property of the "dos" Property of the "/oic/sec/pstat" Resource, the  
3137 Client configures the Device to meet entry conditions for the new Device state. The SVR  
3138 definitions define the entity (Client or Server) expected to perform the specific SVR  
3139 configuration change to meet the entry conditions. Once the Client has configured the aspects  
3140 for which the Client is responsible, it can update the "s" Property of the "dos" Property of the  
3141 "/oic/sec/pstat" Resource. The Server then makes any changes for which the Server is  
3142 responsible, including updating required SVR values, and set the "s" Property of the "dos"  
3143 Property of the "/oic/sec/pstat" Resource to the new value.

3144 When Device state is RESET:

- 3145 – All SVR content is removed and reset to manufacturer default values.
- 3146 – The default manufacturer Device state is RESET.
- 3147 – NCRs are reset to manufacturer default values.
- 3148 – NCRs shall not be accessible.
- 3149 – After successfully processing RESET the SRM transitions to RFOTM by setting the "s" Property  
3150 of the "dos" Property of the "/oic/sec/pstat" Resource to 1 (RFOTM).

3151 When Device state is RFOTM:

- 3152 – NCRs shall not be accessible.
- 3153 – Before OTM is successful, the the "s" Property of the "dos" Property of the "/oic/sec/pstat"  
3154 Resource is read-only by unauthenticated requestors
- 3155 – After the OTM is successful, the "s" Property of the "dos" Property of the "/oic/sec/pstat"  
3156 Resource is read-write by authorized requestors.
- 3157 – The negotiated Device OC is used to create an authenticated session over which the DOTS  
3158 directs the Device state to transition to RFPRO.

- 3159 – If an authenticated session cannot be established the ownership transfer session should be  
3160 disconnected and SRM sets back the Device state to RESET.
- 3161 – Ownership transfer session, especially Random PIN OTM, should not exceed 60 seconds. If  
3162 the SRM asserts the OTM failed, the ownership transfer session should be disconnected, and  
3163 the Device should transition to RESET ("/pstat.dos.s"=0 (RESET)).
- 3164 – The DOTS UPDATES the "devowneruuid" Property in the "/oic/sec/doxm" Resource to a non-  
3165 nil UUID value. The DOTS (or other authorized client) can update it multiple times while in  
3166 RFOTM. It is not updatable while in other device states except when the Device state returns  
3167 to RFOTM through RESET.
- 3168 – The DOTS can have additional provisioning tasks to perform while in RFOTM. When done, the  
3169 DOTS UPDATES the "owned" Property in the "/oic/sec/doxm" Resource to "true".
- 3170 – After successful OTM, the DOTS triggers the transition to RFPRO and the "s" Property of the  
3171 "dos" Property of the "/oic/sec/pstat" Resource is set to 2 (RFPRO).
- 3172 When Device state is RFPRO:
- 3173 – The "s" Property of the "dos" Property of the "/oic/sec/pstat" Resource is read-only by  
3174 unauthorized requestors and read-write by authorized requestors.
- 3175 – NCRs shall not be accessible, except for Easy Setup Resources, if supported.
- 3176 – An authorized Client may provision SVRs as needed for normal functioning in RFNOP.
- 3177 – An authorized Client may perform consistency checks on SVRs to determine which shall be re-  
3178 provisioned.
- 3179 – Failure to successfully provision SVRs may trigger a state change to RESET. For example, if  
3180 the Device has already transitioned from SRESET but consistency checks continue to fail.
- 3181 – The authorized Client sets the "s" Property of the "dos" Property of the "/oic/sec/pstat" Resource  
3182 to 3 (RFNOP).
- 3183 When Device state is RFNOP:
- 3184 – The "s" Property of the "dos" Property of the "/oic/sec/pstat" Resource is read-only by  
3185 unauthorized requestors and read-write by authorized requestors.
- 3186 – NCRs, SVRs and core Resources are accessible following normal access processing.
- 3187 – When additional provisioning is necessary, the Device may be transitioned to RFPRO by an  
3188 authorized Client. Only the Device owner should transition to SRESET or RESET.
- 3189 When Device state is SRESET:
- 3190 – NCRs shall not be accessible. The integrity of NCRs may be suspect but the SRM doesn't  
3191 attempt to access or reference them.
- 3192 – SVR integrity is not guaranteed, but access to some SVR Properties is necessary. These  
3193 include "devowneruuid" Property of the "/oic/sec/doxm" Resource,  
3194 "creds":[{...,"subjectuuid":<devowneruuid>,...}] Property of the "/oic/sec/cred" Resource and  
3195 "pstat.dos.s" "/oic/sec/pstat" Resource.
- 3196 – The certificates that identify and authorize the Device owner are sufficient to re-create  
3197 minimalist "/oic/sec/cred" and "/oic/sec/doxm" Resources enabling Device owner control of  
3198 SRESET. If the SRM can't establish these Resources, then it will transition to RESET.
- 3199 – An authorized Client performs SVR consistency checks. The authorized Client can provision  
3200 SVRs as needed to ensure they are available for continued provisioning in RFPRO or for normal  
3201 functioning in RFNOP.
- 3202 – The authorized Device owner can avoid entering RESET and RFOTM by UPDATING  
3203 "pstat.dos.s" with RFPRO or RFNOP values.

- 3204 – ACLs on SVR are presumed to be invalid. Access authorization is granted according to Device  
3205 owner privileges only.
- 3206 – The SRM asserts a Client-directed operational mode (e.g. "/pstat.om"=4).
- 3207 The provisioning mode type is a 16-bit mask enumerating the various Device provisioning modes.  
3208 "{ProvisioningMode}" should be used in this document to refer to an instance of a provisioning  
3209 mode without selecting any particular value.
- 3210 "oic.sec.dpmttype" is defined in Table 40.

3211 **Table 40 – Definition of the "oic.sec.dpmttype" Property**

| Type Name                | Type URN         | Description  |
|--------------------------|------------------|--|
| Device Provisioning Mode | oic.sec.dpmttype | Device provisioning mode is a 16-bit bitmask describing various provisioning modes |

3212 Table 41 and Table 42 define the values of "oic.sec.dpmttype".

3213 **Table 41 – Value Definition of the "oic.sec.dpmttype" Property (Low-Byte)**

| Value             | Device Mode                          | Description   |
|-------------------|--------------------------------------|---|
| bx0000,0001 (1)   | Deprecated                           |   |
| bx0000,0010 (2)   | Deprecated                           |   |
| bx0000,0100 (4)   | Deprecated                           |   |
| bx0000,1000 (8)   | Deprecated                           |   |
| bx0001,0000 (16)  | Deprecated                           |   |
| bx0010,0000 (32)  | Deprecated                           |   |
| bx0100,0000 (64)  | Initiate Software Version Validation | Software version validation requested/pending (1)<br>Software version validation complete (0)<br>Requires software download to verify integrity of software package |
| bx1000,0000 (128) | Initiate Secure Software Update      | Secure software update requested/pending (1)<br>Secure software update complete (0)   |

3214 **Table 42 – Value Definition of the "oic.sec.dpmttype" Property (High-Byte)**

| Value           | Device Mode                          | Description   |
|-----------------|--------------------------------------|---|
| bx0000,0001 (1) | Initiate Software Availability Check | Checks if new software is available on remote endpoint.<br>Does not require to download software.<br>Methods used are out of bound. |
| Bits 2-8        | <Reserved>                           | Reserved for later use  |

3215 The provisioning operation mode type is an 8-bit mask enumerating the various provisioning  
3216 operation modes.

3217 "oic.sec.pomtype" is defined in Table 43.

3218 **Table 43 – Definition of the "oic.sec.pomtype" Property**

| Type Name                         | Type URN        | Description   |
|-----------------------------------|-----------------|---|
| Device Provisioning OperationMode | oic.sec.pomtype | Device provisioning operation mode is a 8-bit bitmask describing various provisioning operation modes |

3219 Table 44 defines the values of "oic.sec.pomtype".

3220

**Table 44 – Value Definition of the "oic.sec.pomtype" Property**

| Value                             | Operation Mode   | Description   |
|-----------------------------------|--|---|
| bx0000,0001 (1)                   | Server-directed utilizing multiple provisioning services | Deprecated  |
| bx0000,0010 (2)                   | Server-directed utilizing a single provisioning service  | Deprecated  |
| bx0000,0100 (4)                   | Client-directed provisioning                             | Device supports provisioning service control of this Device's provisioning operations. This bit is always TRUE. |
| bx0000,1000(8) – bx1000,0000(128) | <Reserved>   | Reserved for later use  |
| bx1111,11xx                       | <Reserved>   | Reserved for later use  |

3221 **13.9 Certificate Signing Request Resource**

3222 The "/oic/sec/csr" Resource is used by a Device to provide its desired identity, public key to be  
 3223 certified, and a proof of possession of the corresponding private key in the form of a IETF RFC  
 3224 2986 PKCS#10 Certification Request. If the Device supports certificates (i.e. the "sct" Property of  
 3225 "/oic/sec/doxm" Resource has a 1 in the 0x8 bit position), the Device shall have a "/oic/sec/csr"  
 3226 Resource.

3227 "/oic/sec/csr" Resource is defined in Table 45.

3228

**Table 45 – Definition of the "/oic/sec/csr" Resource**

| Fixed URI    | Resource Type Title         | Resource Type ID ("rt" value) | OCF Interfaces             | Description  | Related Functional Interaction |
|--------------|-----------------------------|-------------------------------|----------------------------|--|--------------------------------|
| /oic/sec/csr | Certificate Signing Request | oic.r.csr                     | oic.if.baseline, oic.if.rw | The CSR Resource contains a Certificate Signing Request for the Device's public key. | Configuration                  |

3229 Table 46 defines the Properties of "/oic/sec/csr".

3230

**Table 46 – Properties of the "oic.r.csr" Resource**

| Property Title              | Property Name | Value Type | Access Mode | Mandatory | Description  |
|-----------------------------|---------------|------------|-------------|-----------|--|
| Certificate Signing Request | csr           | String     | R           | Yes       | Contains the signed CSR encoded according to the encoding Property   |
| Encoding                    | encoding      | String     | R           | Yes       | A string specifying the encoding format of the data contained in the csr Property<br>"oic.sec.encoding.pem" – Encoding for PEM-encoded certificate signing request |

3231 The Device chooses which public key to use, and may optionally generate a new key pair for this  
 3232 purpose.

3233 In the CSR, the Common Name component of the Subject Name shall contain a string of the format  
 3234 "uuid:X" where X is the Device's requested UUID in the format defined by IETF RFC 4122. The  
 3235 Common Name, and other components of the Subject Name, may contain other data. If the Device  
 3236 chooses to include additional information in the Common Name component, it shall delimit it from  
 3237 the UUID field by white space, a comma, or a semicolon.

3238 If the Device does not have a pre-provisioned key pair to use, but is capable and willing to generate  
 3239 a new key pair, the Device may begin generation of a key pair as a result of a RETRIEVE of this  
 Copyright Open Connectivity Foundation, Inc. © 2016-2020. All rights Reserved 105



3240 Resource. If the Device cannot immediately respond to the RETRIEVE request due to time required  
3241 to generate a key pair, the Device shall return an "operation pending" error. This indicates to the  
3242 Client that the Device is not yet ready to respond, but will be able at a later time. The Client should  
3243 retry the request after a short delay.

### 3244 13.10 Roles Resource

3245 The "roles" Resource maintains roles that have been asserted with role certificates, as described  
3246 in clause 10.4.2. Asserted roles have an associated public key, i.e., the public key in the role  
3247 certificate. Servers shall only grant access to the roles information associated with the public key  
3248 of the Client. The roles Resource should be viewed as an extension of the (D)TLS session state.  
3249 See 10.4.2 for how role certificates are validated.

3250 The roles Resource shall be created by the Server upon establishment of a secure (D)TLS session  
3251 with a Client, if is not already created. The roles Resource shall only expose a secured OCF  
3252 Endpoint in the "/oic/res" response. A Server shall retain the roles Resource at least as long as the  
3253 (D)TLS session exists. A Server shall retain each certificate in the roles Resource at least until the  
3254 certificate expires or the (D)TLS session ends, whichever is sooner. The requirements of clause  
3255 10.3 and 10.4.2 to validate a certificate's time validity at the point of use always apply. A Server  
3256 should regularly inspect the contents of the roles Resource and purge contents based on a policy  
3257 it determines based on its resource constraints. For example, expired certificates, and certificates  
3258 from Clients that have not been heard from for some arbitrary period of time could be candidates  
3259 for purging.

3260 The OCF namespace ("oic.role.\*") is restricted to OCF-defined roles. "oic.role.owner" is an OCF-  
3261 defined Role that is intended to provide Resource Owner privileges to multiple Clients in a scalable  
3262 way. Servers shall grant access to perform all supported operations in the current Device state  
3263 (see clause 8) on all supported SVRs regardless of ACL configuration the Clients asserting  
3264 "oic.role.owner" Role. Servers shall reject assertion of any Role, which starts with "oic.role.", but  
3265 is not one of the following Roles:

3266 – "oic.role.owner"

3267 The "roles" Resource is implicitly created by the Server upon establishment of a (D)TLS session.  
3268 In more detail, the RETRIEVE, UPDATE and DELETE operations on the roles Resource shall  
3269 behave as follows. Unlisted operations are implementation specific and not reliable.

- 3270 1) A RETRIEVE request shall return all previously asserted roles associated with the currently  
3271 connected and authenticated Client's identity. RETRIEVE requests with a "credid" query  
3272 parameter is not supported; all previously asserted roles associated with the currently  
3273 connected and authenticated Client's identity are returned.
- 3274 2) An UPDATE request that includes the "roles" Property shall replace or add to the Properties  
3275 included in the array as follows:
  - 3276 a) If either the "publicdata" or the "optionaldata" are different than the existing entries in the  
3277 "roles" array, the entry shall be added to the "roles" array with a new, unique "credid" value.
  - 3278 b) If both the "publicdata" and the "optionaldata" match an existing entry in the "roles" array,  
3279 the entry shall be considered to be the same. The Server shall reply with a 2.04 Changed  
3280 response and a duplicate entry shall not be added to the array.
  - 3281 c) The "credid" Property is optional in an UPDATE request and if included, it may be ignored  
3282 by the Server. The Server shall assign a unique "credid" value for every entry of the "roles"  
3283 array.
- 3284 3) A DELETE request without a "credid" query parameter shall remove all entries from the  
3285 "/oic/sec/roles" Resource array corresponding to the currently connected and authenticated  
3286 Client's identity.

3287 4) A DELETE request with a "credid" query parameter shall remove only the entries of the  
 3288 "/oic/sec/roles" Resource array corresponding to the currently connected and authenticated  
 3289 Client's identity and where the corresponding "credid" matches the entry.

3290 NOTE The "/oic/sec/roles" Resource's use of the DELETE operation is not in accordance with the OCF Interfaces  
 3291 defined in ISO/IEC 30118-1.

3292 See clause 8 for restrictions on the states in which this Resource may be modified.

3293 "/oic/sec/roles" Resource is defined in Table 47.

3294 **Table 47 – Definition of the "/oic/sec/roles" Resource**

| Fixed URI      | Resource Type Title | Resource Type ID ("rt" value) | OCF Interfaces                    | Description   | Related Functional Interaction |
|----------------|---------------------|-------------------------------|-----------------------------------|---|--------------------------------|
| /oic/sec/roles | Roles               | oic.r.roles                   | oic.if.basel<br>ine,<br>oic.if.rw | Resource containing roles that have previously been asserted to this Server | Security                       |

3295 Table 48 defines the Properties of "/oic/sec/roles".

3296 **Table 48 – Properties of the "/oic/sec/roles" Resource**

| Property Title | Property Name | Value Type   | Value Rule | Access Mode | Mandatory | Description                                      |
|----------------|---------------|--------------|------------|-------------|-----------|--|
| Roles          | roles         | oic.sec.cred | array      | RW          | Yes       | List of roles previously asserted to this Server |

3297 Because "/oic/sec/roles" shares the "oic.sec.cred" schema with "/oic/sec/cred", "subjectuuid" is a required Property.  
 3298 However, "subjectuuid" is not used in a role certificate. Therefore, a Device may ignore the "subjectuuid" Property if the  
 3299 Property is contained in an UPDATE request to the "/oic/sec/roles" Resource.

### 3300 13.11 Auditable Events List Resource

#### 3301 13.11.1 Auditable Events List Resource General

3302 The "/oic/sec/ael" Resource maintains a list of logged Auditable Events. Every OCF Device logs  
 3303 AEEs filtered according to the values of the "categoryfilter" and "priorityfilter" Properties of  
 3304 "/oic/sec/ael" Resource. All Devices shall have a "/oic/sec/ael" Resource to maintain AEEs. The  
 3305 new AEE shall be added to the "events" Property of "/oic/sec/ael" Resource as the last entry in the  
 3306 array. A Device shall store all AEEs of the "/oic/sec/ael" Resource in non-volatile memory. A Device  
 3307 shall be able to store at least 1 AEE.

3308 The "categoryfilter" Property determines what categories of AEEs are to be logged. The  
 3309 "categoryfilter" Property is an integer value which is a composition of bitmasks. A Device shall log  
 3310 all AEEs filtered by this value. If the "categoryfilter" is either set to 0xff or is not set, then the Device  
 3311 shall log AEEs of all categories. Refer to Table 50 for more details.

3312 The "priorityfilter" Property determines the lowest priority of AEE to be logged. A smaller value  
 3313 means higher priority. The AEEs whose "priority" Property values are equal to or smaller than this  
 3314 value shall be logged. If the "priorityfilter" Property is either set to the highest priority or is not set,  
 3315 then the Device shall log all AEEs. No matter what value is set to "priorityfilter", an AEE of CRIT  
 3316 (== 0) "priority" shall always be logged. Refer to Table 50 for more details.

3317 When an AEE is added, the "usedspace" Property shall be updated to reflect the total storage used  
 3318 by all logged events. When the reserved storage for AEEs is full, the oldest AEE shall be purged.

3319 A Device logs a new AEE as follows:

3320 5) If a new AEE is not filtered by "categoryfilter" and "priorityfilter", then it is dropped.

```
3321 /* c-like pseudo code */
3322 If ((categoryfilter & new_aee->category) && (priorityfilter >= new_aee->priority))
```

```

3323     {
3324         addAEE(new_aee);
3325     }
3326     else
3327     {
3328         free(new_aee);
3329     }

```

6) If the value of "usedspace" Property is equal to, or the sum of the "usedspace" Property value and the size of the new AEE is bigger than the value of the "maxspace" Property of "/oic/sec/ael" Resource, then:

a) Remove the oldest AEE continuously while the sum of the "usedspace" Property value and the size of the new AEE is bigger than the "maxspace" Property value.

```

3335     /* c-like pseudo code */
3336     Int addAEE(AEType *new_aee)
3337     {
3338
3339         While ((usespace + new_aee->size) > maxspace)
3340         {
3341             /* purgeAEE() returns the size of purged AEE */
3342             sizeOfPurgedAEE = purgeAEE();
3343             usedspace -= sizeOfPurgedAEE;
3344         }
3345
3346         ...
3347         ...
3348     }

```

7) Add the new AEE to the "events" array Property of the "/oic/sec/ael" Resource as the last entry in the array.

8) Increase the value of the "usedspace" Property by the size of the new AEE.

In order to provide a mechanism which allows management of the "events" array Property, the RETRIEVE and UPDATE operations on the "/oic/sec/ael" Resource shall behave as follows:

9) A RETRIEVE operation shall return the full Resource representation.

10) An UPDATE operation may set the "categoryfilter" and/or "priorityfilter" Properties.

The "/oic/sec/ael" Resource is defined in Table 49.

**Table 49 – Definition of the "/oic/sec/ael" Resource**

| Fixed URI    | Resource Type Title  | Resource Type ID ("rt" value) | OCF Interfaces                | Description               | Related Functional Interaction |
|--------------|----------------------|-------------------------------|-------------------------------|---------------------------|--------------------------------|
| /oic/sec/ael | Auditable Event List | oic.r.ael                     | oic.if.baseline,<br>oic.if.rw | Resource for storing AEEs | Security                       |

Table 50 defines the Properties of the "/oic/sec/ael" Resource.

**Table 50 – Properties of the "/oic/sec/ael" Resource**

| Property Title | Property Name | Value Type | Value Rule | Mandatory | Device State | Access Mode | Description       |
|----------------|---------------|------------|------------|-----------|--------------|-------------|-------------------|
| AEE list       | "events"      | "array"    |            | Yes       | RESET        | R           | The Device clears |

|                                       |                  |           |   |     |        |    |  |
|---------------------------------------|------------------|-----------|---|-----|--------|----|--|
|                                       |                  |           | Array of "oic.sec.aee" entries              |     | RFOTM  | R  | This list stores AEEs whose "category" Property value is filtered by "categoryfilter" Property and "priority" Property value is equal or less than the value of "priorityfilter" Property.   |
|                                       |                  |           |   |     | RFPRO  |    |  |
|                                       |                  |           |   |     | RFNOP  |    |  |
|                                       |                  |           |   |     | SRESET |    |  |
| current used storage size             | "usedspace"      | "integer" | >= 0<br>(default: 0)                        | Yes | RESET  | R  | The Device sets to 0   |
|                                       |                  |           |   |     | RFOTM  | R  | Current used space for logged AEEs. The Device updates this Property whenever new AEEs are logged.   |
|                                       |                  |           |   |     | RFPRO  |    |  |
|                                       |                  |           |   |     | RFNOP  |    |  |
|                                       |                  |           |   |     | SRESET |    |  |
| maximum allowed storage size for AEEs | "maxspace"       | "integer" | > 0   | Yes |        | R  | This means the maximum allowable storage size for AEEs that can be stored in "events" list. The Manufacturer chooses this value.   |
| unit for storage size                 | "unit"           | "string"  | enum ["Kbyte", "Byte"]<br>(default: "Byte") | No  |        | R  | The unit for "usedspace" and "maxspace" Properties. The Manufacturer chooses this value.   |
| Categories of AEE to be logged        | "categoryfilter" | "integer" | bitmask<br>(default: 0xff)                  | Yes | RESET  | R  | The Device sets to the manufacturer default value  |
|                                       |                  |           |   |     | RFOTM  | RW | This value decides what categories of AEEs are to be logged.<br>Meaning of each bit:<br><ul style="list-style-type: none"> <li>• 0x01 (Access Control)</li> <li>• 0x02 (Onboarding)</li> <li>• 0x04 (Device)</li> <li>• 0x08 (Authentication)</li> <li>• 0x10 (SVR Modification)</li> <li>• 0x20 (Cloud)</li> <li>• 0x40 (Communication)</li> <li>• 0x80 (Reserved)</li> </ul> e.g.) if "categoryfilter" == 0xff: log all events of all categories<br>e.g.) if "categoryfilter" == 0x03: log all events of 'AC' (== 0x01) and 'OB' (==0x02) categories |
|                                       |                  |           |   |     | RFPRO  |    |  |
|                                       |                  |           |   |     | RFNOP  | R  |  |
|                                       |                  |           |   |     | SRESET | RW |  |
| Minimum priority of AEEs to be logged | "priorityfilter" | "integer" | enum [0, 1, 2, 3, 4]<br>(default: 4)        | Yes | RESET  | R  | Device sets to manufacturer default value  |
|                                       |                  |           |   |     | RFOTM  | RW | The AEEs whose "priority" values are equal to or smaller than this value are logged. A smaller value means a higher priority.<br>Meaning of each value:<br><ul style="list-style-type: none"> <li>• 0 (CRIT)</li> <li>• 1 (ERR)</li> <li>• 2 (WARN)</li> <li>• 3 (INFO)</li> <li>• 4 (DEBUG)</li> </ul> e.g.) if "priorityfilter" is set to DEBUG (==4) all AEEs will be logged  |
|                                       |                  |           |   |     | RFPRO  |    |  |
|                                       |                  |           |   |     | RFNOP  | R  |  |
|                                       |                  |           |   |     | SRESET | RW |  |

|  |  |  |  |  |  |   |
|--|--|--|--|--|--|---|
|  |  |  |  |  |  | e.g.) if "priorityfilter" is set to 1, CRIT (==0) and ERR (==1) SEEs will be logged |
|--|--|--|--|--|--|---|

Table 51 defines the Properties of the "oic.sec.aee" type.

**Table 51 – "oic.sec.aee" data type definition**

| Property Title             | Property Name   | Value Type | Value Rule                            | Access Mode | Mandatory | Device State | Description   |
|----------------------------|-----------------|------------|---------------------------------------|-------------|-----------|--------------|---|
| Auditable Event Identifier | "aaid"          | "string"   | N/A                                   | R           | Yes       | -            | Identity of the logged event  |
| Category of AEE            | "category"      | "integer"  | enum<br>[1, 2, 4, 8, 16, 32, 64, 128] | R           | Yes       | -            | The category of this AEE:<br><ul style="list-style-type: none"> <li>• 0x01 (Access Control)</li> <li>• 0x02 (Onboarding)</li> <li>• 0x04 (Device)</li> <li>• 0x08 (Authentication)</li> <li>• 0x10 (SVR Modification)</li> <li>• 0x20 (Cloud)</li> <li>• 0x40 (Communication)</li> <li>• 0x80 (Reserved)</li> </ul> |
| Priority of AEE            | "priority"      | "integer"  | enum<br>[0, 1, 2, 3, 4]               | R           | Yes       | -            | The priority of this AEE:<br><ul style="list-style-type: none"> <li>• 0 (CRIT)</li> <li>• 1 (ERR)</li> <li>• 2 (WARN)</li> <li>• 3 (INFO)</li> <li>• 4 (DEBUG)</li> </ul>   |
| Time stamp                 | "timestamp"     | "string"   | date-time<br>(RFC3339 clause 5.6)     | R           | Yes       | -            | The time when the AEE occurred  |
| Event message              | "message"       | "string"   | N/A                                   | R           | Yes       | -            | The description of the logged AEE.  |
| Auxiliary info             | "auxiliaryinfo" | "array"    | Array of strings                      | R           | Yes       | -            | Supplementary information for the "message" Property<br>e.g.) URI of specific Resource in ACE2  |

OCF-defined AEEs are listed in Table 53, and each such AEE has its own values for the "category" and "priority" Properties.

The "timestamp" Property follows a full-date and partial-time format of RFC3339. Every new AEE shall have a later timestamp than the latest previously logged AEE.

The "auxiliaryinfo" Property provides supplementary info which is not covered by the description in "message" Property. For example, the URI of specific Resource in ACE2 could be "auxiliaryinfo" for "Access Denied" AEE. Please see Table 53 "List of Auditable Events".

### 13.12 Security Virtual Resources (SVRs) and Access Policy

The SVRs expose the security-related Properties of the Device.

Granting access requests (RETRIEVE, UPDATE, DELETE, etc.) for these SVRs to unauthenticated (anonymous) Clients could create privacy or security concerns.

3374 For example, when the Device onboarding State is RFOTM, it is necessary to grant requests for  
 3375 the "/oic/sec/doxm" Resource to anonymous requesters, so that the Device can be discovered and  
 3376 onboarded by an OBT. Subsequently, it might be preferable to deny requests for the  
 3377 "/oic/sec/doxm" Resource to anonymous requesters, to preserve privacy.

### 3378 13.13 SVRs, Discoverability and OCF Endpoints

3379 All implemented SVRs shall be "discoverable" (reference ISO/IEC 30118-1, Policy Parameter  
 3380 clause 7.8.2.1.2).

3381 All implemented discoverable SVRs shall expose a Secure OCF Endpoint (e.g. CoAPS) (reference  
 3382 ISO/IEC 30118-1, clause 10).

3383 The "/oic/sec/doxm" Resource shall expose an Unsecure OCF Endpoint (e.g. CoAP) in RFOTM  
 3384 (reference ISO/IEC 30118-1, clause 10).

### 3385 13.14 Additional Privacy Consideration for Core Resources

3386 Unique immutable identifiers are a privacy consideration due to their potential for being used as a  
 3387 tracking mechanism. These include the following Resources and Properties:

- 3388 – "/oic/d" Resource containing the "piid" Property.
- 3389 – "/oic/p" Resource containing the "pi" Property.

3390 These identifiers are unique values that are visible at various times throughout the Device lifecycle  
 3391 by anonymous requestors. This implies any Client Device, including those with malicious intent,  
 3392 are able to reliably obtain identifiers useful for building a log of activity correlated with a specific  
 3393 Platform and Device.

3394 The "di" Property in the "/oic/d" Resource shall mirror that of the "deviceuuid" Property of the  
 3395 "/oic/sec/doxm" Resource. The DOTS should provision an ACL policy that restricts access to the  
 3396 "/oic/d" Resource such that only authenticated Clients are able to obtain the "di" Property of "/oic/d"  
 3397 Resource. See clause 13.1 for deviceuuid Property lifecycle requirements.

3398 Servers should expose a temporary, non-repeated, "piid" Property of "/oic/d" Resource Value upon  
 3399 entering RESET. Servers shall expose a persistent value via the "piid" Property of "/oic/d" Property  
 3400 when the DOTS sets "devowneruuid" Property to a non-nil-UUID value. The DOTS should provision  
 3401 an ACL policy on the "/oic/d" Resource such that only authenticated Clients are able to obtain the  
 3402 "piid" Property of "/oic/d" Resource

3403 Servers should expose a temporary, non-repeated, "pi" Property value upon entering RESET.  
 3404 Servers shall expose a persistent value via the "pi" Property of the "/oic/p" Resource when the  
 3405 DOTS sets "devowneruuid" Property to a non-nil-UUID value. The DOTS should provision an ACL  
 3406 policy on the "/oic/p" Resource such that only authenticated Clients are able to obtain the "pi"  
 3407 Property.

3408 Table 52 depicts Core Resource Properties Access Modes given various Device States.

3409 **Table 52 – Core Resource Properties Access Modes given various Device States**

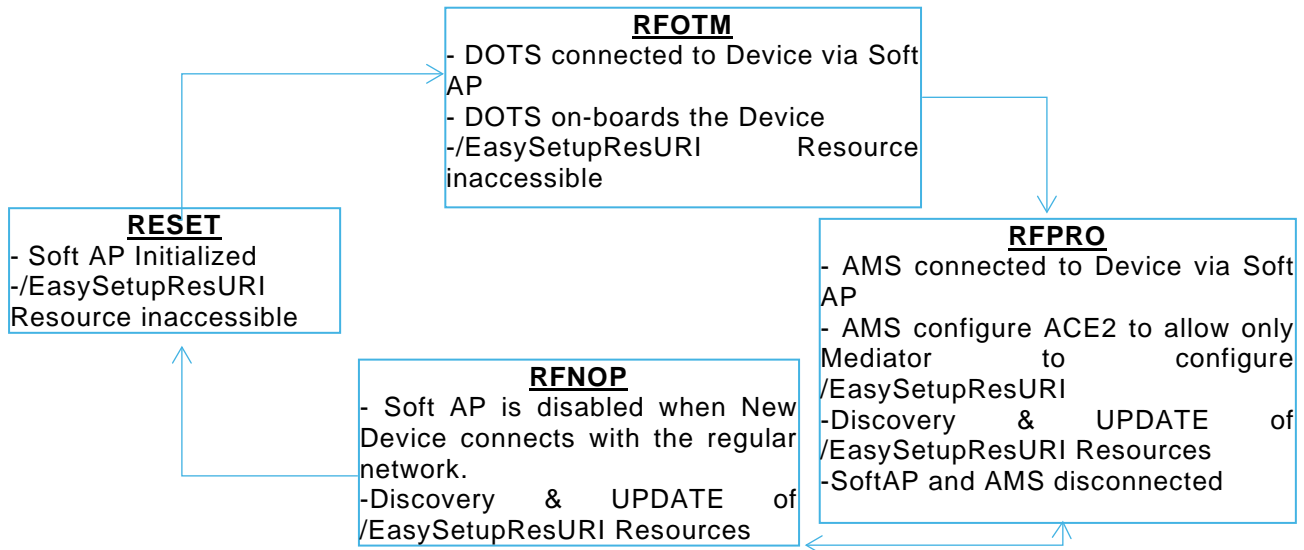
| Resource Type | Property title | Property name | Value type            | Access Mode |   | Behaviour   |
|---------------|----------------|---------------|-----------------------|-------------|---|---|
| oic.wk.p      | Platform ID    | pi            | oic.types-schema.uuid | All States  | R | Server exposes a temporary random UUID when in RESET. |

|          |                        |      |                       |            |   |   |
|----------|------------------------|------|-----------------------|------------|---|---|
| oic.wk.d | Permanent Immutable ID | piid | oic.types-schema.uuid | All States | R | Server exposes a temporary random UUID when in RESET.                           |
| oic.wk.d | Device Identifier      | di   | oic.types-schema.uuid | All states | R | /d di mirrors the value contained in "/doxm" "deviceuuid" in all device states. |

### 13.15 Easy Setup Resource Device State

This clause only applies to a new Device that uses Easy Setup for ownership transfer as defined in OCF Wi-Fi Easy Setup. Easy Setup has no impact to new Devices that have a different way of connecting to the network i.e. DOTS and AMS don't use a Soft AP to connect to non-Easy Setup Devices.

Figure 29 shows an example of Soft AP and Easy Setup Resource in different Device states.



**Figure 29 – Example of Soft AP and Easy Setup Resource in different Device states**

Device enters RFOTM, Soft AP may be accessible in RFOTM and RFPRO.

While it is reasonable for an End User to expect that power cycling a new Device will turn on the Soft AP for Easy Setup during the initial setup, since that is potentially how it behaved on first boot, it is a security risk to make this the default behaviour of a device that remains unenrolled beyond a reasonable period after first boot.

Therefore, the Soft AP for Easy Setup has several requirements to improve security:

- Time availability of Easy Setup Soft AP should be minimised, and shall not exceed 30 minutes after Device factory reset, RESET or first power boot, or when an End User initiates the Soft AP for Easy Setup.
- If a new Device tried and failed to complete Easy Setup Enrolment immediately following the first boot, or after a factory reset, it may turn the Easy Setup Soft AP back on automatically for another 30 minutes upon being power cycled, provided that the power cycle occurs within 3 hours of first boot or the most recent factory reset. If the End User has initiated the Easy Setup Soft AP directly without a factory reset, it is not necessary to turn it back on if it was on immediately prior to power cycle, because the End User obviously knows how to initiate the process manually.

3434 – After 3 hours from first boot or factory reset without successfully enrolling the device, the Soft  
3435 AP should not turn back on for Easy Setup until another factory reset occurs, or the End User  
3436 initiates the Easy Setup Soft AP directly.

3437 – Easy Setup Soft AP may stay enabled during RFNOP, until the Mediator instructs the new  
3438 Device to connect to the Enroller.

3439 – The Easy Setup Soft AP shall be disabled when the new Device successfully connects to the  
3440 Enroller.

3441 – Once a new Device has successfully connected to the Enroller, it shall not turn the Easy Setup  
3442 Soft AP back on for Easy Setup Enrolment again unless the Device is factory reset, or the End  
3443 User initiates the Easy Setup Soft AP directly.

3444 – Just Works OTM shall not be enabled on Devices which support Easy Setup.

3445 – The Soft AP shall be secured (e.g. shall not expose an open AP).

3446 – The Soft AP shall support a passphrase for connection by the Mediator, and the passphrase  
3447 shall be between and 8 and 64 ASCII printable characters. The passphrase may be printed on  
3448 a label, sticker, packaging etc., and may be entered by the End User into the Mediator device.

3449 – The Soft AP should not use a common passphrase across multiple Devices. Instead, the  
3450 passphrase may be sufficiently unique per device, to prevent guessing of the passphrase by an  
3451 attacker with knowledge of the Device type, model, manufacturer, or any other information  
3452 discoverable through Device's exposed interfaces.

3453 The Enrollee shall support WPA2 security (i.e. shall list WPA2 in the "swat" Property of the  
3454 "/example/WiFiConfResURI" Resource), for potential selection by the Mediator in connecting the  
3455 Enrollee to the Enroller. The Mediator should select the best security available on the Enroller, for  
3456 use in connecting the Enrollee to the Enroller.

3457 The Enrollee may not expose any interfaces (e.g. web server, debug port, NCRs, etc.) over the  
3458 Soft AP, other than SVRs, and Resources required for Wi-Fi Easy Setup.

3459 The "/example/EasySetupResURI" Resource should not be discoverable in RFOTM or SRESET.  
3460 After ownership transfer process is completed with the DOTS, and the Device enters in RFPRO,  
3461 the "/example/EasySetupResURI" may be Discoverable.

3462 The OTM CoAPS session may be used by Mediator for connection over Soft AP for ownership  
3463 transfer and initial Easy Setup provisioning. SoftAP or regular network connection may be used by  
3464 AMS for "/oic/sec/acl2" Resource provisioning in RFPRO. The CoAPS session authentication and  
3465 encryption is already defined in the Security spec.

3466 In RFPRO, AMS is expected to configure ACL2 Resource on the Device with ACE2 for following  
3467 Resources to be only configurable by the Mediator with permission to UPDATE or RETRIEVE  
3468 access:

3469 – "/example/EasySetupResURI"  
3470 – "/example/WifiConfResURI"  
3471 – "/example/DevConfResURI"

3472 An ACE2 granting RETRIEVE or UPDATE access to the Easy Setup Resource

3473 {  
3474     "subject": { "uuid": "<insert-UUID-of-Mediator>" },  
3475     "resources": [  
3476         { "href": "/example/EasySetupResURI" },  
3477         { "href": "/example/WiFiConfResURI" },  
3478         { "href": "/example/DevConfResURI" },  
3479     ],



3480 "permission": 6 // RETRIEVE (2) or UPDATE and RETRIEVE(6)  
 3481 }

3482 ACE2 may be re-configured after Easy Setup process. These ACE2s should be installed prior to  
 3483 the Mediator performing any RETRIEVE/UPDATE operations on these Resources.

3484 In RFPRO or RFNOP, the Mediator should discover /EasySetupResURI Resources and UPDATE  
 3485 these Resources. The Mediator may UPDATE /EasySetupResURI Resources in RFNOP Device  
 3486 state.

3487 A Mediator shall be hosted on an OCF Device.

### 3488 13.16 List of Auditable Events

3489 Whenever a Device detects an occurrence of any of the Auditable Events in Table 53, then the  
 3490 Device shall log an AEE using the corresponding "category", "priority" and "auxiliaryinfo" Properties  
 3491 defined in Table 53. The "auxiliaryinfo" Property shall contain the entries in the "auxiliaryinfo"  
 3492 column of Table 53 in the order specified in the table with each bullet contained in a separate array  
 3493 entry. The "auxiliaryinfo" Property may contain additional entries for further information following  
 3494 the entries for mandatory information. The "aaid" Property shall include the corresponding  
 3495 Auditable Event Identifier from Table 53.

3496 **Table 53 – List of mandatory Auditable Events and corresponding Property values**

| Auditable Event Identifier ("aaid") | Auditable Event Description   | Example "message"  | "category"            | "priority" | "auxiliaryinfo"  |
|-------------------------------------|---|--|-----------------------|------------|--|
| <b>AC-1</b>                         | A Device received a request from an authenticated Client with valid URI path, valid interface and valid operation for that Resource, but for which access was denied. | "Access Denied"  | 0x01 (Access Control) | 2 (WARN)   | <ul style="list-style-type: none"> <li>Client IP address &amp; port in format [xxxx:....xxxx]:xxxx</li> <li>Client UUID in UUID format (e.g. "00000000-0000-0000-0000-000000000000")</li> <li>Resource URI (e.g. "/oic/sec/ael")</li> <li>Requested CRUDN operation (e.g. "CREATE")</li> <li>Server security state (e.g. "RFNOP")</li> <li>Asserted roles by Client (e.g. "oic.role.owner"), or "No roles asserted" if there are none</li> </ul> |
| <b>AUTH-1</b>                       | The Device encountered an error during a DTLS handshaking procedure due to a credential validation failure.   | "DTLS handshake failed due to a credential validation failure" | 0x08 (Authentication) | 1 (ERR)    | <ul style="list-style-type: none"> <li>Client IP address &amp; port in format [xxxx:....xxxx]:xxxx</li> </ul>  |
| <b>COMM-1</b>                       | The Device received a CoAP request which contained unexpected /unsupported CoAP header parameters or unexpected/unsupported CoAP options.                             | "Unexpected CoAP Command"                                      | 0x40 (COMM)           | 2 (WARN)   | <ul style="list-style-type: none"> <li>Client IP address &amp; port in format [xxxx:....xxxx]:xxxx</li> <li>Hex-encoded CoAP header in format [xx:xx:xx:xx]</li> <li>Hex-encoded CoAP options except payload (empty if not present)</li> </ul>   |

3497 Whenever a Device detects an occurrence of any of the Auditable Events in Table 54, then the  
 3498 Device should log an AEE using the corresponding "category", "priority" and "auxiliaryinfo"  
 3499 Properties defined in Table 54. The "auxiliaryinfo" Property shall contain the entries in the  
 3500 "auxiliaryinfo" column of Table 54 in the order specified in the table with each bullet contained in a  
 3501 separate array entry. The "auxiliaryinfo" Property may contain additional entries for further

3502 information following the entries for mandatory information. The "aeid" Property shall include the  
3503 corresponding Auditable Event Identifier from Table 54.

3504 **Table 54 – List of recommended Auditable Events and corresponding Property values**

| Auditable Event Identifier | Auditable Event Description   | Example "message"          | "category"                 | "priority"  | "auxiliaryinfo"  |
|----------------------------|---|----------------------------|----------------------------|-------------|--|
| SVR-1                      | The Device's attempted to use one of its credentials, and detected that the credential is expired | "My credential is expired" | 0x10<br>(SVR Modification) | 2<br>(WARN) | <ul style="list-style-type: none"><li>• credid</li><li>• Credential expiration value</li></ul>                     |
| SVR-2                      | The Device could not validate the role certificate being asserted                                 | "Role assertion failed"    | 0x10<br>(SVR Modification) | 2<br>(WARN) | <ul style="list-style-type: none"><li>• Client IP address &amp; port in format [xxxx:...:xxxx]:xxx<br/>x</li></ul> |

3505

### 13.17 Security Domain Information Resource

The "/oic/sec/sdi" Resource contains the information that identifies the OCF Security Domain to which the Device belongs. OCF Security Domains are uniquely identifiable.

This Resource is optional to implement. When it is exposed by a Device, an OCF Onboarding Tool (OBT) is expected to provision a random UUID and a Security Domain Name for the OCF Security Domain. These two fields are provisioned to a Device during the onboarding process.

"oic.r.sdi" Resource Type is defined in Table 55.

**Table 55 –Definition of the "oic.r.sdi" Resource Type**

| Fixed URI    | Resource Type Title         | Resource Type ID ("rt" value) | OCF Interfaces                   | Description                                     | Related Functional Interaction |
|--------------|-----------------------------|-------------------------------|----------------------------------|---|--------------------------------|
| /oic/sec/sdi | Security Domain Information | "oic.r.sdi"                   | "oic.if.baseline"<br>"oic.if.rw" | Resource containing Security Domain information | Configuration                  |

Table 56 defines the Properties of "oic.r.sdi".

**Table 56 – Properties of the "oic.r.sdi" Resource Type**

| Property Title       | Property Name | Value Type | Value Rule | Mandatory | Access Mode | Device State | Description   |
|----------------------|---------------|------------|------------|-----------|-------------|--------------|---|
| Security Domain UUID | "uuid"        | string     | "uuid"     | Yes       | R           | RESET        | A UUID that identifies the Security Domain, set by DOTS during onboarding.  |
|                      |               |            |            |           | RW          | RFOTM        |   |
|                      |               |            |            |           | R           | RFPRO        |   |
|                      |               |            |            |           | R           | RFNOP        |   |
|                      |               |            |            |           | R           | SRESET       |   |
| Security Domain Name | "name"        | string     | N/A        | Yes       | R           | RESET        | Human-friendly name for the Security Domain, set by DOTS during onboarding.   |
|                      |               |            |            |           | RW          | RFOTM        |   |
|                      |               |            |            |           | RW          | RFPRO        |   |
|                      |               |            |            |           | R           | RFNOP        |   |
|                      |               |            |            |           | RW          | SRESET       |   |
| Privacy Flag         | "priv"        | boolean    | N/A        | Yes       | R           | RESET        | Flag to indicate whether the Security Domain Information is copied to "/oic/res", and thus whether it is publicly visible or private. |
|                      |               |            |            |           | RW          | RFOTM        |   |
|                      |               |            |            |           | RW          | RFPRO        |   |
|                      |               |            |            |           | R           | RFNOP        |   |
|                      |               |            |            |           | RW          | SRESET       |   |

The purpose of the "priv" Property is to control whether information about a Device's OCF Security Domain is exposed during multicast discoveries.

If the "priv" Property is set to "false", then the "/oic/res" Resource shall expose its "sduuid" and "sdname" Properties with values copied from the "uuid" and "name" Properties of the "/oic/sec/sdi" Resource, respectively.

If the "priv" Property is set to "true", then the "/oic/res" Resource shall not expose its "sduuid" and "sdname" Properties.

## 3523 **14 Security Hardening Guidelines/ Execution Environment Security**

### 3524 **14.1 Preamble**

3525 This is an informative clause. Many TGs in OCF have security considerations for their protocols  
3526 and environments. These security considerations are addressed through security mechanisms  
3527 specified in the security documents for OCF. However, effectiveness of these mechanisms depends  
3528 on security robustness of the underlying hardware and software Platform. This clause defines the  
3529 components required for execution environment security.

### 3530 **14.2 Execution Environment Elements**

#### 3531 **14.2.1 Execution Environment Elements General**

3532 Execution environment within a computing Device has many components. To perform security  
3533 functions in a robustness manner, each of these components has to be secured as a separate  
3534 dimension. For instance, an execution environment performing AES cannot be considered secure  
3535 if the input path entering keys into the execution engine is not secured, even though the partitions  
3536 of the CPU, performing the AES encryption, operate in isolation from other processes. Different  
3537 dimensions referred to as elements of the execution environment are listed below. To qualify as a  
3538 secure execution environment (SEE), the corresponding SEE element must qualify as secure.

- 3539 – (Secure) Storage
- 3540 – (Secure) Execution engine
- 3541 – (Trusted) Input/output paths
- 3542 – (Secure) Time Source/clock
- 3543 – (Random) number generator
- 3544 – (Approved) cryptographic algorithms
- 3545 – Hardware Tamper (protection)

3546 NOTE Software security practices (such as those covered by Open Web Application Security Project) are outside  
3547 scope of this document, as development of secure code is a practice to be followed by the open source development  
3548 community. This document will however address the underlying Platform assistance required for executing software.  
3549 Examples are secure boot and secure software upgrade.

3550 Each of the elements above are described in the clauses 14.2.2, 14.2.3, 14.2.4, 14.2.5, 14.2.6,  
3551 14.2.7.

#### 3552 **14.2.2 Secure Storage**

##### 3553 **14.2.2.1 Secure Storage General**

3554 Secure storage refers to the physical method of housing sensitive or confidential data ("Sensitive  
3555 Data"). Such data could include but not be limited to symmetric or asymmetric private keys,  
3556 certificate data, OCF Security Domain access credentials, or personal user information. Sensitive  
3557 Data requires that its integrity be maintained, whereas Critical Sensitive Data requires that both its  
3558 integrity and confidentiality be maintained.

3559 It is strongly recommended that IoT Device makers provide reasonable protection for Sensitive  
3560 Data so that it cannot be accessed by unauthorized Devices, groups or individuals for either  
3561 malicious or benign purposes. In addition, since Sensitive Data is often used for authentication and  
3562 encryption, it must maintain its integrity against intentional or accidental alteration.

3563 A partial list of Sensitive Data is outlined in Table 57:

**Table 57 – Examples of Sensitive Data**

| Data  | Integrity protection | Confidentiality protection |
|---|----------------------|----------------------------|
| Owner PSK (Symmetric Keys)                      | Yes                  | Yes                        |
| Service provisioning keys                       | Yes                  | Yes                        |
| Asymmetric Private Keys                         | Yes                  | Yes                        |
| Certificate Data and Signed Hashes              | Yes                  | Not required               |
| Public Keys                                     | Yes                  | Not required               |
| Access credentials (e.g. SSID, passwords, etc.) | Yes                  | Yes                        |
| ECDH/ECDH Dynamic Shared Key                    | Yes                  | Yes                        |
| Root CA Public Keys                             | Yes                  | Not required               |
| Device and Platform IDs                         | Yes                  | Not required               |
| Easy Setup Resources                            | Yes                  | Yes                        |
|   |                      |                            |
|   |                      |                            |
| Access Token                                    | Yes                  | Yes                        |

Exact method of protection for secure storage is implementation specific, but typically combinations of hardware and software methods are used.

#### 14.2.2.2 Hardware Secure Storage

Hardware secure storage is recommended for use with critical Sensitive Data such as symmetric and asymmetric private keys, access credentials, and personal private data. Hardware secure storage most often involves semiconductor-based non-volatile memory ("NVRAM") and includes countermeasures for protecting against unauthorized access to Critical Sensitive Data.

Hardware-based secure storage not only stores Sensitive Data in NVRAM, but also provides protection mechanisms to prevent the retrieval of Sensitive Data through physical and/or electronic attacks. It is not necessary to prevent the attacks themselves, but an attempted attack should not result in an unauthorized entity successfully retrieving Sensitive Data.

Protection mechanisms should provide JIL Moderate protection against access to Sensitive Data from attacks that include but are not limited to:

- 1) Physical decapping of chip packages to optically read NVRAM contents
- 2) Physical probing of decapped chip packages to electronically read NVRAM contents
- 3) Probing of power lines or RF emissions to monitor voltage fluctuations to discern the bit patterns of Critical Sensitive Data
- 4) Use of malicious software or firmware to read memory contents at rest or in transit within a microcontroller
- 5) Injection of faults that induce improper Device operation or loss or alteration of Sensitive Data

#### 14.2.2.3 Software Storage

It is generally NOT recommended to rely solely on software and unsecured memory to store Sensitive Data even if it is encrypted. Critical Sensitive Data such as authentication and encryption keys should be housed in hardware secure storage whenever possible.

Sensitive Data stored in volatile and non-volatile memory shall be encrypted using acceptable algorithms to prevent access by unauthorized parties through methods described in 14.2.2.2.

#### 14.2.2.4 Additional Security Guidelines and Best Practices

Some general practices that can help ensure that Sensitive Data is not compromised by various forms of security attacks:

- 1) FIPS Random Number Generator ("RNG") – Insufficient randomness or entropy in the RNG used for authentication challenges can substantially degrade security strength. For this reason, it is recommended that a FIPS 800-90A-compliant RNG with a certified noise source be used for all authentication challenges.
- 2) Secure download and boot – To prevent the loading and execution of malicious software, where it is practical, it is recommended that Secure Download and Secure Boot methods that authenticate a binary's source as well as its contents be used.
- 3) Deprecated algorithms – Algorithms included but not limited to the list below are considered unsecure and shall not be used for any security-related function:
  - a) SHA-1
  - b) MD5
  - c) RC4
  - d) RSA 1024
- 4) Encrypted transmission between blocks or components – Even if critical Sensitive Data is stored in Secure Storage, any use of that data that requires its transmission out of that Secure Storage should be encrypted to prevent eavesdropping by malicious software within an MCU/MPU.
- 5) It is recommended to avoid using wildcard in Subject Id ("\*"), when setting up "/oic/sec/cred" Resource entries, since this opens up an identity spoofing opportunity.
- 6) Device vendor understands that it is the Device vendor's responsibility to ensure the Device meets security requirements for its intended uses. As an example, IoTivity is a reference implementation intended to be used as a basis for a product, but IoTivity has not undergone 3rd party security review, penetration testing, etc. Any Device based on IoTivity should undergo appropriate penetration testing and security review prior to sale or deployment.
- 7) Device vendor agrees to publish the expected support lifetime for the Device to OCF and to consumers. Changes should be made to a public and accessible website. Expectations should be clear as to what will be supported and for how long the Device vendor expects to support security updates to the software, operating system, drivers, networking, firmware and hardware of the device.
- 8) Device vendor has not implemented test or debug interfaces on the Device which are operable or which can be enabled which might present an attack vector on the Device which circumvents the interface-level security or access policies of the Device.
- 9) Device vendor understands that if an application running on the Device has access to cryptographic elements such as the private keys or Ownership Credential, then those elements have become vulnerable. If the Device vendor is implementing a Bridge, an OBT, or a Device with access to the Internet beyond the local network, the execution of critical functions should take place within a Trusted or Secure Execution Environment (TEE/SEE).
- 10) Any PINs or fixed passphrases used for onboarding, Wi-Fi Easy Setup, SoftAP management or access, or other security-critical function, should be sufficiently unique (do not duplicate passphrases. The creation of these passphrases or PINS should not be algorithmically deterministic nor should they use insufficient entropy in their creation.
- 11) Ensure that there are no remaining "VENDOR\_TODO" items in the source code.
- 12) If the implementation of this document uses the "Just Works" onboarding method, understand that there is a man-in-the-middle vulnerability during the onboarding process where a malicious party could intercept messages between the device being onboarded and the OBT and could persist, acting as an intermediary with access to message traffic, during the lifetime of that

onboarded device. The recommended best practice would be to use an alternate ownership transfer method (OTM) instead of "Just Works".

- 13) It is recommended that at least one static and dynamic analysis tool<sup>1</sup> be applied to any proposed major production release of the software before its release, and any vulnerabilities resolved.

#### 14.2.3 Secure execution engine

Execution engine is the part of computing Platform that processes security functions, such as cryptographic algorithms or security protocols (e.g. DTLS). Securing the execution engine requires the following

- Isolation of execution of sensitive processes from unauthorized parties/ processes. This includes isolation of CPU caches, and all of execution elements that needed to be considered as part of trusted (crypto) boundary.
- Isolation of data paths into and out of execution engine. For instance, both unencrypted but sensitive data prior to encryption or after decryption, or cryptographic keys used for cryptographic algorithms, such as decryption or signing. See clause 14.2.4 for more details.

#### 14.2.4 Trusted input/output paths

Platform implementations should only expose information, network interfaces, ports and other functions that are necessary for the correct functioning of the Platform. It is also strongly recommended that Vendors configure a Platform to expose only a fixed set of explicitly documented open network ports and/or port ranges.

#### 14.2.5 Secure clock

Many security functions depend on time-sensitive credentials. Examples are time stamped Kerberos tickets, OAuth tokens, X.509 certificates, OSCP response, software upgrades, etc. Lack of secure source of clock can mean an attacker can modify the system clock and fool the validation mechanism. Thus an SEE needs to provide a secure source of time that is protected from tampering. Trustworthiness from security robustness standpoint is not the same as accuracy. Protocols such as NTP can provide rather accurate time sources from the network, but are not immune to attacks. A secure time source on the other hand can be off by seconds or minutes depending on the time-sensitivity of the corresponding security mechanism. Secure time source can be external as long as it is signed by a trusted source and the signature validation in the local Device is a trusted process (e.g. backed by secure boot).

#### 14.2.6 Approved algorithms

An important aspect of security of the entire ecosystem is the robustness of publicly vetted and peer-reviewed (e.g. NIST-approved) cryptographic algorithms. Security is not achieved by obscurity of the cryptographic algorithm. To ensure both interoperability and security, not only widely accepted cryptographic algorithms must be used, but also a list of approved cryptographic functions must be specified explicitly. As new algorithms are NIST approved or old algorithms are deprecated, the list of approved algorithms must be maintained by OCF. All other algorithms (even if they deemed stronger by some parties) must be considered non-approved.

The set of algorithms to be considered for approval are algorithms for

- Hash functions
- Signature algorithms
- Encryption algorithms
- Key exchange algorithms
- Pseudo Random functions (PRF) used for key derivation

---

<sup>1</sup> A general discussion of analysis tools can be found here: <https://www.ibm.com/developerworks/library/se-static/>

3685 This list will be included in this or a separate security robustness rules document and must be  
3686 followed for all security specifications within OCF.

#### 3687 **14.2.7 Hardware tamper protection**

3688 Various levels of hardware tamper protection exist. We borrow FIPS 140-2 terminology (not  
3689 requirements) regarding tamper protection for cryptographic module

3690 – Production-grade (lowest level): this means components that include conformal sealing coating  
3691 applied over the module's circuitry to protect against environmental or other physical damage.  
3692 This does not however require zeroization of secret material during physical maintenance. This  
3693 definition is borrowed from FIPS 140-2 security level 1.

3694 – Tamper evident/proof (mid-level), This means the Device shows evidence (through covers,  
3695 enclosures, or seals) of an attempted physical tampering. This definition is borrowed from FIPS  
3696 140-2 security level 2.

3697 – Tamper resistance (highest level), this means there is a response to physical tempering that  
3698 typically includes zeroization of sensitive material on the module. This definition is borrowed  
3699 from FIPS 140-2 security level 3.

3700 It is difficult of specify quantitative certification test cases for accreditation of these levels. Content  
3701 protection regimes usually talk about different tools (widely available, specialized and professional  
3702 tools) used to circumvent the hardware protections put in place by manufacturing. If needed, OCF  
3703 can follow that model, if and when OCF engage in distributing sensitive key material (e.g. PKI) to  
3704 its members.

### 3705 **14.3 Secure Boot**

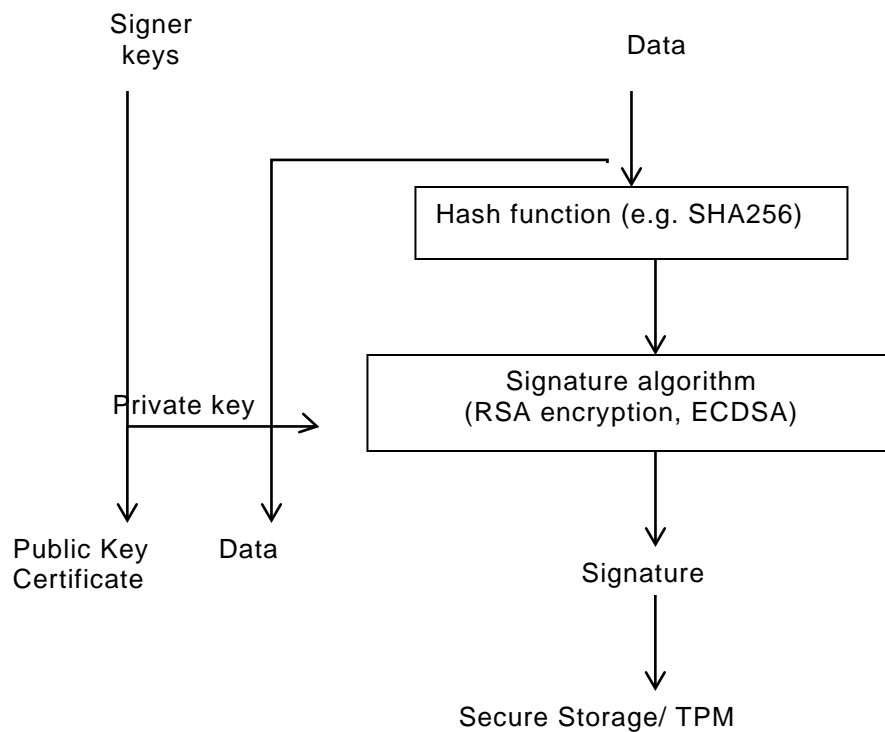
#### 3706 **14.3.1 Concept of software module authentication**

3707 In order to ensure that all components of a Device are operating properly and have not been  
3708 tampered with, it is best to ensure that the Device is booted properly. There may be multiple stages  
3709 of boot. The end result is an application running on top an operating system that takes advantage  
3710 of memory, CPU and peripherals through drivers.

3711 The general concept is that each software module is invoked only after cryptographic integrity  
3712 verification is complete. The integrity verification relies on the software module having been hashed  
3713 (e.g. SHA\_1, SHA\_256) and then signed with a cryptographic signature algorithm with (e.g. RSA),  
3714 with a key that only a signing authority has access to.

3715 Figure 30 depicts software module authentication.

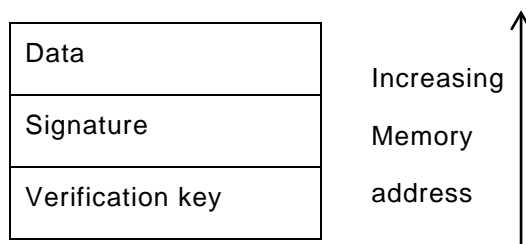




**Figure 30 – Software Module Authentication**

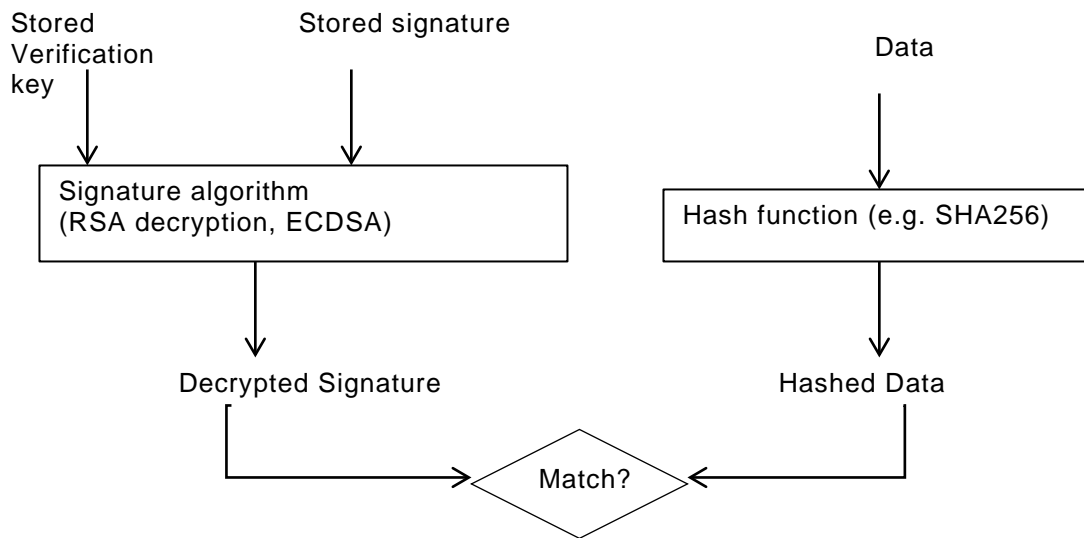
After the data is signed with the signer’s signing key (a private key), the verification key (the public key corresponding to the private signing key) is provided for later verification. For lower level software modules, such as bootloaders, the signatures and verification keys are inserted inside tamper proof memory, such as one-time programmable memory or TPM. For higher level software modules, such as application software, the signing is typically performed according to the PKCS#7 format IETF RFC 2315, where the signed data format includes both indications for signature algorithm, hash algorithm as well as the signature verification key (or certificate). Secure boot does not require use of PKCS#7 format.

Figure 31 depicts verification software module.



**Figure 31 – Verification Software Module**

As shown in Figure 32 the verification module first decrypts the signature with the verification key (public key of the signer). The verification module also calculates a hash of the data and then compares the decrypted signature (the original) with the hash of data (actual) and if the two values match, the software module is authentic.



**Figure 32 – Software Module Authenticity**

### 14.3.2 Secure Boot process

Depending on the Device implementation, there may be several boot stages. Typically, in a PC/Linux type environment, the first step is to find and run the BIOS code (first-stage bootloader) to find out where the boot code is and then run the boot code (second-stage boot loader). The second stage bootloader is typically the process that loads the operating system (Kernel) and transfers the execution to the where the Kernel code is. Once the Kernel starts, it may load external Kernel modules and drivers.

When performing a secure boot, it is required that the integrity of each boot loader is verified before executing the boot loader stage. As mentioned, while the signature and verification key for the lowest level bootloader is typically stored in tamper-proof memory, the signature and verification key for higher levels should be embedded (but attached in an easily accessible manner) in the data structures software.

### 14.3.3 Robustness Requirements

#### 14.3.3.1 Robustness General

To qualify as high robustness secure boot process, the signature and hash algorithms shall be one of the approved algorithms, the signature values and the keys used for verification shall be stored in secure storage and the algorithms shall run inside a secure execution environment and the keys shall be provided the SEE over trusted path.

#### 14.3.3.2 Next steps

Develop a list of approved algorithms and data formats

## 14.4 Attestation

## 14.5 Software Update

### 14.5.1 Overview

The Device lifecycle does not end at the point when a Device is shipped from the manufacturer; the distribution, retailing, purchase, installation/onboarding, regular operation, maintenance and end-of-life stages for the Device remain outstanding. It is possible for the Device to require update during any of these stages, although the most likely times are during onboarding, regular operation

and maintenance. The manufacturer shall have a defined policy available to OCF Security Domain Owner (e.g. via a website link) covering handling of any device vulnerabilities, including the software update information (e.g. if and how such updates are provided). This policy shall also cover any post end-of-life or end-of-service vulnerabilities. The aspects of the software include, but are not limited to, firmware, operating system, networking stack, application code, drivers, etc.

## 14.5.2 Recognition of Current Differences

Different manufacturers approach software update utilizing a collection of tools and strategies: over-the-air or wired USB connections, full or partial replacement of existing software, signed and verified code, attestation of the delivery package, verification of the source of the code, package structures for the software, etc.

It is recommended that manufacturers review their processes and technologies for compliance with industry best-practices that a thorough security review of these takes place and that periodic review continue after the initial architecture has been established.

This document applies to software updates as recommended to be implemented by OCF Devices; it does not have any bearing on the above-mentioned alternative proprietary software update mechanisms. The described steps are being triggered by an OCF Client, the actual implementation of the steps and how the software package is downloaded and upgraded is vendor specific.

The triggers that can be invoked from OCF clients can:

- 1) Check if new software is available
- 2) Download and verify the integrity of the software package
- 3) Install the verified software package

The triggers are not sequenced; each trigger can be invoked individually.

The state of the transitions of software update is in Figure 33.

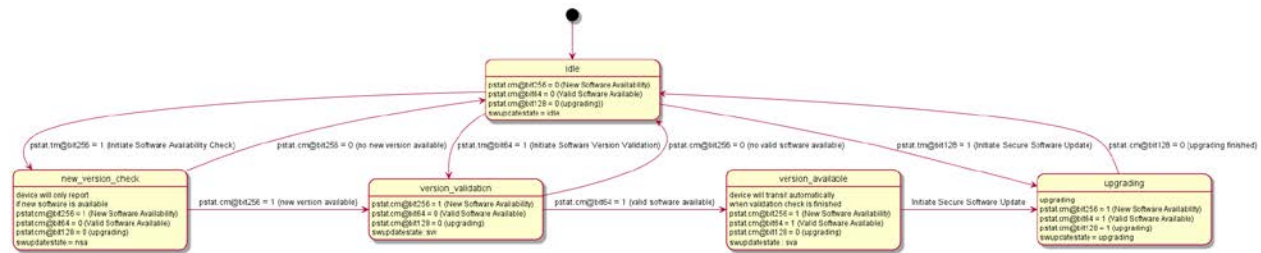


Figure 33 – State transitioning diagram for software download

Table 58 – Description of the software update bits

| Bit   | TM property                          | CM property              |
|-------|--------------------------------------|--------------------------|
| Bit 9 | Initiate Software Availability Check | New Software Available   |
| Bit 7 | Initiate Software Version Validation | Valid Software Available |
| Bit 8 | Initiate Secure Software Update      | Upgrading                |

### 14.5.2.1 Checking availability of new software

Setting the Initiate Software Availability Check bit in the "/oic/sec/pstat.tm" Property (see Table 38 of clause 13.8) indicates a request to initiate the process to check if new software is available, e.g.

the process whereby the Device checks if a newer software version is available on the external endpoint. Once the Device has determined if an newer software version is available, it sets the Initiate Software Availability Check bit in the `"/oic/sec/pstat.cm"` Property to 1 (TRUE), indicating that new software is available or to 0 (FALSE) if no newer software version is available, See also Table 58 where the bits in property TM indicates that the action is initiated and the CM bits are indicating the result of the action. The Device receiving this trigger is not downloading and not validating the software to determine if new software is available. The version check is determined by the current software version and the software version on the external endpoint. The determination if a software package is newer is vendor defined.

### 14.5.3 Software Version Validation

Setting the Initiate Software Version Validation bit in the `"/oic/sec/pstat.tm"` Property (see Table 38 of 13.8) indicates a request to initiate the software version validation process, the process whereby the Device validates the software (including firmware, operating system, Device drivers, networking stack, etc.) against a trusted source to see if, at the conclusion of the check, the software update process will need to be triggered (see clause 14.5.4). When the Initiate Software Version Validation bit of `"/oic/sec/pstat.tm"` is set to 1 (TRUE) by a sufficiently privileged Client, the Device sets the `"/oic/sec/pstat.cm"` Initiate Software Version Validation bit to 0 and initiates a software version check. Once the Device has determined if a valid software is available, it sets the Initiate Software Version Validation bit in the `"/oic/sec/pstat.cm"` Property to 1 (TRUE) if an update is available or 0 (FALSE) if no update is available. To signal completion of the Software Version Validation process, the Device sets the Initiate Software Version Validation bit in the `"/oic/sec/pstat.tm"` Property back to 0 (FALSE). If the Initiate Software Version Validation bit of `"/oic/sec/pstat.tm"` is set to 0 (FALSE) by a Client, it has no effect on the validation process. The Software Version Validation process can download the software from the external endpoint to verify the integrity of the software package.

### 14.5.4 Software Update

The software of a Device shall be updatable.

Setting the Initiate Secure Software Update bit in the `"/oic/sec/pstat.tm"` Property (see Table 38 of clause 13.8) indicates a request to initiate the software update process. When the Initiate Secure Software Update bit of `"/oic/sec/pstat.tm"` is set to 1 (TRUE) by a sufficiently privileged Client, the Device sets the `"/oic/sec/pstat.cm"` Initiate Software Version Validation bit to 0 and initiates a software update process. Once the Device has completed the software update process, it sets the Initiate Secure Software Update bit in the `"/oic/sec/pstat.cm"` Property to 1 (TRUE) if/when the software was successfully updated or 0 (FALSE) if no update was performed. To signal completion of the Secure Software Update process, the Device sets the Initiate Secure Software Update bit in the `"/oic/sec/pstat.tm"` Property back to 0 (FALSE). If the Initiate Secure Software Update bit of `"/oic/sec/pstat.tm"` is set to 0 (FALSE) by a Client, it has no effect on the update process.

#### 14.5.4.1 State of Device after software update

The state of all Resources implemented in the Device should be the same as after boot, meaning that the software update is not resetting user data and retaining a correct state.

User data of a Device is defined as:

- Retain the SVR states, e.g. the on boarded state, registered clients.
- Retain all created Resources
- Retain all stored data of a Resource
  - For example the preferences stored for the brewing Resource (`"/oic.r.brewing"`).

### 14.5.5 Recommended Usage

The Initiate Secure Software Update bit of `"/oic/sec/pstat.tm"` should only be set by a Client after the Initiate Software Version Validation check is complete.

3837 The process of updating Device software may involve state changes that affect the Device  
3838 Operational State ("/oic/sec/pstat.dos"). Devices with an interest in the Device(s) being updated  
3839 should monitor "/oic/sec/pstat.dos" and be prepared for pending software update(s) to affect Device  
3840 state(s) prior to completion of the update.

3841 The Device itself may indicate that it is autonomously initiating a software version check/update or  
3842 that a check/update is complete by setting the "pstat.tm" and "pstat.cm" Initiate Software Version  
3843 Validation and Secure Software Update bits when starting or completing the version check or  
3844 update process. As is the case with a Client-initiated update, Clients can be notified that an  
3845 autonomous version check or software update is pending and/or complete by observing pstat  
3846 Resource changes.

3847 The "oic.r.softwareupdate" Resource Type specifies additional features to control the software  
3848 update process see core specification.

#### 3849 **14.6 Non-OCF Endpoint interoperability**

#### 3850 **14.7 Security Levels**

3851 Security Levels are a way to differentiate Devices based on their security criteria. This need for  
3852 differentiation is based on the requirements from different verticals such as industrial and health  
3853 care and may extend into smart home. This differentiation is distinct from Device classification  
3854 (e.g. IETF RFC 7228)

3855 These categories of security differentiation may include, but is not limited to:

- 3856 1) Security Hardening
- 3857 2) Identity Attestation
- 3858 3) Certificate/Trust
- 3859 4) Onboarding Technique
- 3860 5) Regulatory Compliance
  - 3861 a) Data at rest
  - 3862 b) Data in transit
- 3863 6) Cipher Suites – Crypto Algorithms & Curves
- 3864 7) Key Length
- 3865 8) Secure Boot/Update

3866 In the future security levels can be used to define interoperability.

3867 The following applies to the OCF Security Specification 1.1:

3868 The current document does not define any other level beyond Security Level 0. All Devices will be  
3869 designated as Level 0. Future versions may define additional levels.

3870 Additional comments:

- 3871 – The definition of a given security level will remain unchanged between versions of the document.
- 3872 – Devices that meet a given level may, or may not, be capable of upgrading to a higher level.
- 3873 – Devices may be evaluated and re-classified at a higher level if it meets the requirements of the  
3874 higher level (e.g. if a Device is manufactured under the 1.1 version of the document, and a later  
3875 document version defines a security level 1, the Device could be evaluated and classified as  
3876 level 1 if it meets level 1 requirements).
- 3877 – The security levels may need to be visible to the End User.

## **14.8 Security Profiles**

### **14.8.1 Security Profiles General**

Security Profiles are a way to differentiate OCF Devices based on their security criteria. This need for differentiation is based on the requirements from different verticals such as industrial and health care and may extend into smart home. This differentiation is distinct from device classification (e.g. IETF RFC 7228)

These categories of security differentiation may include, but is not limited to:

- 1) Security Hardening and assurances criteria
- 2) Identity Attestation
- 3) Certificate/Trust
- 4) Onboarding Technique
- 5) Regulatory Compliance
  - a) Data at rest
  - b) Data in transit
- 6) Cipher Suites – Crypto Algorithms & Curves
- 7) Key Length
- 8) Secure Boot/Update

Each Security Profile definition must specify the version or versions of the OCF Security Specification(s) that form a baseline set of normative requirements. The profile definition may include security requirements that supersede baseline requirements (not to relax security requirements).

Security Profiles have the following properties:

- A given profile definition is not specific to the version of the document that defines it. For example, the profile may remain constant for subsequent OCF Security Specification versions.
- A specific OCF Device and platform combination may be used to satisfy the security profile.
- Profiles may have overlapping criteria; hence it may be possible to satisfy multiple profiles simultaneously.
- An OCF Device that satisfied a profile initially may be re-evaluated at a later time and found to satisfy a different profile (e.g. if a device is manufactured under the 1.1 version of the document, and a later document version defines a security profile Black, the device could be evaluated and classified as profile Black if it meets profile Black requirements).
- A machine-readable representation of compliance results specifically describing profiles satisfied may be used to facilitate OCF Device onboarding. (e.g. a manufacturer certificate or manifest may contain security profiles attributes).

### **14.8.2 Identification of Security Profiles (Normative)**

#### **14.8.2.1 Security Profiles in Prior Documents**

OCF Devices conforming to versions of the OCF Security Specifications where Security Profiles Resource was not defined may be presumed to satisfy the "sp-baseline-v0" profile (defined in 14.8.3.3) or may be regarded as unspecified. If Security Profile is unspecified, the Client may use the OCF Security Specification version to characterize expected security behaviour.

#### **14.8.2.2 Security Profile Resource Definition**

The "/oic/sec/sp" Resource is used by the OCF Device to show which OCF Security Profiles the OCF Device is capable of supporting and which are authorized for use by the OCF Security Domain owner. Properties of the Resource identify which OCF Security Profile is currently operational. The

3922 ocfSecurityProfileOID value type shall represent OID values and may reference an entry in the form  
3923 of strings (UTF-8).

3924 "/oic/sec/sp" Resource is defined in Table 59.

3925 **Table 59 – Definition of the "/oic/sec/sp" Resource**

| Fixed URI   | Resource Type Title                  | Resource Type ID ("rt" value) | OCF Interfaces                 | Description   | Related Functional Interaction |
|-------------|--------------------------------------|-------------------------------|--------------------------------|---|--------------------------------|
| /oic/sec/sp | Security Profile Resource Definition | oic.r.sp                      | oic.if.baselin<br>e, oic.if.rw | Resource specifying supported and current security profile(s) | Discoverable                   |

3926 Table 60 defines the Properties of "/oic/sec/sp" Resource.

3927 **Table 60 – Properties of the "/oic/sec/sp" Resource**

| Property Title              | Property Name     | Value Type            | Value Rule | Access Mode | Mandatory | Description   |
|-----------------------------|-------------------|-----------------------|------------|-------------|-----------|---|
| Supported Security Profiles | supportedprofiles | ocfSecurityProfileOID | array      | RW          | Yes       | Array of supported Security Profiles (e.g. ["1.3.6.1.4.1.51414.0.0.2.0","1.3.6.1.4.1.51414.0.0.3.0"]) |
| SecurityProfile             | currentprofile    | ocfSecurityProfileOID | N/A        | RW          | Yes       | Currently active Security Profile (e.g. "1.3.6.1.4.1.51414.0.0.3.0")                                  |

3928 The following OIDs are defined to uniquely identify Security Profiles. Future Security Profiles or  
3929 changes to existing Security Profiles may result in a new ocfSecurityProfileOID.

3930 id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)  
3931 private(4) enterprise(1) OCF(51414) }

3932  
3933 id-ocfSecurity OBJECT IDENTIFIER ::= { id-OCF 0 }

3934  
3935 id-ocfSecurityProfile ::= { id-ocfSecurity 0 }

3936  
3937 sp-unspecified ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 0 }  
3938 --The Security Profile is not specified  
3939 sp-baseline ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 1 }  
3940 --This specifies the OCF Baseline Security Profile(s)  
3941 sp-black ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 2 }  
3942 --This specifies the OCF Black Security Profile(s)  
3943 sp-blue ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 3 }  
3944 --This specified the OCF Blue Security Profile(s)  
3945 sp-purple ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 4 }  
3946 --This specifies the OCF Purple Security Profile(s)

3947  
3948 --versioned Security Profiles  
3949 sp-unspecified-v0 ::= ocfSecurityProfileOID (id-sp-unspecified 0)  
3950 --v0 of unspecified security profile, "1.3.6.1.4.1.51414.0.0.0.0"  
3951 sp-baseline-v0 ::= ocfSecurityProfileOID {id-sp-baseline 0}  
3952 --v0 of baseline security profile, "1.3.6.1.4.1.51414.0.0.1.0"  
3953 sp-black-v0 ::= ocfSecurityProfileOID {id-sp-black 0}  
3954 --v0 of black security profile, "1.3.6.1.4.1.51414.0.0.2.0"  
3955 sp-blue-v0 ::= ocfSecurityProfileOID {id-sp-blue 0}  
3956 --v0 of blue security profile, "1.3.6.1.4.1.51414.0.0.3.0"  
3957 sp-purple-v0 ::= ocfSecurityProfileOID {id-sp-purple 0}  
3958 --v0 of purple security profile, "1.3.6.1.4.1.51414.0.0.4.0"

3959  
3960 ocfSecurityProfileOID ::= UTF8String  
3961

### 3962 **14.8.3 Security Profiles**

#### 3963 **14.8.3.1 Security Profiles General**

3964 The Security Profiles Resource shall be pre-populated with manufacturer default values (Refer to  
3965 the Security Profile clauses for additional details).

3966 The OCF Conformance criteria may require vendor attestation that establishes the expected  
3967 environment in which the OCF Device is hosted (Refer to the Security Profile clauses for specific  
3968 requirements).

#### 3969 **14.8.3.2 Security Profile Unspecified (sp-unspecified-v0)**

3970 The Security Profile "sp-unspecified-v0" is reserved for future use.

#### 3971 **14.8.3.3 Security Profile Baseline v0 (sp-baseline-v0)**

3972 The Security Profile "sp-baseline-v0" is defined for all OCF Security Specification versions where  
3973 the "/oic/sec/sp" Resource is defined. All Devices shall include the "sp-baseline-v0" OID in the  
3974 "supportedprofiles" Property of the "/oic/sec/sp" Resource.

3975 It indicates the OCF Device satisfies the normative security requirements for this document.

3976 When a device supports the baseline profile, the "supportedprofiles" Property shall contain sp-  
3977 baseline-v0, represented by the OID string "1.3.6.1.4.1.51414.0.0.1.0", and may contain other  
3978 profiles.

3979 When a manufacturer makes sp-baseline-v0 the default, by setting the "currentprofile" Property to  
3980 "1.3.6.1.4.1.51414.0.0.1.0", the "supportedprofiles" Property shall contain sp-baseline-v0.

#### 3981 **14.8.3.4 Security Profile Black (sp-black-v0)**

##### 3982 **14.8.3.4.1 Black Profile General**

3983 The need for Security Profile Black v0 is to support devices and manufacturers who wish to certify  
3984 their devices meeting this specific set of security criteria. A Device may satisfy the Black  
3985 requirements as well as requirements of other profiles, the Black Security Profile is not necessarily  
3986 mutually exclusive with other Security Profiles unless those requirements conflict with the explicit  
3987 requirements of the Black Security Profile.

##### 3988 **14.8.3.4.2 Devices Targeted for Security Profile Black v0**

3989 Security Profile Black devices could include any device a manufacturer wishes to certify at this  
3990 profile, but healthcare devices and industrial devices with additional security requirements are the  
3991 initial target. Additionally, manufacturers of devices at the edge of the network (or fog), or devices  
3992 with exceptional profiles of trust bestowed upon them, may wish to certify at this profile; these types  
3993 of devices may include, but are not limited to the following:

- 3994 – Bridges (Mapping devices between ecosystems handling virtual devices from different  
3995 ecosystems)
- 3996 – Resource Directories (Devices trusted to manage OCF Security Domain Resources)
- 3997 – Remote Access (Devices which have external access but can also act within the OCF Security  
3998 Domain)
- 3999 – Healthcare Devices (Devices with specific requirements for enhanced security and privacy)
- 4000 – Industrial Devices (Devices with advanced management, security and attestation requirements)

##### 4001 **14.8.3.4.3 Requirements for Certification at Security Profile Black (Normative)**

4002 Every device with "currentprofile" Property of the "/oic/sec/sp" Resource designating a Security  
4003 Profile of "sp-black-v0", as defined in clause 14.8.2, must support each of the following:



- 4004 – Onboarding via OCF Rooted Certificate Chain, including PKI chain validation
- 4005 – Support for AES 128 encryption for data at rest and in transit.
- 4006 – Hardening minimums: manufacturer assertion of secure credential storage
- 4007 – In – in enumerated item #10 “The “/oic/sec/cred” Resource should contain credential(s) if
- 4008 required by the selected OTM” is changed to require the credential be stored: “The
- 4009 “/oic/sec/cred” Resource shall contain credential(s).”
- 4010 – The OCF Device shall include an X.509v3 OCF Compliance Extension (clause 9.4.2.2.4) in its
- 4011 certificate and the extension's 'securityProfile' field shall contain sp-black-v0 represented by
- 4012 the ocfSecurityProfileOID string, "1.3.6.1.4.1.51414.0.0.2.0".
- 4013 When a device supports the black profile, the "supportedprofiles" Property shall contain sp-black-
- 4014 v0, represented by the OID string "1.3.6.1.4.1.51414.0.0.2.0", and may contain other profiles.
- 4015 When a manufacturer makes sp-black-v0 the default, by setting the "currentprofile" Property to
- 4016 "1.3.6.1.4.1.51414.0.0.2.0", the "supportedprofiles" Property shall contain sp-black-v0.
- 4017 The OCF Rooted Certificate Chain and PKI Is defined by and structured within a framework
- 4018 described in the supporting documents:
- 4019 – Certificate Profile (See 9.4.2)
- 4020 – Certificate Policy (see Certificate Policy document:
- 4021 <https://openconnectivity.org/specs/OCF%20Certificate%20Policy.pdf>)
- 4022 **14.8.3.5 Security Profile Blue v0 (sp-blue-v0)**
- 4023 **14.8.3.5.1 Blue Profile General**
- 4024 The Security Profile Blue is used when manufacturers issue platform certificates for platforms
- 4025 containing manufacturer-embedded keys. Compatibility with interoperable trusted platforms is
- 4026 anticipated using certificate extensions defined by the Trusted Computing Group (TCG). OCF
- 4027 Security Domain owners evaluate manufacturer supplied certificates and attributed data to
- 4028 determine an appropriate OCF Security Profile that is configured for OCF Devices at onboarding.
- 4029 OCF Devices may satisfy multiple OCF Security Profiles. The OCF Security Domain owner may
- 4030 configure deployments using the Security Profile as OCF Security Domain partitioning criteria.
- 4031 Certificates issued to Blue Profile Devices shall be issued by a CA conforming to the CA Vetting
- 4032 Criteria defined by OCF.
- 4033 **14.8.3.5.2 Platforms and Devices for Security Profile Blue v0**
- 4034 The OCF Security Profile Blue anticipates an ecosystem where platform vendors may differ from
- 4035 OCF Device vendor and where platform vendors may implement trusted platforms that may conform
- 4036 to industry standards defining trusted platforms. The OCF Security Profile Blue specifies
- 4037 mechanisms for linking platforms with OCF Device(s) and for referencing quality assurance criteria
- 4038 produced by OCF conformance operations. The OCF Security Domain owner evaluates these data
- 4039 when an OCF Device is onboarded into the OCF Security Domain. Based on this evaluation the
- 4040 OCF Security Domain owner determines which Security Profile may be applied during OCF Device
- 4041 operation. All OCF Device types may be considered for evaluation using the OCF Security Profile
- 4042 Blue.
- 4043 **14.8.3.5.3 Requirements for Certification at Security Profile Blue v0**
- 4044 The OCF Device satisfies the Blue profile v0 (sp-blue-v0) when all of the security normative for this
- 4045 document version are satisfied and the following additional criteria are satisfied.
- 4046 OCF Blue profile defines the following OCF Device quality assurances:

- 4047 – The OCF Conformance criteria shall require vendor attestation that the conformant OCF Device  
4048 was hosted on one or more platforms that satisfies OCF Blue platform security assurances and  
4049 platform security and privacy functionality requirements.
- 4050 – The OCF Device achieving OCF Blue Security Profile compliance will be registered by OCF and  
4051 published by OCF in a machine readable format.
- 4052 – The OCF Blue Security Profile compliance registry may be digitally signed by an OCF owned  
4053 signing key.
- 4054 – The OCF Device shall include an X.509v3 OCF Compliance Extension (clause 9.4.2.2.4) in its  
4055 certificate and the extension's 'securityProfile' field shall contain sp-blue-v0 represented by the  
4056 ocfSecurityProfileOID string, "1.3.6.1.4.1.51414.0.0.3.0".
- 4057 – The OCF Device shall include an X.509v3 OCF CPL Attributes Extension (clause 9.4.2.2.7) in  
4058 its certificate.
- 4059 – The DOTS is expected to perform a lookup of the certification status of the OCF Device using  
4060 the OCF CPL Attributes Extension values and verify that the sp-blue-v0 OID is listed in the  
4061 extension's "securityprofiles" field.
- 4062 OCF Blue profile defines the following OCF Device security functionality:
  - 4063 – OCF Device(s) shall be hosted on a platform where a cryptographic and secure storage  
4064 functions are hardened by the platform.
  - 4065 – OCF Device(s) hosted on a platform shall expose accompanying manufacturer credentials using  
4066 the "/oic/sec/cred" Resource where the "credusage" Property contains the value  
4067 "oic.sec.cred.mfgcert".
  - 4068 – OCF Device(s) that are hosted on a TCG-defined trusted platform should use an IEEE802.1AR  
4069 IDevID and should verify the "TCG Endorsement Key Credential". All TCG-defined  
4070 manufacturer credentials may be identified by the "oic.sec.cred.mfgcert" value of the  
4071 "credusage" Property of the "/oic/sec/cred" Resource. They may be used in response to  
4072 selection of the "oic.sec.doxm.mfgcert" owner transfer method.
  - 4073 – OCF Device(s) shall use AES128 equivalent minimum protection for transmitted data. (See  
4074 NIST SP 800-57).
  - 4075 – OCF Device(s) shall use AES128 equivalent minimum protection for stored data. (See NIST SP  
4076 800-57).
  - 4077 – OCF Device(s) should use AES256 equivalent minimum protection for stored data. (See NIST  
4078 SP 800-57).
  - 4079 – OCF Device(s) should protect the "/oic/sec/cred" Resource using the platform provided secure  
4080 storage.
  - 4081 – OCF Device(s) shall protect trust anchors (aka policy defining trusted CAs and pinned  
4082 certificates) using platform provided secure storage.
  - 4083 – OCF Device(s) should check certificate revocation status for locally issued certificates.
  - 4084 – The DOTS is expected to check certificate revocation status for all certificates in manufacturer  
4085 certificate path(s) if available. If a certificate is revoked, certificate validation fails and the  
4086 connection is refused. The DOTS may disregard revocation status results if unavailable.
- 4087 OCF Blue profile defines the following platform security assurances:
  - 4088 – Platforms implementing cryptographic service provider (CSP) functionality and secure storage  
4089 functionality should be evaluated with a minimum FIPS140-2 Level 2 or Common Criteria EAL  
4090 Level 2.
  - 4091 – Platforms implementing trusted platform functionality should be evaluated with a minimum  
4092 Common Criteria EAL Level 1.
- 4093 OCF Blue profile defines the following platform security and privacy functionality:

- 4094 – The Platform shall implement cryptographic service provider (CSP) functionality.
- 4095 – Platform CSP functionality shall include cryptographic algorithms, random number generation,
- 4096 secure time.
- 4097 – The Platform shall implement AES128 equivalent protection for transmitted data. (See NIST SP
- 4098 800-57).
- 4099 – The Platform shall implement AES128 and AES256 equivalent protection for stored data. (See
- 4100 NIST SP 800-57).
- 4101 – Platforms hosting OCF Device(s) should implement a platform identifier following IEEE802.1AR
- 4102 or Trusted Computing Group(TCG) specifications.
- 4103 – Platforms based on Trusted Computing Group (TCG) platform definition that host OCF Device(s)
- 4104 should supply TCG-defined manufacture certificates; also known as "TCG Endorsement Key
- 4105 Credential" (which complies with IETF RFC 5280) and "TCG Platform Credential" (which
- 4106 complies with IETF RFC 5755).
- 4107 When a device supports the blue profile, the "supportedprofiles" Property shall contain sp-blue-v0,
- 4108 represented by the OID string "1.3.6.1.4.1.51414.0.0.3.0", and may contain other profiles.
- 4109 When a manufacturer makes sp-blue-v0 the default, by setting the "currentprofile" Property to
- 4110 "1.3.6.1.4.1.51414.0.0.3.0", the "supportedprofiles" Property shall contain sp-blue-v0.
- 4111 During onboarding, while the device state is RFOTM, the DOTS may update the "currentprofile"
- 4112 Property to one of the other values found in the "supportedprofiles" Property.
- 4113 **14.8.3.6 Security Profile Purple v0 (sp-purple-v0)**
- 4114 Every device with the "/oic/sec/sp" Resource designating "sp-purple-v0", as defined in clause
- 4115 14.8.2 must support following minimum requirements
- 4116 – Hardening minimums: secure credential storage, software integrity validation, secure update.
- 4117 – If a Certificate is used, the OCF Device shall include an X.509v3 OCF Compliance Extension
- 4118 (clause 9.4.2.2.4) in its certificate and the extension's 'securityProfile' field shall contain sp-
- 4119 purple-v0 represented by the ocfSecurityProfileOID string, "1.3.6.1.4.1.51414.0.0.4.0"
- 4120 – The OCF Device shall include a X.509v3 OCF CPLAttributes Extension (clause 9.4.2.2.7) in its
- 4121 End-Entity Certificate when manufacturer certificate is used.
- 4122 Security Profile Purple has following optional security hardening requirements that the device can
- 4123 additionally support.
- 4124 – Hardening additions: secure boot, hardware backed secure storage
- 4125 – The OCF Device shall include a X.509v3 OCFSecurityClaims Extension (clause 9.4.2.2.6) in its
- 4126 End-Entity Certificate and it shall include corresponding OIDs to the hardening additions
- 4127 implemented and attested by the vendor. If there is no additional support for hardening
- 4128 requirements, X.509v3 OCFSecurityClaims Extension shall be omitted.
- 4129 For software integrity validation, OCF Device(s) shall provide the integrity validation mechanism
- 4130 for security critical executables such as cryptographic modules or secure service applications, and
- 4131 they should be validated before the execution. The key used for validating the integrity must be
- 4132 pinned at the least to the validating software module.
- 4133 For secure update, OCF Device(s) shall be able to update its firmware in a secure manner.
- 4134 For secure boot, OCF Device(s) shall implement the BIOS code (first-stage bootloader on ROM) to
- 4135 be executed by the processor on power-on, and secure boot parameters to be provisioned by
- 4136 tamper-proof memory. Also OCF Device(s) shall provide software module authentication for the
- 4137 security critical executables and stop the boot process if any integrity of them is compromised.

4138 For hardware backed secure storage, OCF Device(s) shall store sensitive data in non-volatile  
4139 memory ("NVRAM") and prevent the retrieval of sensitive data through physical and/or electronic  
4140 attacks.

4141 More details on security hardening guidelines for software integrity validation, secure boot, secure  
4142 update, and hardware backed secure storage are described in 14.3, 14.5 and 14.2.2.2.

4143 Certificates issued to Purple Profile Devices shall be issued by a CA conforming to the CA Vetting  
4144 Criteria defined by OCF.

4145 When a device supports the purple profile, the "supportedprofiles" Property shall contain sp-purple-  
4146 v0, represented by the OID string "1.3.6.1.4.1.51414.0.0.4.0", and may contain other profiles.

4147 When a manufacturer makes sp-purple-v0 the default, by setting the "currentprofile" Property to  
4148 "1.3.6.1.4.1.51414.0.0.4.0", the "supportedprofiles" Property shall contain sp-purple-v0.

## 4149 **15 Device Type Specific Requirements**

### 4150 **15.1 Bridging Security**

#### 4151 **15.1.1 Universal Requirements for Bridging to another Ecosystem**

4152 The Bridge shall go through OCF ownership transfer as any other onboarder would.

4153 The software of a Bridge shall be field updatable. (This requirement need not be tested but can be  
4154 certified via a vendor declaration.)

4155 Each VOD shall be onboarded by an OCF OBT. Each Virtual Bridged Device should be provisioned  
4156 as appropriate in the Bridged Protocol. In other words, VODs and Virtual Bridged Devices are  
4157 treated the same way as physical Devices. They are entities that have to be provisioned in their  
4158 network.

4159 Each VOD shall implement the behaviour required by ISO/IEC 30118-1 and this document. Each  
4160 VOD shall perform authentication, access control, and encryption according to the security settings  
4161 it received from the OCF OBT. Each Virtual Bridged Device shall implement the security  
4162 requirements of the Bridged Protocol.

4163 In addition, in order to be considered secure from an OCF perspective, the Bridge Platform shall  
4164 use appropriate ecosystem-specific security options for communication between the Virtual Bridged  
4165 Devices instantiated by the Bridge and Bridged Devices. This security shall include mutual  
4166 authentication, and encryption and integrity protection of messages in the bridged ecosystem.

4167 A VOD may authenticate itself to the DOTS using the Manufacturer Certificate Based OTM (see  
4168 clause 7.3.6) with the Manufacturer Certificate and corresponding private key of the Bridge which  
4169 instantiated that VOD.

4170 A VOD may authenticate itself to the OCF Cloud using the Manufacturer Certificate and  
4171 corresponding private key of the Bridge which instantiated that VOD.

4172 A Bridge and the VODs created by that Bridge shall operate as independent Devices, with the  
4173 following exceptions:

- 4174 – If a Bridge creates a VOD while the Bridge is in an Unowned State, then the VOD shall be  
4175 created in an Unowned State.
- 4176 – An Unowned VOD shall not accept DTLS connection attempts nor TLS connection attempts nor  
4177 any other requests, including discovery requests, while the Bridge (that created that VOD) is  
4178 Unowned.
- 4179 – At any time when a Bridge is transitioning from Owned to Unowned State, all Unowned VODs  
4180 (created by that Bridge prior to the transition) shall drop any existing TLS and/or DTLS  
4181 connections.
- 4182 – At any time when a Bridge is transitioning from Unowned to Owned State, the Bridge shall  
4183 trigger all Unowned VODs (created by that Bridge prior to the transition) to become accessible  
4184 in RFOTM, with internal state as if the VOD has just transitioned from RESET to RFOTM.
- 4185 – If a Bridge creates a VOD while the Bridge is in an Owned State, then the VOD shall become  
4186 accessible in RFOTM, with internal state as if the VOD has just transitioned from RESET to  
4187 RFOTM.

4188 Table 61 intends to clarify this behaviour.

**Table 61 – Dependencies of VOD Behaviour on Bridge state, as clarification of accompanying text**

| Bridge state  | Additional dependencies on VOD behaviour   |                      |
|---|--|----------------------|
|   | VOD is Unowned (either just created, or created previously)  | VOD is Owned         |
| From unboxing Bridge until just prior to the end of transition of Bridge from Unowned to Owned              | No accepting DTLS connection attempts nor TLS connection attempts nor any other requests, including discovery requests | Not applicable       |
| At end of transition from Unowned to Owned  | VOD becomes accessible in RFOTM following Bridge's transition. Internal state as if just transitioned from RESET.      | As per normal Device |
| Owned   | As per normal Device   | As per normal Device |
| At Start of transition from Owned to Unowned  | Drop any established TLS/DTLS connections, even if already partway through Device ownership                            | As per normal Device |
| Start of transition from Owned to Unowned, until just prior to the end of transition from Unowned to Owned. | No accepting DTLS connection attempts nor TLS connection attempts nor any other requests, including discovery requests | As per normal Device |

The "vods" Property of the "oic.r.vodlist" Resource on a Bridge reflects the details of all currently Owned VODs which have been created by that Bridge since the most recent hardware reset (if any) of the Bridge Platform (which removes all the created VODs), regardless of whether the VODs have the same owner as the Bridge or not. The entries in the "vods" Property are added and removed according to the following criteria:

- Whenever a VOD created by a Bridge transitions from being Unowned to being Owned, then an entry for that VOD shall be added to the "vods" Property of the "oic.r.vodlist" Resource of that Bridge.
- Whenever a VOD created by a Bridge transitions from being Owned to being Unowned, then entry for that VOD shall be removed from the "vods" Property of the "oic.r.vodlist" Resource of that Bridge. If that Bridge is currently in Unowned state, then the "oic.r.vodlist" Resource is not accessible, and the entry for that VOD shall be removed from the "vods" Property before or during the transition of that Bridge to the Owned state.
- All other modifications of the list are not allowed.

A Bridge shall only expose a secure OCF Endpoint for the "oic.r.vodlist" Resource.

## **15.1.2 Additional Security Requirements specific to Bridged Protocols**

### **15.1.2.1 Additional Security Requirements specific to the AllJoyn Protocol**

For AllJoyn translator, an authenticated and authorized Client shall be able to block the communication of all OCF Devices with all Bridged Devices that don't communicate securely with the Bridge, by using the Bridge Device's "oic.r.securemode" Resource specified in ISO/IEC 30118-3

### **15.1.2.2 Additional Security Requirements specific to the Bluetooth LE Protocol**

A Bridge shall block the communication of all OCF Devices with all Bridged Devices that don't communicate securely with the Bridge.

### **15.1.2.3 Additional Security Requirements specific to the oneM2M Protocols**

The Bridge shall implement oneM2M application access control as defined in the oneM2M Release 3 Specifications.

An Bridge shall block the communication of all OCF Devices with all Bridged Devices that don't communicate securely with the Bridge.

4220   **15.1.2.4   Additional Security Requirements specific to the U+ Protocol**  
4221   A Bridge shall block the communication of all OCF Devices with all Bridged Devices that don't  
4222   communicate securely with the Bridge.

4223   **15.1.2.5   Additional Security Requirements specific to the Z-Wave Protocol**  
4224   A Bridge shall block the communication of all OCF Devices with all Bridged Devices that don't  
4225   communicate securely with the Bridge.

4226   **15.1.2.6   Additional Security Requirements specific to the Zigbee Protocol**  
4227   A Bridge shall block the communication of all OCF Devices with all Bridged Devices that don't  
4228   communicate securely with the Bridge.

4229   **15.1.2.7   Additional Security Requirements specific to the EnOcean Radio Protocol**  
4230   A Bridge shall block the communication of all OCF Devices with all Bridged Devices that don't  
4231   communicate securely with the Bridge.

4232  
4233  
4234  
4235  
4236  
4237  
4238  
4239  
4240  
4241  
4242  
4243  
4244  
4245  
4246  
4247  
4248  
4249  
4250  
4251  
4252   .

## Annex A (informative) Access Control Examples

### 16 Alternative in-transit protection mechanisms

#### 16.1 Introduction to in-transit protection mechanisms

In addition to the DTLS protection mechanisms for device-to-device communication specified in clause 10 and clause 11.2, and TLS protection specified in OCF Cloud Security Specification, OCF supports the following in-transit protection mechanisms:

- End-to-End Security of Unicast Messages using OSCORE, specified in clause 16.2.
- Simple Secure Multicast, specified in clause 16.3

#### 16.2 End-to-End Security of Unicast Messages using OSCORE

##### 16.2.1 Introduction to End-to-End Security of Unicast Messages using OSCORE

End-to-End Security of Unicast Messages is accomplished by applying a layer of in-transit protection above the transport layer Security (provided by DTLS or TLS) and below the resource-access authorization layer, using Object Security for Constrained RESTful Environments (OSCORE) IETF RFC 8613.

Relative to an exchange of an OCF CRUDN Request message and OCF CRUDN Response message:

- The Device acting as a Client (that is, sending an OCF CRUDN Request message and receiving the corresponding OCF CRUDN Response message) acts as an OSCORE client. Within the scope of clause 16.2, all Clients are assumed to support OSCORE and perform OSCORE client processing.
- The Device acting as a Server (that is, receiving an OCF CRUDN Request message and sending one or more corresponding OCF CRUDN Response messages) acts as an OSCORE server. Within the scope of clause 16.2, all Servers are assumed to support OSCORE and perform OSCORE server processing.

Clause 16.2.4 specifies the supported mechanism for establishing an OSCORE Security Context between two Devices. For each Device, an authorized Client (e.g. OBT) provisions the OSCORE Security Context parameters to a credential entry of the "/oic/sec/cred" Resource. The "subjectuuiid" of that credential entry identifies the other Device that shares that OSCORE Security Context (similar to how a DTLS endpoint associates each DTLS PSK session with the Device UUID of the other DTLS endpoint).

##### 16.2.2 OSCORE ID Namespace Prefix

Clause 16.2.4 specifies one mechanism for establishing an OSCORE Security Context between two Devices. Different mechanisms have different entities responsible for managing the selection of OSCORE Sender ID and OSCORE Recipient ID. There is value in preventing Devices having multiple OSCORE Security Contexts with identical Recipient IDs: this simplifies processing and avoids inefficiencies.

If a set of one or more coordinated entities (e.g. a group of OBTs) assigns a set of OSCORE Recipient IDs to OSCORE Security Contexts on a Device, then that set of entities is responsible for avoiding duplicate OSCORE Recipient IDs. However, two non-coordinated entities assigning OSCORE Recipient IDs might assign identical OSCORE Recipient IDs if there is no predefined agreement on assignment of OSCORE Recipient IDs.

For this reason, the first byte of the OSCORE Sender ID and OSCORE Recipient ID use a OSCORE Identifier Namespace Prefix. The Table Y is the authoritative definition of the assigned OSCORE Identifier Namespace Prefix values.



**Table 62 – OSCORE Identifier Namespace Prefix**

| Value     | Interpretation                               | Applicable clauses |
|-----------|--|--------------------|
| 0x00      | Reserved for future use                      |                    |
| 0x01      | Directly provisioned OSCORE Security Context | 16.2.4             |
| 0x02      | Simple Secure Multicast                      | 16.3               |
| 0x03-0x0F | Reserved for future use                      |                    |

### 4300 **16.2.3 OSCORE protection and verification of unicast OCF CRUDN messages**

4301 All OSCORE message processing requirements in clause 8 in IETF RFC 8613 apply.

4302 NOTE 1: Clause 8 in IETF RFC 8613 requires the Client keep the association of the request Token (see IETF RFC 7252)  
 4303 with the Security Context and Partial IV of the request, in order to be able to find the Security Context and compute the  
 4304 OSCORE Additional Authenticated Data when verifying the response.

4305 If a Client has an established OSCORE Security Context associated with a Server, then the  
 4306 following call flow applies whenever the Client sends unicast OCF CRUDN request targeting  
 4307 Resources hosted on the Server. The Client may send multiple OSCORE requests to multiple  
 4308 Servers

4309 1) The Client shall apply the OSCORE request protection processing to OCF CRUDN requests  
 4310 targeting Resources hosted on the Server as specified in clause 8.1 in IETF RFC 8613, using  
 4311 the OSCORE Security Context. See ISO/IEC 30118-1 for details on setting the Proxy-URI  
 4312 option.

4313 The Client sends the OSCORE request message to the Server (optionally via OCF Proxies).  
 4314 The OSCORE request message shall be delivered over secure transports: Device-to-Device  
 4315 communication is secured as specified in clause 10; Device to Cloud communication is secured  
 4316 as specified in OCF Cloud Specification and OCF Cloud Security Specification; and Cloud-to-  
 4317 Cloud communication is secured as specified in OCF Cloud API for Cloud Services  
 4318 Specification.

4319 2) The Server receives a unicast OSCORE request message. The Server shall apply the OSCORE  
 4320 request verification and decryption processing in clause 8.2 of IETF RFC 8613 with the  
 4321 following clarifications:

4322 a) At Step 2 in clause 8.2 of IETF RFC 8613

4323 i) If either the decompression or the COSE message fails to decode, the Server shall  
 4324 respond with error response message (e.g. "Bad Option") including an Outer Max-Age  
 4325 option with value zero.

4326 ii) The Server attempts to retrieve the OSCORE Security Contexts associated with the  
 4327 Recipient ID in the 'kid' parameter. If the Server fails to retrieve a OSCORE Security  
 4328 Context with OSCORE Recipient ID corresponding to the 'kid' parameter received, then  
 4329 the Server shall respond with an error response message (e.g. "Unauthorized")  
 4330 including an Outer Max-Age option with value zero.

4331 b) At step 6 in clause 8.2 of IETF RFC 8613, if the decryption failed then the Server shall  
 4332 respond with an error response message (e.g. "Bad Request") including an Outer Max-Age  
 4333 option with value zero.

4334 c) If a Server exposes one or more observable Resources, then the Server shall support  
 4335 receiving OSCORE request messages using the Observe option.

4336 3) The Server shall process the OCF CRUDN request message (encapsulated in the OSCORE  
 4337 request message) resulting in OCF CRUDN response message(s). The Server shall treat the  
 4338 value of "subjectuuid" in the credential entry which contains the OSCORE Security Context  
 4339 used to verify and decrypt the OSCORE request message in Step 2 as Client's Device UUID  
 4340 for access control processing. The Server shall treat the connection type as "auth-crypt" for  
 4341 access control processing.

4342 NOTE 2: Multiple OCF CRUDN response messages are only sent in scenarios where the OCF CRUDN Request message  
4343 is an Observe Request message.

4344 4) The Server shall apply the OSCORE response protection processing of clause 8.3 of IETF RFC  
4345 8613 to each OCF CRUDN response message, using the OSCORE Security Context used to  
4346 successfully decrypt the OSCORE request (in Step 2 of the present clause).  
4347 At Step 3 in clause 8.3 of IETF RFC 8613, the Server shall compute the AEAD nonce as  
4348 described in clause 5.2 of IETF RFC 8613 by applying the following steps:

4349 a) Encode the Partial IV (OSCORE Sender Sequence Number in network byte order) and  
4350 increment the OSCORE Sender Sequence Number by one.

4351 b) Compute the OSCORE AEAD nonce from the Sender ID, Common IV, and Partial IV.

4352 The Server shall support sending the OCF CRUDN response messages using the Observe  
4353 option in OSCORE response messages. If an OCF CRUDN response message uses the  
4354 Observe option, then the OSCORE response message shall include an Outer Max-Age option  
4355 with value zero. The Server sends the OSCORE response message to the Client (optionally via  
4356 OCF Proxies). As with the OSCORE request message, the OSCORE response message shall  
4357 be delivered over secure transports - see Step 1 for details.

4358 The Server shall update the value of the "ssn" Property in the matching credential entry of the  
4359 "/oic/sec/cred" Resource to reflect the next value of the OSCORE Sender Sequence Number  
4360 to be sent to a corresponding Endpoint.

4361 NOTE 3: If a Client retrieves the "/oic/sec/cred" Resource over the OSCORE channel, the OSCORE Sender Sequence  
4362 Number in the header of the OSCORE message is expected to match the "ssn" value within the Resource representation.

4363 5) The Client receives the OSCORE response message. The Client uses the Token (see IETF  
4364 RFC 7252) in this response message to determine the corresponding OCF CRUDN request  
4365 message, the OSCORE Security Context and Partial IV in Step 1 of the present clause; see  
4366 Note 1. The Client shall apply OSCORE response protection processing of clause 8.3 of IETF  
4367 RFC 8613 using this OSCORE Security Context and Partial IV. The Client should ignore a  
4368 success response to an OSCORE-protected request if the response is not an OSCORE  
4369 response message (indicated by the presence of the OSCORE option).

#### 4370 **16.2.4 Direct provisioning of an OSCORE Security Context**

4371 This is a mechanism for establishing an OSCORE Security Context for communication between  
4372 two Endpoints. All configurable parameters of the OSCORE Security Context are either:

- 4373 – fixed to the OSCORE-specified default value, or
- 4374 – directly provisioned by an authorized Client (e.g. OBT) to a credential entry of the  
4375 "/oic/sec/cred" Resource of the two Endpoints.

4376 The following OSCORE Security Context parameters shall use the default values defined in clause  
4377 3.2 of IETF RFC 8613 (this information is not configured by the OBT):

- 4378 – AEAD Algorithm,
- 4379 – HKDF,
- 4380 – Replay Window,
- 4381 – Master Salt,
- 4382 – ID Context.

4383 The following OSCORE Security Context parameters and associated Device UUID shall be  
4384 provisioned to a credential entry of "/oic/sec/cred" of the Device:

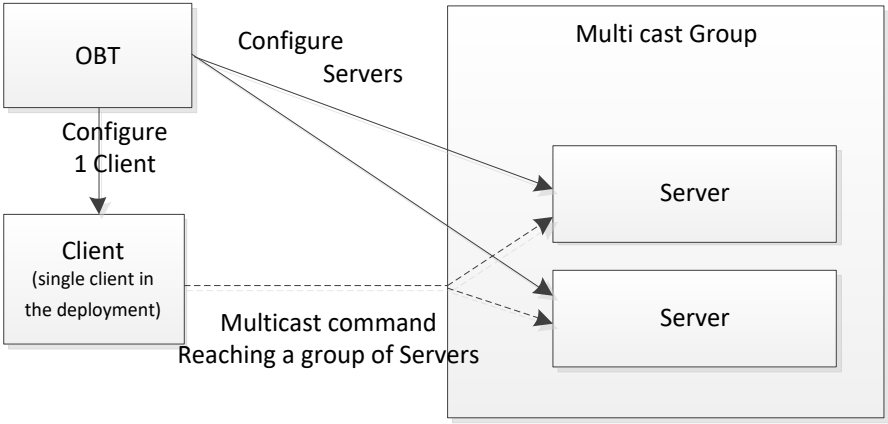
- 4385 – The "subjectuuid" shall be set to the deviceUUID of the other Endpoint to be associated with  
4386 the OSCORE Security Context.
- 4387 – The "credtype" shall be set to the value specified for a directly provisioned OSCORE Security  
4388 Context in Table 21, clause 13.3.1.

- 4389 – The "privatedata" Property of the credential entry shall be set to the 256-bit secret generated  
4390 by the provisioning client (e.g. OBT). This value shall be used as the OSCORE Master Secret.  
4391 Two Endpoints provisioned using this mechanism can communicate securely only if provisioned  
4392 with identical values for the OSCORE Master Secret.
- 4393 – The OSCORE Configuration parameters ("oscore") Property shall be present, and shall include  
4394 the following Properties:
- 4395 – The OSCORE Sender ID of the OSCORE Security Context is in the "senderid" Property.  
4396 That value shall be set to the hexadecimal representation of a 56-bit value selected by the  
4397 provisioning Client (e.g. OBT). When using the mechanism described in the present clause,  
4398 the first byte of this value is expected to have the value assigned in Table 62 for a directly  
4399 provisioned OSCORE Security Context.
  - 4400 – The OSCORE Recipient ID of the OSCORE Security Context is in the "recipientid" Property.  
4401 That value shall be set to the hexadecimal representation of a 56-bit value selected by the  
4402 provisioning Client (e.g. OBT). The first byte of this value is expected to have the value  
4403 assigned in Table 62 for a directly provisioned OSCORE Security Context.
- 4404 NOTE: The values for the OSCORE Sender ID and OSCORE Recipient ID of the OSCORE Security Context for one  
4405 Device are provisioned as the values for the OSCORE Recipient ID and OSCORE Sender ID of the OSCORE Security  
4406 Context for the other Device respectively.
- 4407 On Device powering down, for each such credential entry, the Device shall write the value of  
4408 corresponding OSCORE Sender Sequence Number as "ssn" Property to non-volatile memory. In  
4409 event of a crash, devices should apply Appendix B.1.1 of IETF RFC 8613.

4410 **16.3 Simple Secure Multicast**

4411 **16.3.1 Introduction to Simple Secure Multicast**

4412 The communication model is that one (1) Client communicates to a group of Servers with a single  
4413 UPDATE request, as shown in Figure 34. Each Server receives the UPDATE request at  
4414 approximately the same time and can execute the UPDATE request at approximately the same  
4415 time. As example of this kind of communication is sending an "on" command to a group of lights,  
4416 all lights that are member of that group turn on at approximately the same time. Sending UPDATE  
4417 requests to a group of devices can be achieved on IP by means of sending messages to a  
4418 predefined URL on a multicast address.



4419

4420 **Figure 34 – Simple Multicast requests**

4421 Security of SSM is accomplished by applying an application layer of in-transit protection and below  
4422 the resource-access authorization layer, using OSCORE IETF RFC 8613.

4423 Relative to an exchange of an UPDATE non-confirmable message:

- 4424 – The Device acting as a Client (that is, sending an UPDATE request message) acts as an  
4425 OSCORE client. Within the scope of clause 16.3 the single Client is assumed to support  
4426 OSCORE and perform OSCORE client processing.
- 4427 – The Device acting as a Server (that is, receiving an UPDATE request message) acts as an  
4428 OSCORE server. Within the scope of clause 16.3, all Servers are assumed to support OSCORE  
4429 and perform OSCORE server processing.

4430 Clause 16.3.2 details the assumptions and prerequisites for correct functioning of SSM. Clause  
4431 16.3.3 describes the process for encapsulating an UPDATE request message into an SSM Request  
4432 at the Client of an SSM Group, and subsequent extraction of an UPDATE request message from  
4433 an SSM Request at the Server of an SSM Group. Clause 16.3.4 specifies how a Client generates  
4434 an OSCORE Common Context and OSCORE Sender Context from an SSM Client Context and  
4435 specifies how a Server generates an OSCORE Common Context and OSCORE Recipient Context  
4436 from an SSM Server Context.

### 4437 **16.3.2 Assumptions and Prerequisites for Simple Secure Multicast**

4438 As shown in the following example, any Server of the SSM Group can generate an SSM Request  
4439 which other Servers in the SSM Group will interpret as being securely sent by the Client of the  
4440 SSM Group, for the purposes of privilege escalation. The security of SSM relies on the assumption  
4441 that no Server in the SSM Group attempts to generate an SSM Request using the credentials for  
4442 the SSM Group. SSM should only be used in scenarios where the Security Domain Owner is  
4443 confident that this is a valid assumption.

4444 SSM Requests are delivered to SSM-capable Servers via the All OCF Nodes multicast address  
4445 defined in ISO/IEC 30118-1. As specified in ISO/IEC 30118-1, all Servers subscribe to this multicast  
4446 address to facilitate discovery of "oic/res", and consequently all Servers can receive SSM Requests  
4447 delivered in this manner. A Server that supports the reception of SSM Requests for one or more  
4448 Resources that it hosts shall populate the All OCF nodes multicast address in the "eps" Parameter  
4449 of the Resource Links of those Resources in the "oic/res" discovery response.

4450 The configured Client is aware of Multicast enabled Servers by means of detecting the multicast  
4451 enabled resources in the Device discovery "oic/res" responses. The Client also knows how to  
4452 create the multicast request to that resource, by means of the Introspection Device Data hosted on  
4453 the Device. Therefore, the Client is able to send an UPDATE request to the multicast enabled  
4454 Resources.

4455 The Client of an SSM Group cannot form SSM Requests for the SSM Group until the Client is  
4456 provisioned with the SSM Client Context for the SSM Group. Likewise, each Server in an SSM  
4457 Group cannot process SSM Requests for the SSM Group until the Server is provisioned with the  
4458 SSM Server Context for the SSM Group. The SSM Client Context and SSM Server Context are  
4459 provisioned by an OBT as specified in OCF Onboarding Tool Specification. Clause 16.3.4 specifies  
4460 how the OSCORE Sender Context at a Client is derived from an SSM Client Context, and how the  
4461 OSCORE Recipient Context at a Server is derived from an SSM Server Context.

4462 The UPDATE request encapsulated in an SSM Request includes a local URI path for a target  
4463 Resource. A Server in the SSM Group for whom the request is intended, will process the request  
4464 using the Resource at this local URI path, if such a Resource exists and the Resource matches the  
4465 Resource Type and OCF Interface in the request. The SSM feature is designed with the  
4466 assumption that the local URI path, Resource Type and supported OCF Interfaces on the intended  
4467 Servers are consistent; but the SSM feature does not specify how such consistency is achieved.

4468 The UPDATE request message itself is expected to contain information in such way that the Server  
4469 can determine if the received UPDATE request message is intended for the Server, but the  
4470 specification of this information is not part of the SSM feature.

### 4471 **16.3.3 OSCORE protection and verification of Simple Secure Multicast Requests**

4472 All OSCORE message processing requirements in clauses 8.1 and 8.2 in IETF RFC 8613 apply.

4473 If a Client has an established SSM Client Context associated with an SSM Group, then the following  
4474 call flow applies whenever the Client sends a multicast non-confirmable UPDATE request targeting  
4475 multicast enabled Resources hosted on one or more Servers of the SSM Group.

4476 1) The Client shall apply the OSCORE request protection processing to the UPDATE request as  
4477 specified in clause 8.1 in IETF RFC 8613, using the OSCORE Security Context derived from  
4478 the SSM Client Context as specified in clause 16.3.4. See ISO/IEC 30118-1 for details on  
4479 setting the Proxy-URI option.

4480 The Client shall send the resulting OSCORE request message to the predefined All OCF Nodes  
4481 multicast address. Dependent on the deployment scenario the different scopes as defined in  
4482 clause 12.2.9 of ISO/IEC 30118-1 can be used.

4483 2) All Servers subscribed to the predefined multicast address receive a copy of the OSCORE  
4484 request message. Each Server supporting SSM which receives the OSCORE request message  
4485 shall apply the OSCORE request verification and decryption processing in clause 8.2 of IETF  
4486 RFC 8613 with the following clarifications:

4487 a) At Step 2 in clause 8.2 of IETF RFC 8613

4488 i) If either the decompression or the COSE message fails to decode, the Server shall  
4489 ignore the message and shall not respond.

4490 ii) The Server attempts to retrieve the SSM Server Contexts with "recipientID" matching  
4491 the 'kid' parameter. If the Server fails to retrieve an SSM Server Context with  
4492 "recipientID" matching the 'kid' parameter received, then the Server shall ignore the  
4493 message and shall not respond.

4494 b) At step 6 in clause 8.2 of IETF RFC 8613, if the decryption failed then the Server shall  
4495 ignore the message and shall not respond.

4496 3) If any of the following criteria are met, then the CRUDN request message shall be silently  
4497 discarded, and a response shall not be sent:

4498 – The operation of the CRUDN request is not the non-confirmable UPDATE operation on a  
4499 multicast address.

4500 – The UPDATE request message is not intended for the Server – see clause 16.3.2 for further  
4501 details.

4502 – There is no Resource hosted on the Server at the local URI path in the UPDATE request  
4503 message.

4504 4) The Server shall process the UPDATE request message (encapsulated in the OSCORE request  
4505 message). The Server shall treat the value of "subjectuuid" in the credential entry which  
4506 contains the OSCORE Security Context used to verify and decrypt the OSCORE request  
4507 message in Step 2 as Client's Device UUID for access control processing. The Server shall  
4508 treat the connection type as "auth-crypt" for access control processing. The Server shall not  
4509 send a response.

4510 The mechanism outlined is for sending a message in a send and forget mode, i.e. sending a  
4511 message to a group of Servers, where each Server does not acknowledge the receipt. Since  
4512 multicast requests are typically unreliable (e.g. non-confirmable messages) the best practice is to  
4513 send the same UPDATE request more than once in a short time frame. This is sufficient since the  
4514 multicast communication has in most cases a unicast variant for the same UPDATE request.

4515 Notification (see clause 11.3 of ISO/IEC 30118-1) may be used to verify if the actual UPDATE  
4516 request has been executed. If a subset of the group of Servers did not receive the UPDATE request,  
4517 unicast (confirmable) messages can be used to complete the desired overall state of the system.

#### 4518 **16.3.4 Creating OSCORE Security Context for Simple Secure Multicast**

4519 The present clause specifies how

- 4520 – a Client of an SSM Group creates a OSCORE Security Context from a SSM Client Context  
4521 provisioned to a credential entry of the Client.
- 4522 – a Server of an SSM Group creates a OSCORE Security Context from a SSM Server Context  
4523 provisioned to a credential entry of the Server.

4524 All configurable parameters of the OSCORE Security Context are either:

- 4525 – fixed to the OSCORE-specified default value, or
- 4526 – directly provisioned by an OBT to a credential entry of the "/oic/sec/cred" Resource.

4527 The following parameters of the OSCORE Security Context used for encryption by the Client of an  
4528 SSM Group shall be set to the default values defined in clause 3.2 of IETF RFC 8613 (this  
4529 information is not configured by the OBT):

- 4530 – AEAD Algorithm,
- 4531 – HKDF,
- 4532 – Master Salt,
- 4533 – ID Context.

4534 The following parameters of the OSCORE Security Context parameters used for encryption by the  
4535 Client of an SSM Group are derived from the SSM Client Context provisioned to a credential entry  
4536 of "/oic/sec/cred" of the Client:

- 4537 – The "subjectuuid" may be any schema compliant value. This Property serves no purpose when  
4538 used in an SSM Client Context.
- 4539 – The credential entry is identified as an SSM Client Context when the "credtype" matches the  
4540 value specified for a SSM Client Context in Table 21, clause 13.3.1.
- 4541 – The "privatedata" Property contains a 256-bit value which shall be used as the OSCORE Master  
4542 Secret.
- 4543 – The OSCORE Configuration parameters ("oscore") Property is present, and includes the  
4544 following Properties:
  - 4545 – The "senderid" Property shall be used as the OSCORE Sender ID of the OSCORE Security  
4546 Context. The "recipientid" Property value shall be interpreted as the hexadecimal  
4547 representation of a 56-bit value. The first byte of this value is expected to have the value  
4548 assigned in Table Y for Simple Secure Multicast.
  - 4549 – The "desc" Property is not used in security processing. This Property is described in clause  
4550 9.3.9.

4551 On the Device shutting down, for each such credential entry, the Device shall write the value of  
4552 corresponding OSCORE Sender Sequence Number as "ssn" Property to non-volatile memory. In  
4553 event of a crash, devices should apply Appendix B.1.1 of IETF RFC 8613.

4554 The following parameters of the OSCORE Security Context used by a Server of an SSM Group for  
4555 verification and decryption shall be set to the default values defined in clause 3.2 of IETF RFC  
4556 8613 (this information is not configured by the OBT):

- 4557 – AEAD Algorithm,
- 4558 – HKDF,

- 4559 – Replay Window,
- 4560 – Master Salt,
- 4561 – ID Context.

4562 The following parameters of the OSCORE Security Context parameters used by a Server of an  
4563 SSM Group for verification and decryption are derived from the SSM Server Context provisioned  
4564 to a credential entry of "/oic/sec/cred" of the Server:

- 4565 – The "subjectuuid" is used for access control processing as described in Step 4 of clause 16.3.3.
- 4566 – The credential entry is identified as an SSM Server Context when the "credtype" matches to  
4567 the value specified for an SSM Server Context in Table 21, clause 13.3.1.
- 4568 – The "privatedata" Property of the credential entry contains a 256-bit value which shall be used  
4569 as the OSCORE Master Secret.
- 4570 – The OSCORE Configuration parameters ("oscore") Property is present, and includes the  
4571 following Properties:
- 4572 – The "recipientid" Property shall be used as the OSCORE Recipient ID of the OSCORE Security  
4573 Context. The "recipientid" Property value shall be interpreted as the hexadecimal representation  
4574 of a 56-bit value. The first byte of this value is expected to have the value assigned in Table Y  
4575 for Simple Secure Multicast.
- 4576 – The "desc" Property is not used in security processing. This Property is described in clause  
4577 9.3.9.

## 4578 **A.1 Example OCF ACL Resource**

4579 Figure A-1 shows how an "/oic/sec/acl2" Resource could be configured to enforce an example  
4580 access policy on the Server.

```
4581 {
4582     "aclist2": [
4583         {
4584             // Subject with ID ...01 should access two named Resources with access mode "CRUDN" (Create, Retrieve, Update,
4585             Delete and Notify)
4586             "subject": {"uuid": "XXXX-...-XX01"},
4587             "resources": [
4588                 {"href": "/oic/sh/light/1"},
4589                 {"href": "/oic/sh/temp/0"}
4590             ],
4591             "permission": 31, // 31 dec = 0b0001 1111 which maps to ---N DURC
4592             "validity": [
4593                 // The period starting at 18:00:00 UTC, on January 1, 2015 and
4594                 // ending at 07:00:00 UTC on January 2, 2015
4595                 "period": ["20150101T180000Z/20150102T070000Z"],
4596                 // Repeats the {period} every week until the last day of Jan. 2015.
4597                 "recurrence": ["RRULE:FREQ=WEEKLY;UNTIL=20150131T070000Z"]
4598             ],
4599             "aceid": 1
4600         }
4601     ],
4602     // An ACL provisioning and management service should be identified as
4603     // the resource owner
```

```
4604     "owneruuid": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1"  
4605 }  
4606
```

**Figure A-1 – Example "/oic/sec/acl2" Resource**



**Annex B**  
**(Informative)**  
**Execution Environment Security Profiles**

Given that IoT verticals and Devices will not be of uniform capabilities, a one-size-fits all security robustness requirements meeting all IOT applications and services will not serve the needs of OCF, and security profiles of varying degree of robustness (trustworthiness), cost and complexity have to be defined. To address a large ecosystem of vendors, the profiles can only be defined as requirements and the exact solutions meeting those requirements are specific to the vendors' open or proprietary implementations, and thus in most part outside scope of this document.

To align with the rest of OCF documents, where Device classifications follow IETF RFC 7228 (Terminology for constrained node networks) methodology, we limit the number of security profiles to a maximum of 3 (see Table B.1). However, our understanding is OCF capabilities criteria for each of 3 classes will be more fit to the current IoT chip market than that of IETF.

Given the extremely low level of resources at class 0, our expectation is that class 0 Devices are either capable of no security functionality or easily breakable security that depend on environmental (e.g. availability of human) factors to perform security functions. This means the class 0 will not be equipped with an SEE.

**Table B.1 – OCF Security Profile**

| Platform class | SEE | Robustness level |
|----------------|-----|------------------|
| 0              | No  | N/A              |
| 1              | Yes | Low              |
| 2              | Yes | High             |

NOTE This analysis acknowledges that these Platform classifications do not take into consideration of possibility of security co-processor or other hardware security capability that augments classification criteria (namely CPU speed, memory, storage).

## Annex C (normative) Resource Type definitions

### C.1 List of Resource Type definitions

Table C.1 contains the list of defined security Resources in this document.

**Table C.1 – Alphabetized list of security Resources**

| Friendly Name (informative)  | Resource Type (rt) | Clause |
|------------------------------|--------------------|--------|
| Access Control List 2        | oic.r.acl2         | C.2    |
| Auditable Event List         | oic.r.ael          | C.9    |
| Certificate Signing Request  | oic.r.csr          | C.4    |
| Credential                   | oic.r.cred         | C.3    |
| Device owner transfer method | oic.r.doxm         | C.5    |
| Device Provisioning Status   | oic.r.pstat        | C.6    |
| Roles                        | oic.r.roles        | C.7    |
| Security Profile             | oic.r.sp           | C.8    |
| Security Domain Information  | oic.r.sdi          | C.10   |

### C.2 Access Control List-2

#### C.2.1 Introduction

This Resource specifies the local access control list.

When used without query parameters, all the ACE entries are returned.

When used with a query parameter, only the ACEs matching the specified parameter are returned.

#### C.2.2 Well-known URI

/oic/sec/acl2

#### C.2.3 Resource type

The Resource Type is defined as: "oic.r.acl2".

#### C.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Access Control List-2",
    "version": "2019-01-11",
    "license": {
      "name": "OCF Data Model License",
      "url":
        "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
        CENSE.md",
      "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
        reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
```

```

4665     "/oic/sec/acl2" : {
4666         "get": {
4667             "description": "This Resource specifies the local access control list.\nWhen used without
4668 query parameters, all the ACE entries are returned.\nWhen used with a query parameter, only the ACEs
4669 matching the specified\nparameter are returned.\n",
4670             "parameters": [
4671                 {"$ref": "#/parameters/interface"},
4672                 {"$ref": "#/parameters/ace-filtered"}
4673             ],
4674             "responses": {
4675                 "200": {
4676                     "description": "",
4677                     "x-example":
4678                     {
4679                         "rt" : ["oic.r.acl2"],
4680                         "aclist2": [
4681                             {
4682                                 "aceid": 1,
4683                                 "subject": {
4684                                     "authority": "484b8a51-cb23-46c0-a5f1-b4aebef50ebe",
4685                                     "role": "SOME_STRING"
4686                                 },
4687                                 "resources": [
4688                                     {
4689                                         "href": "/light"
4690                                     },
4691                                     {
4692                                         "href": "/door"
4693                                     }
4694                                 ],
4695                                 "permission": 24
4696                             },
4697                             {
4698                                 "aceid": 2,
4699                                 "subject": {
4700                                     "uuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
4701                                 },
4702                                 "resources": [
4703                                     {
4704                                         "href": "/light"
4705                                     },
4706                                     {
4707                                         "href": "/door"
4708                                     }
4709                                 ],
4710                                 "permission": 24
4711                             },
4712                             {
4713                                 "aceid": 3,
4714                                 "subject": {"conntype": "anon-clear"},
4715                                 "resources": [
4716                                     {
4717                                         "href": "/light"
4718                                     },
4719                                     {
4720                                         "href": "/door"
4721                                     }
4722                                 ],
4723                                 "permission": 16,
4724                                 "validity": [
4725                                     {
4726                                         "period": "20160101T180000Z/20170102T070000Z",
4727                                         "recurrence": [ "DSTART:XXXXX",
4728 "RRULE:FREQ=DAILY;UNTIL=20180131T140000Z;BYMONTH=1" ]
4729                                     },
4730                                     {
4731                                         "period": "20160101T180000Z/PT5H30M",
4732                                         "recurrence": [ "RRULE:FREQ=DAILY;UNTIL=20180131T140000Z;BYMONTH=1" ]
4733                                     }
4734                                 ]
4735                             }
4736                         ]
4737                     }
4738                 }
4739             }
4740         }
4741     }

```

```

4737         "rowneruuid": "de305d54-75b4-431b-adb2-eb6b9e546014"
4738     },
4739     "schema": { "$ref": "#/definitions/Acl2" }
4740 },
4741 "400": {
4742     "description": "The request is invalid."
4743 }
4744 },
4745 },
4746 "post": {
4747     "description": "Updates the ACL Resource with the provided ACEs.\n\nACEs provided in the
4748 update with aceids not currently in the ACL\nResource are added.\n\nACEs provided in the update with
4749 aceid(s) already in the ACL completely\nreplace the ACE(s) in the ACL Resource.\n\nACEs provided in
4750 the update without aceid properties are added and\nassigned unique aceids in the ACL Resource.\n",
4751     "parameters": [
4752         { "$ref": "#/parameters/interface" },
4753         { "$ref": "#/parameters/ace-filtered" },
4754     ],
4755     "name": "body",
4756     "in": "body",
4757     "required": true,
4758     "schema": { "$ref": "#/definitions/Acl2-Update" },
4759     "x-example":
4760     {
4761         "aclist2": [
4762             {
4763                 "aceid": 1,
4764                 "subject": {
4765                     "authority": "484b8a51-cb23-46c0-a5f1-b4aebef50ebe",
4766                     "role": "SOME_STRING"
4767                 },
4768                 "resources": [
4769                     {
4770                         "href": "/light"
4771                     },
4772                     {
4773                         "href": "/door"
4774                     }
4775                 ],
4776                 "permission": 24
4777             },
4778             {
4779                 "aceid": 3,
4780                 "subject": {
4781                     "uuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
4782                 },
4783                 "resources": [
4784                     {
4785                         "href": "/light"
4786                     },
4787                     {
4788                         "href": "/door"
4789                     }
4790                 ],
4791                 "permission": 24
4792             }
4793         ],
4794         "rowneruuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
4795     }
4796 },
4797 ],
4798 "responses": {
4799     "400": {
4800         "description": "The request is invalid."
4801     },
4802     "201": {
4803         "description": "The ACL entry is created."
4804     },
4805     "204": {
4806         "description": "The ACL entry is updated."
4807     }
4808 }

```

```

4809     },
4810     "delete": {
4811         "description": "Deletes ACL entries.\nWhen DELETE is used without query parameters, all the
4812 ACE entries are deleted.\nWhen DELETE is used with a query parameter, only the ACEs matching
4813 the\nspecified parameter are deleted.\n",
4814         "parameters": [
4815             {"$ref": "#/parameters/interface"},
4816             {"$ref": "#/parameters/ace-filtered"}
4817         ],
4818         "responses": {
4819             "200": {
4820                 "description": "The matching ACEs or the entire ACL Resource has been successfully
4821 deleted."
4822             },
4823             "400": {
4824                 "description": "The request is invalid."
4825             }
4826         }
4827     }
4828 },
4829 },
4830 "parameters": {
4831     "interface": {
4832         "in": "query",
4833         "name": "if",
4834         "type": "string",
4835         "enum": [ "oic.if.baseline", "oic.if.rw" ]
4836     },
4837     "ace-filtered": {
4838         "in": "query",
4839         "name": "aceid",
4840         "required": false,
4841         "type": "integer",
4842         "description": "Only applies to the ACE with the specified aceid.",
4843         "x-example": 2112
4844     }
4845 },
4846 "definitions": {
4847     "Acl2": {
4848         "properties": {
4849             "owneruuid": {
4850                 "description": "The value identifies the unique Resource owner\nFormat pattern according
4851 to IETF RFC 4122.",
4852                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
4853 9]{12}$",
4854                 "type": "string"
4855             },
4856             "rt": {
4857                 "description": "Resource Type of the Resource.",
4858                 "items": {
4859                     "maxLength": 64,
4860                     "type": "string",
4861                     "enum": [ "oic.r.acl2" ]
4862                 },
4863                 "minItems": 1,
4864                 "readOnly": true,
4865                 "type": "array"
4866             },
4867             "aclist2": {
4868                 "description": "Access Control Entries in the ACL Resource.",
4869                 "items": {
4870                     "properties": {
4871                         "aceid": {
4872                             "description": "An identifier for the ACE that is unique within the ACL. In cases
4873 where it isn't supplied in an update, the Server will add the ACE and assign it a unique value.",
4874                             "minimum": 1,
4875                             "type": "integer"
4876                         },
4877                         "permission": {
4878                             "description": "Bitmask encoding of CRUDN permission\nThe encoded bitmask indicating
4879 permissions.",
4880                             "x-detail-desc": [

```

```

4881         "0 - No permissions",
4882         "1 - Create permission is granted",
4883         "2 - Read, observe, discover permission is granted",
4884         "4 - Write, update permission is granted",
4885         "8 - Delete permission is granted",
4886         "16 - Notify permission is granted"
4887     ],
4888     "maximum": 31,
4889     "minimum": 0,
4890     "type": "integer"
4891 },
4892 "resources": {
4893     "description": "References the application's Resources to which a security policy
4894 applies.",
4895     "items": {
4896         "description": "Each Resource must have at least one of these properties set.",
4897         "properties": {
4898             "href": {
4899                 "description": "When present, the ACE only applies when the href matches\nThis
4900 is the target URI, it can be specified as a Relative Reference or fully-qualified URI.",
4901                 "format": "uri",
4902                 "maxLength": 256,
4903                 "type": "string"
4904             },
4905             "wc": {
4906                 "description": "A wildcard matching policy.",
4907                 "pattern": "^[~+]*$",
4908                 "type": "string"
4909             }
4910         },
4911         "type": "object"
4912     },
4913     "type": "array"
4914 },
4915 "subject": {
4916     "anyOf": [
4917         {
4918             "description": "This is the Device identifier.",
4919             "properties": {
4920                 "uuid": {
4921                     "description": "A UUID Device ID\nFormat pattern according to IETF RFC
4922 4122.",
4923                     "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-
4924 fA-F0-9]{12}$",
4925                     "type": "string"
4926                 }
4927             },
4928             "required": [
4929                 "uuid"
4930             ],
4931             "type": "object"
4932         },
4933         {
4934             "description": "Security role specified as an <Authority> & <Rolename>. A NULL
4935 <Authority> refers to the local entity or Device.",
4936             "properties": {
4937                 "authority": {
4938                     "description": "The Authority component of the entity being identified. A
4939 NULL <Authority> refers to the local entity or Device.",
4940                     "type": "string"
4941                 },
4942                 "role": {
4943                     "description": "The ID of the role being identified.",
4944                     "type": "string"
4945                 }
4946             },
4947             "required": [
4948                 "role"
4949             ],
4950             "type": "object"
4951         }
4952     ]

```

```

4953         "properties": {
4954             "conntype": {
4955                 "description": "This property allows an ACE to be matched based on the
4956 connection or message type.",
4957                 "x-detail-desc": [
4958                     "auth-crypt - ACE applies if the Client is authenticated and the data
4959 channel or message is encrypted and integrity protected",
4960                     "anon-clear - ACE applies if the Client is not authenticated and the data
4961 channel or message is not encrypted but may be integrity protected"
4962                 ],
4963                 "enum": [
4964                     "auth-crypt",
4965                     "anon-clear"
4966                 ],
4967                 "type": "string"
4968             }
4969         },
4970         "required": [
4971             "conntype"
4972         ],
4973         "type": "object"
4974     }
4975 ]
4976 },
4977 "validity": {
4978     "description": "validity is an array of time-pattern objects.",
4979     "items": {
4980         "description": "The time-pattern contains a period and recurrence expressed in
4981 RFC5545 syntax.",
4982         "properties": {
4983             "period": {
4984                 "description": "String represents a period using the RFC5545 Period.",
4985                 "type": "string"
4986             },
4987             "recurrence": {
4988                 "description": "String array represents a recurrence rule using the RFC5545
4989 Recurrence.",
4990                 "items": {
4991                     "type": "string"
4992                 },
4993                 "type": "array"
4994             }
4995         },
4996         "required": [
4997             "period"
4998         ],
4999         "type": "object"
5000     },
5001     "type": "array"
5002 }
5003 },
5004 "required": [
5005     "aceid",
5006     "resources",
5007     "permission",
5008     "subject"
5009 ],
5010 "type": "object"
5011 },
5012 "type": "array"
5013 },
5014 "n": {
5015     "$ref":
5016 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5017 schema.json#/definitions/n"
5018 },
5019 "id": {
5020     "$ref":
5021 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5022 schema.json#/definitions/id"
5023 },
5024 "if" : {

```

```

5025         "description": "The interface set supported by this Resource.",
5026         "items": {
5027             "enum": [ "oic.if.baseline", "oic.if.rw" ],
5028             "type": "string"
5029         },
5030         "minItems": 1,
5031         "readOnly": true,
5032         "type": "array"
5033     },
5034 },
5035 "type" : "object",
5036 "required": ["acllist2", "rowneruuid"]
5037 },
5038 "Acl2-Update" : {
5039     "properties": {
5040         "rowneruuid" : {
5041             "description": "The value identifies the unique Resource owner\n Format pattern according
5042 to IETF RFC 4122.",
5043             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
5044 9]{12}$",
5045             "type": "string"
5046         },
5047         "acllist2" : {
5048             "description": "Access Control Entries in the ACL Resource.",
5049             "items": {
5050                 "properties": {
5051                     "aceid": {
5052                         "description": "An identifier for the ACE that is unique within the ACL. In cases
5053 where it isn't supplied in an update, the Server will add the ACE and assign it a unique value.",
5054                         "minimum": 1,
5055                         "type": "integer"
5056                     },
5057                     "permission": {
5058                         "description": "Bitmask encoding of CRUDN permission\nThe encoded bitmask indicating
5059 permissions.",
5060                         "x-detail-desc": [
5061                             "0 - No permissions",
5062                             "1 - Create permission is granted",
5063                             "2 - Read, observe, discover permission is granted",
5064                             "4 - Write, update permission is granted",
5065                             "8 - Delete permission is granted",
5066                             "16 - Notify permission is granted"
5067                         ],
5068                         "maximum": 31,
5069                         "minimum": 0,
5070                         "type": "integer"
5071                     },
5072                     "resources": {
5073                         "description": "References the application's Resources to which a security policy
5074 applies.",
5075                         "items": {
5076                             "description": "Each Resource must have at least one of these properties set.",
5077                             "properties": {
5078                                 "href": {
5079                                     "description": "When present, the ACE only applies when the href matches\nThis
5080 is the target URI, it can be specified as a Relative Reference or fully-qualified URI.",
5081                                     "format": "uri",
5082                                     "maxLength": 256,
5083                                     "type": "string"
5084                                 },
5085                                 "wc": {
5086                                     "description": "A wildcard matching policy.",
5087                                     "x-detail-desc": [
5088                                         "+ - Matches all discoverable Resources",
5089                                         "- - Matches all non-discoverable Resources",
5090                                         "*" - Matches all Resources"
5091                                     ],
5092                                     "enum": [
5093                                         "+",
5094                                         "-",
5095                                         "*"
5096                                     ],

```



```

5097         "type": "string"
5098     },
5099 },
5100     "type": "object"
5101 },
5102     "type": "array"
5103 },
5104     "subject": {
5105         "anyOf": [
5106             {
5107                 "description": "This is the Device identifier.",
5108                 "properties": {
5109                     "uuid": {
5110                         "description": "A UUID Device ID\n Format pattern according to IETF RFC
5111 4122.",
5112                         "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-
5113 fA-F0-9]{12}$",
5114                         "type": "string"
5115                     }
5116                 },
5117                 "required": [
5118                     "uuid"
5119                 ],
5120                 "type": "object"
5121             },
5122             {
5123                 "description": "Security role specified as an <Authority> & <Rolename>. A NULL
5124 <Authority> refers to the local entity or Device.",
5125                 "properties": {
5126                     "authority": {
5127                         "description": "The Authority component of the entity being identified. A
5128 NULL <Authority> refers to the local entity or Device.",
5129                         "type": "string"
5130                     },
5131                     "role": {
5132                         "description": "The ID of the role being identified.",
5133                         "type": "string"
5134                     }
5135                 },
5136                 "required": [
5137                     "role"
5138                 ],
5139                 "type": "object"
5140             },
5141             {
5142                 "properties": {
5143                     "conntype": {
5144                         "description": "This property allows an ACE to be matched based on the
5145 connection or message type.",
5146                         "x-detail-desc": [
5147                             "auth-crypt - ACE applies if the Client is authenticated and the data
5148 channel or message is encrypted and integrity protected",
5149                             "anon-clear - ACE applies if the Client is not authenticated and the data
5150 channel or message is not encrypted but may be integrity protected"
5151                         ],
5152                         "enum": [
5153                             "auth-crypt",
5154                             "anon-clear"
5155                         ],
5156                         "type": "string"
5157                     }
5158                 },
5159                 "required": [
5160                     "conntype"
5161                 ],
5162                 "type": "object"
5163             }
5164         ]
5165     },
5166     "validity": {
5167         "description": "validity is an array of time-pattern objects.",
5168         "items": {

```

```

5169         "description": "The time-pattern contains a period and recurrence expressed in
5170 RFC5545 syntax.",
5171         "properties": {
5172             "period": {
5173                 "description": "String represents a period using the RFC5545 Period.",
5174                 "type": "string"
5175             },
5176             "recurrence": {
5177                 "description": "String array represents a recurrence rule using the RFC5545
5178 Recurrence.",
5179                 "items": {
5180                     "type": "string"
5181                 },
5182                 "type": "array"
5183             }
5184         },
5185         "required": [
5186             "period"
5187         ],
5188         "type": "object"
5189     },
5190     "type": "array"
5191 }
5192 },
5193 "required": [
5194     "resources",
5195     "permission",
5196     "subject"
5197 ],
5198 "type": "object"
5199 },
5200 "type": "array"
5201 }
5202 },
5203 "type" : "object"
5204 }
5205 }
5206 }
5207

```

## 5208 C.2.5 Property definition

5209 Table C-1 defines the Properties that are part of the "oic.r.acl2" Resource Type.

5210 **Table C-1 – The Property definitions of the Resource with type "rt" = "oic.r.acl2".**

| Property name | Value type                 | Mandatory | Access mode | Description  |
|---------------|----------------------------|-----------|-------------|--|
| rowneruuid    | string                     | Yes       | Read Write  | The value identifies the unique Resource owner<br>Format pattern according to IETF RFC 4122. |
| rt            | array: see schema          | No        | Read Only   | Resource Type of the Resource.   |
| aclist2       | array: see schema          | Yes       | Read Write  | Access Control Entries in the ACL Resource.  |
| n             | multiple types: see schema | No        | Read Write  |  |
| id            | multiple types: see schema | No        | Read Write  |  |
| if            | array: see schema          | No        | Read Only   | The interface set supported by this Resource.  |

|            |                   |    |            |  |
|------------|-------------------|----|------------|--|
| rowneruuid | string            | No | Read Write | The value identifies the unique Resource owner<br>Format pattern according to IETF RFC 4122. |
| aclist2    | array: see schema | No | Read Write | Access Control Entries in the ACL Resource.  |

## C.2.6 CRUDN behaviour

Table C-2 defines the CRUDN operations that are supported on the "oic.r.acl2" Resource Type.

**Table C-2 – The CRUDN operations of the Resource with type "rt" = "oic.r.acl2".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   | delete | observe |

## C.3 Credential

### C.3.1 Introduction

This Resource specifies credentials a Device may use to establish secure communication.

Retrieves the credential data.

When used without query parameters, all the credential entries are returned.

When used with a query parameter, only the credentials matching the specified parameter are returned.

Note that write-only credential data will not be returned.

### C.3.2 Well-known URI

/oic/sec/cred

### C.3.3 Resource type

The Resource Type is defined as: "oic.r.cred".

### C.3.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Credential",
    "version": "2020-10-19",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "copyright 2016-2017, 2019, 2020 Open Connectivity Foundation, Inc. All rights
reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/sec/cred": {
      "get": {
        "description": "This Resource specifies credentials a Device may use to establish secure
communication.\nRetrieves the credential data.\nWhen used without query parameters, all the
credential entries are returned.\nWhen used with a query parameter, only the credentials matching
```

```

5253 the specified\nparameter are returned.\n\nNote that write-only credential data will not be
5254 returned.\n",
5255     "parameters": [
5256         {"$ref": "#/parameters/interface"},
5257         {"$ref": "#/parameters/cred-filtered-credid"},
5258         {"$ref": "#/parameters/cred-filtered-subjectuuid"}
5259     ],
5260     "responses": {
5261         "200": {
5262             "description": "",
5263             "x-example": {
5264                 "rt": ["oic.r.cred"],
5265                 "creds": [
5266                     {
5267                         "credid": 55,
5268                         "subjectuuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
5269                         "roleid": {
5270                             "authority": "484b8a51-cb23-46c0-a5f1-b4aebef50ebe",
5271                             "role": "SOME_STRING"
5272                         },
5273                         "credtype": 32,
5274                         "publicdata": {
5275                             "encoding": "oic.sec.encoding.pem",
5276                             "data": "PEM-ENCODED-VALUE"
5277                         },
5278                         "privatedata": {
5279                             "encoding": "oic.sec.encoding.raw",
5280                             "data": "RAW-ENCODED-VALUE",
5281                             "handle": 4
5282                         },
5283                         "optionaldata": {
5284                             "revstat": false,
5285                             "encoding": "oic.sec.encoding.pem",
5286                             "data": "PEM-ENCODED-VALUE"
5287                         },
5288                         "period": "20160101T180000Z/20170102T070000Z",
5289                         "crms": [ "oic.sec.crm.pk10" ]
5290                     },
5291                     {
5292                         "credid": 56,
5293                         "subjectuuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
5294                         "roleid": {
5295                             "authority": "484b8a51-cb23-46c0-a5f1-b4aebef50ebe",
5296                             "role": "SOME_STRING"
5297                         },
5298                         "credtype": 1,
5299                         "publicdata": {
5300                             "encoding": "oic.sec.encoding.pem",
5301                             "data": "PEM-ENCODED-VALUE"
5302                         },
5303                         "privatedata": {
5304                             "encoding": "oic.sec.encoding.base64",
5305                             "data": "BASE-64-ENCODED-VALUE",
5306                             "handle": 4
5307                         },
5308                         "optionaldata": {
5309                             "revstat": false,
5310                             "encoding": "oic.sec.encoding.pem",
5311                             "data": "PEM-ENCODED-VALUE"
5312                         },
5313                         "period": "20160101T180000Z/20170102T070000Z",
5314                         "crms": [ "oic.sec.crm.pk10" ]
5315                     }
5316                 ],
5317                 "rowneruuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
5318             },
5319             "schema": { "$ref": "#/definitions/Cred" }
5320         },
5321         "400": {
5322             "description": "The request is invalid."
5323         }
5324     }

```

```

5325     }
5326   },
5327   "post": {
5328     "description": "Updates the credential Resource with the provided
5329 credentials.\n\nCredentials provided in the update with credid(s) not currently in the\ncredential
5330 Resource are added.\n\nCredentials provided in the update with credid(s) already in the\ncredential
5331 Resource completely replace the creds in the credential\nResource.\n\nCredentials provided in the
5332 update without credid(s) properties are\nadded and assigned unique credid(s) in the credential
5333 Resource.\n",
5334     "parameters": [
5335       { "$ref": "#/parameters/interface" },
5336       {
5337         "name": "body",
5338         "in": "body",
5339         "required": true,
5340         "schema": { "$ref": "#/definitions/Cred-Update" },
5341         "x-example":
5342           {
5343             "creds": [
5344               {
5345                 "credid": 55,
5346                 "subjectuuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
5347                 "roleid": {
5348                   "authority": "484b8a51-cb23-46c0-a5f1-b4aebef50ebe",
5349                   "role": "SOME_STRING"
5350                 },
5351                 "credtype": 32,
5352                 "publicdata": {
5353                   "encoding": "oic.sec.encoding.pem",
5354                   "data": "PEM-ENCODED-VALUE"
5355                 },
5356                 "privatedata": {
5357                   "encoding": "oic.sec.encoding.raw",
5358                   "data": "RAW-ENCODED-VALUE",
5359                   "handle": 4
5360                 },
5361                 "optionaldata": {
5362                   "revstat": false,
5363                   "encoding": "oic.sec.encoding.pem",
5364                   "data": "PEM-ENCODED-VALUE"
5365                 },
5366                 "period": "20160101T180000Z/20170102T070000Z",
5367                 "crms": [ "oic.sec.crm.pk10" ]
5368               },
5369               {
5370                 "credid": 56,
5371                 "subjectuuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
5372                 "roleid": {
5373                   "authority": "484b8a51-cb23-46c0-a5f1-b4aebef50ebe",
5374                   "role": "SOME_STRING"
5375                 },
5376                 "credtype": 1,
5377                 "publicdata": {
5378                   "encoding": "oic.sec.encoding.pem",
5379                   "data": "PEM-ENCODED-VALUE"
5380                 },
5381                 "privatedata": {
5382                   "encoding": "oic.sec.encoding.base64",
5383                   "data": "BASE-64-ENCODED-VALUE",
5384                   "handle": 4
5385                 },
5386                 "optionaldata": {
5387                   "revstat": false,
5388                   "encoding": "oic.sec.encoding.pem",
5389                   "data": "PEM-ENCODED-VALUE"
5390                 },
5391                 "period": "20160101T180000Z/20170102T070000Z",
5392                 "crms": [ "oic.sec.crm.pk10" ]
5393               }
5394             ],
5395             "rowneruuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
5396           }
5397       }
5398     ]
5399   }

```

```

5397     }
5398   ],
5399   "responses": {
5400     "400": {
5401       "description": "The request is invalid."
5402     },
5403     "201": {
5404       "description": "The credential entry is created."
5405     },
5406     "204": {
5407       "description": "The credential entry is updated."
5408     }
5409   }
5410 },
5411 "delete": {
5412   "description": "Deletes credential entries.\nWhen DELETE is used without query parameters,
5413 all the cred entries are deleted.\nWhen DELETE is used with a query parameter, only the entries
5414 matching\nthe query parameter are deleted.\n",
5415   "parameters": [
5416     {"$ref": "#/parameters/interface"},
5417     {"$ref": "#/parameters/cred-filtered-credid"},
5418     {"$ref": "#/parameters/cred-filtered-subjectuuid"}
5419   ],
5420   "responses": {
5421     "400": {
5422       "description": "The request is invalid."
5423     },
5424     "204": {
5425       "description": "The specific credential(s) or the the entire credential Resource has
5426 been successfully deleted."
5427     }
5428   }
5429 }
5430 },
5431 },
5432 "parameters": {
5433   "interface": {
5434     "in": "query",
5435     "name": "if",
5436     "type": "string",
5437     "enum": ["oic.if.baseline", "oic.if.rw"]
5438   },
5439   "cred-filtered-credid": {
5440     "in": "query",
5441     "name": "credid",
5442     "required": false,
5443     "type": "integer",
5444     "description": "Only applies to the credential with the specified credid.",
5445     "x-example": 2112
5446   },
5447   "cred-filtered-subjectuuid": {
5448     "in": "query",
5449     "name": "subjectuuid",
5450     "required": false,
5451     "type": "string",
5452     "description": "Only applies to credentials with the specified subject UUID.",
5453     "x-example": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
5454   }
5455 },
5456 "definitions": {
5457   "Cred": {
5458     "properties": {
5459       "rowneruuid": {
5460         "description": "Format pattern according to IETF RFC 4122.",
5461         "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$",
5462         "type": "string"
5463       },
5464       "rt": {
5465         "description": "Resource Type of the Resource.",
5466         "items": {
5467           "maxLength": 64,

```

```

5469         "type": "string",
5470         "enum": ["oic.r.cred"]
5471     },
5472     "minItems": 1,
5473     "readOnly": true,
5474     "type": "array",
5475     "uniqueItems": true
5476 },
5477 "n": {
5478     "$ref":
5479 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5480 schema.json#/definitions/n"
5481 },
5482 "id": {
5483     "$ref":
5484 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
5485 schema.json#/definitions/id"
5486 },
5487 "creds": {
5488     "description": "List of credentials available at this Resource.",
5489     "items": {
5490         "properties": {
5491             "credid": {
5492                 "description": "Local reference to a credential Resource.",
5493                 "type": "integer"
5494             },
5495             "credtype": {
5496                 "description": "Representation of this credential's type\nCredential Types - Cred
5497 type encoded as a bitmask.0 - Empty credential used for testing\n1 - Symmetric pair-wise key\n2 -
5498 Symmetric group key\n4 - Asymmetric signing key\n8 - Asymmetric signing key with certificate\n16 -
5499 PIN or password\n32 - Asymmetric encryption key. \n128 - SSM Client\n256 - SSM Server",
5500                 "maximum": 256,
5501                 "minimum": 0,
5502                 "type": "integer"
5503             },
5504             "credusage": {
5505                 "description": "A string that provides hints about how/where the cred is used\nThe
5506 type of credusage.oic.sec.cred.trustca - Trust certificateoic.sec.cred.cert -
5507 Certificateoic.sec.cred.rolecert - Role Certificateoic.sec.cred.mfgtrustca - Manufacturer
5508 Certificate Trust Anchoroic.sec.cred.mfgcert - Manufacturer Certificate.",
5509                 "enum": [
5510                     "oic.sec.cred.trustca",
5511                     "oic.sec.cred.cert",
5512                     "oic.sec.cred.rolecert",
5513                     "oic.sec.cred.mfgtrustca",
5514                     "oic.sec.cred.mfgcert"
5515                 ],
5516                 "type": "string"
5517             },
5518             "crms": {
5519                 "description": "The refresh methods that may be used to update this credential.",
5520                 "items": {
5521                     "description": "Each enum represents a method by which the credentials are
5522 refreshed.oic.sec.crm.pro - Credentials refreshed by a provisioning serviceoic.sec.crm.rdp -
5523 Credentials refreshed by a key agreement protocol and random PINoic.sec.crm.psk - Credentials
5524 refreshed by a key agreement protocoloic.sec.crm.skdc - Credentials refreshed by a key distribution
5525 serviceoic.sec.crm.pk10 - Credentials refreshed by a PKCS#10 request to a CA.",
5526                     "enum": [
5527                         "oic.sec.crm.pro",
5528                         "oic.sec.crm.psk",
5529                         "oic.sec.crm.rdp",
5530                         "oic.sec.crm.skdc",
5531                         "oic.sec.crm.pk10"
5532                     ],
5533                     "type": "string"
5534                 },
5535                 "type": "array",
5536                 "uniqueItems": true
5537             },
5538             "optionaldata": {
5539                 "description": "Credential Type dependent. Credential revocation status
5540 information\n1, 2, 4, 32, 64: revocation status information\n8: Revocation information",

```

```

5541         "properties": {
5542             "data": {
5543                 "description": "The encoded structure.",
5544                 "type": "string"
5545             },
5546             "encoding": {
5547                 "description": "A string specifying the encoding format of the data contained in
5548 the optdata.",
5549                 "x-detail-desc": [
5550                     "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain."
5551                 ],
5552                 "enum": [
5553                     "oic.sec.encoding.pem"
5554                 ],
5555                 "type": "string"
5556             },
5557             "revstat": {
5558                 "description": "Revocation status flag - true = revoked.",
5559                 "type": "boolean"
5560             }
5561         },
5562         "required": [
5563             "revstat"
5564         ],
5565         "type": "object"
5566     },
5567     "period": {
5568         "description": "String with RFC5545 Period.",
5569         "type": "string"
5570     },
5571     "privatedata": {
5572         "description": "Private credential information\nnCredencal Resource non-public
5573 contents.",
5574         "properties": {
5575             "data": {
5576                 "description": "The encoded value.",
5577                 "maxLength": 3072,
5578                 "type": "string"
5579             },
5580             "encoding": {
5581                 "description": "A string specifying the encoding format of the data contained in
5582 the privdata.",
5583                 "x-detail-desc": [
5584                     "oic.sec.encoding.pem - Encoding for PEM encoded private key.",
5585                     "oic.sec.encoding.base64 - Encoding for Base64 encoded PSK.",
5586                     "oic.sec.encoding.handle - Data is contained in a storage sub-system
5587 referenced using a handle.",
5588                     "oic.sec.encoding.raw - Raw hex encoded data."
5589                 ],
5590                 "enum": [
5591                     "oic.sec.encoding.pem",
5592                     "oic.sec.encoding.base64",
5593                     "oic.sec.encoding.handle",
5594                     "oic.sec.encoding.raw"
5595                 ],
5596                 "type": "string"
5597             },
5598             "handle": {
5599                 "description": "Handle to a key storage Resource.",
5600                 "type": "integer"
5601             }
5602         },
5603         "required": [
5604             "encoding"
5605         ],
5606         "type": "object"
5607     },
5608     "publicdata": {
5609         "description": "Credential Type dependent. Public credential information\nn1:2:
5610 ticket, public SKDC values\nn4, 32: Public key value\nn8: A chain of one or more certificate",
5611         "properties": {
5612             "data": {

```



```

5613         "description": "The encoded value.",
5614         "maxLength": 3072,
5615         "type": "string"
5616     },
5617     "encoding": {
5618         "description": "A string specifying the encoding format of the data contained in
the pubdata.",
5619         "x-detail-desc": [
5620             "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain."
5621         ],
5622         "enum": [
5623             "oic.sec.encoding.pem"
5624         ],
5625         "type": "string"
5626     },
5627 },
5628 },
5629 "type": "object"
5630 },
5631 "oscore": {
5632     "description": "Contains parameters for use with credentials intended for use with
5633 OSCORE. See type definition for \"oic.sec.oscoretype\",
5634     "properties": {
5635         "senderid": {
5636             "description": "OSCORE Sender ID for this OSCORE Security Context",
5637             "type": "string"
5638         },
5639         "recipientid": {
5640             "description": "OSCORE Recipient ID for this OSCORE Security Context",
5641             "type": "string"
5642         },
5643         "ssn": {
5644             "description": "OSCORE Sender Sequence Number SSN1 being stored in nonvolatile
5645 memory to handle the loss of mutable security context parameters",
5646             "type": "integer",
5647             "readOnly": true
5648         },
5649         "desc": {
5650             "description": "Human readable description of the usage of this OSCORE Security
5651 Context",
5652             "type": "string"
5653         }
5654     },
5655     "type": "object"
5656 },
5657 "roleid": {
5658     "description": "The role this credential possesses\nSecurity role specified as an
5659 <Authority> & <RoleName>. A NULL <Authority> refers to the local entity or Device.",
5660     "properties": {
5661         "authority": {
5662             "description": "The Authority component of the entity being identified. A NULL
5663 <Authority> refers to the local entity or Device.",
5664             "type": "string"
5665         },
5666         "role": {
5667             "description": "The ID of the role being identified.",
5668             "type": "string"
5669         }
5670     },
5671     "required": [
5672         "role"
5673     ],
5674     "type": "object"
5675 },
5676 "subjectuuid": {
5677     "anyOf": [
5678         {
5679             "description": "The id of the Device, which the cred entry applies to or \"*\n
5680 for wildcard identity.",
5681             "pattern": "^[\\*]$",
5682             "type": "string"
5683         },
5684     ]

```

```

5685         "description": "Format pattern according to IETF RFC 4122.",
5686         "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-
5687 F0-9]{12}$",
5688         "type": "string"
5689     }
5690 ]
5691 }
5692 },
5693 "type": "object"
5694 },
5695 "type": "array"
5696 },
5697 "if": {
5698     "description": "The interface set supported by this Resource.",
5699     "items": {
5700         "enum": [
5701             "oic.if.baseline",
5702             "oic.if.rw"
5703         ],
5704         "type": "string"
5705     },
5706     "minItems": 2,
5707     "readOnly": true,
5708     "type": "array"
5709 }
5710 },
5711 "type": "object",
5712 "required": ["creds", "rowneruuid"]
5713 },
5714 "Cred-Update": {
5715     "properties": {
5716         "rowneruuid": {
5717             "description": "Format pattern according to IETF RFC 4122.",
5718             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
5719 9]{12}$",
5720             "type": "string"
5721         },
5722         "creds": {
5723             "description": "List of credentials available at this Resource.",
5724             "items": {
5725                 "properties": {
5726                     "credid": {
5727                         "description": "Local reference to a credential Resource.",
5728                         "type": "integer"
5729                     },
5730                     "credtype": {
5731                         "description": "Representation of this credential's type\nCredential Types - Cred
5732 type encoded as a bitmask.0 - Empty credential used for testing\n1 - Symmetric pair-wise key\n2 -
5733 Symmetric group key\n4 - Asymmetric signing key\n8 - Asymmetric signing key with certificate\n16 -
5734 PIN or password\n32 - Asymmetric encryption key. \n 128 - SSM Client\n256 - SSM Server",
5735                         "maximum": 256,
5736                         "minimum": 0,
5737                         "type": "integer"
5738                     },
5739                     "credusage": {
5740                         "description": "A string that provides hints about how/where the cred is used\nThe
5741 type of credusage.oic.sec.cred.trustca - Trust certificateoic.sec.cred.cert -
5742 Certificateoic.sec.cred.rolecert - Role Certificateoic.sec.cred.mfgtrustca - Manufacturer
5743 Certificate Trust Anchoroic.sec.cred.mfgcert - Manufacturer Certificate.",
5744                         "enum": [
5745                             "oic.sec.cred.trustca",
5746                             "oic.sec.cred.cert",
5747                             "oic.sec.cred.rolecert",
5748                             "oic.sec.cred.mfgtrustca",
5749                             "oic.sec.cred.mfgcert"
5750                         ],
5751                         "type": "string"
5752                     },
5753                     "crms": {
5754                         "description": "The refresh methods that may be used to update this credential.",
5755                         "items": {
5756                             "description": "Each enum represents a method by which the credentials are

```

```

5757 refreshed.oic.sec.crm.pro - Credentials refreshed by a provisioning serviceoic.sec.crm.rdp -
5758 Credentials refreshed by a key agreement protocol and random PINoic.sec.crm.psk - Credentials
5759 refreshed by a key agreement protocoloic.sec.crm.skdc - Credentials refreshed by a key distribution
5760 serviceoic.sec.crm.pk10 - Credentials refreshed by a PKCS#10 request to a CA.",
5761         "enum": [
5762             "oic.sec.crm.pro",
5763             "oic.sec.crm.psk",
5764             "oic.sec.crm.rdp",
5765             "oic.sec.crm.skdc",
5766             "oic.sec.crm.pk10"
5767         ],
5768         "type": "string"
5769     },
5770     "type": "array"
5771 },
5772     "optionaldata": {
5773         "description": "Credential Type dependent. Credential revocation status
5774 information\n1, 2, 4, 32, 64: revocation status information\n8: Revocation information",
5775         "properties": {
5776             "data": {
5777                 "description": "The encoded structure.",
5778                 "type": "string"
5779             },
5780             "encoding": {
5781                 "description": "A string specifying the encoding format of the data contained in
5782 the optdata.",
5783                 "x-detail-desc": [
5784                     "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain."
5785                 ],
5786                 "enum": [
5787                     "oic.sec.encoding.pem"
5788                 ],
5789                 "type": "string"
5790             },
5791             "revstat": {
5792                 "description": "Revocation status flag - true = revoked.",
5793                 "type": "boolean"
5794             }
5795         },
5796         "required": [
5797             "revstat"
5798         ],
5799         "type": "object"
5800     },
5801     "period": {
5802         "description": "String with RFC5545 Period.",
5803         "type": "string"
5804     },
5805     "privatedata": {
5806         "description": "Private credential information\nCredential Resource non-public
5807 contents.",
5808         "properties": {
5809             "data": {
5810                 "description": "The encoded value.",
5811                 "maxLength": 3072,
5812                 "type": "string"
5813             },
5814             "encoding": {
5815                 "description": "A string specifying the encoding format of the data contained in
5816 the privdata.",
5817                 "x-detail-desc": [
5818                     "oic.sec.encoding.pem - Encoding for PEM encoded private key.",
5819                     "oic.sec.encoding.base64 - Encoding for Base64 encoded PSK.",
5820                     "oic.sec.encoding.handle - Data is contained in a storage sub-system
5821 referenced using a handle.",
5822                     "oic.sec.encoding.raw - Raw hex encoded data."
5823                 ],
5824                 "enum": [
5825                     "oic.sec.encoding.pem",
5826                     "oic.sec.encoding.base64",
5827                     "oic.sec.encoding.handle",
5828                     "oic.sec.encoding.raw"

```

```

5829         ],
5830         "type": "string"
5831     },
5832     "handle": {
5833         "description": "Handle to a key storage Resource.",
5834         "type": "integer"
5835     }
5836 },
5837 "required": [
5838     "encoding"
5839 ],
5840 "type": "object"
5841 },
5842 "publicdata": {
5843     "description": "Credential Type dependent. Public credential information\n1:2:
5844 ticket, public SKDC values\n4, 32: Public key value\n8: A chain of one or more certificate",
5845     "properties": {
5846         "data": {
5847             "description": "The encoded value.",
5848             "maxLength": 3072,
5849             "type": "string"
5850         },
5851         "encoding": {
5852             "description": "A string specifying the encoding format of the data contained in
5853 the pubdata.",
5854             "x-detail-desc": [
5855                 "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain."
5856             ],
5857             "enum": [
5858                 "oic.sec.encoding.pem"
5859             ],
5860             "type": "string"
5861         }
5862     },
5863     "type": "object"
5864 },
5865 "oscore": {
5866     "description": "Contains parameters for use with credentials intended for use with
5867 OSCORE. See type definition for \"oic.sec.oscoretype\"",
5868     "properties": {
5869         "senderid": {
5870             "description": "OSCORE Sender ID for this OSCORE Security Context",
5871             "type": "string"
5872         },
5873         "recipientid": {
5874             "description": "OSCORE Recipient ID for this OSCORE Security Context",
5875             "type": "string"
5876         },
5877         "desc": {
5878             "description": "Human readable description of the usage of this OSCORE Security
5879 Context",
5880             "type": "string"
5881         }
5882     },
5883     "type": "object"
5884 },
5885 "roleid": {
5886     "description": "The role this credential possesses\nSecurity role specified as an
5887 <Authority> & <Rolename>. A NULL <Authority> refers to the local entity or Device.",
5888     "properties": {
5889         "authority": {
5890             "description": "The Authority component of the entity being identified. A NULL
5891 <Authority> refers to the local entity or Device.",
5892             "type": "string"
5893         },
5894         "role": {
5895             "description": "The ID of the role being identified.",
5896             "type": "string"
5897         }
5898     },
5899     "required": [
5900         "role"

```

```

5901         ],
5902         "type": "object"
5903     },
5904     "subjectuuid": {
5905         "anyOf": [
5906             {
5907                 "description": "The id of the Device, which the cred entry applies to or \"*\
5908 for wildcard identity.",
5909                 "pattern": "^\\*$",
5910                 "type": "string"
5911             },
5912             {
5913                 "description": "Format pattern according to IETF RFC 4122.",
5914                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-
5915 F0-9]{12}$",
5916                 "type": "string"
5917             }
5918         ]
5919     },
5920 },
5921 "type": "object"
5922 },
5923 "type": "array"
5924 },
5925 "if": {
5926     "description": "The interface set supported by this Resource.",
5927     "items": {
5928         "enum": [
5929             "oic.if.baseline"
5930         ],
5931         "type": "string"
5932     },
5933     "minItems": 1,
5934     "readOnly": true,
5935     "type": "array"
5936 },
5937 },
5938 "type": "object"
5939 }
5940 }
5941 }
5942

```

### 5943 C.3.5 Property definition

5944 Table C-3 defines the Properties that are part of the "oic.r.cred" Resource Type.

5945 **Table C-3 – The Property definitions of the Resource with type "rt" = "oic.r.cred".**

| Property name | Value type                 | Mandatory | Access mode | Description                                     |
|---------------|----------------------------|-----------|-------------|---|
| rowneruuid    | string                     | Yes       | Read Write  | Format pattern according to IETF RFC 4122.      |
| rt            | array: see schema          | No        | Read Only   | Resource Type of the Resource.                  |
| n             | multiple types: see schema | No        | Read Write  |   |
| id            | multiple types: see schema | No        | Read Write  |   |
| creds         | array: see schema          | Yes       | Read Write  | List of credentials available at this Resource. |
| if            | array: see schema          | No        | Read Only   | The interface set supported by this Resource.   |

|            |                   |    |            |   |
|------------|-------------------|----|------------|---|
| rowneruuid | string            | No | Read Write | Format pattern according to IETF RFC 4122.      |
| creds      | array: see schema | No | Read Write | List of credentials available at this Resource. |
| if         | array: see schema | No | Read Only  | The interface set supported by this Resource.   |

### C.3.6 CRUDN behaviour

Table C-4 defines the CRUDN operations that are supported on the "oic.r.cred" Resource Type.

**Table C-4 – The CRUDN operations of the Resource with type "rt" = "oic.r.cred".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   | delete | observe |

## C.4 Certificate Signing Request

### C.4.1 Introduction

This Resource specifies a Certificate Signing Request.

### C.4.2 Well-known URI

/oic/sec/csr

### C.4.3 Resource type

The Resource Type is defined as: "oic.r.csr".

### C.4.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Certificate Signing Request",
    "version": "2015-08-19",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/sec/csr" : {
      "get": {
        "description": "This Resource specifies a Certificate Signing Request.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example":
{
  "rt": ["oic.r.csr"],
```

```

5989         "encoding" : "oic.sec.encoding.pem",
5990         "csr" : "PEMENCODEDCSR"
5991     },
5992     "schema": { "$ref": "#/definitions/Csr" }
5993 },
5994 "404": {
5995     "description" : "The Device does not support certificates and generating CSRs."
5996 },
5997 "503": {
5998     "description" : "The Device is not yet ready to return a response. Try again later."
5999 }
6000 }
6001 }
6002 }
6003 },
6004 "parameters": {
6005     "interface" : {
6006         "in" : "query",
6007         "name" : "if",
6008         "type" : "string",
6009         "enum" : [ "oic.if.baseline", "oic.if.rw" ]
6010     }
6011 },
6012 "definitions": {
6013     "Csr" : {
6014         "properties": {
6015             "rt" : {
6016                 "description": "Resource Type of the Resource.",
6017                 "items": {
6018                     "maxLength": 64,
6019                     "type": "string",
6020                     "enum": [ "oic.r.csr" ]
6021                 },
6022                 "minItems": 1,
6023                 "readOnly": true,
6024                 "type": "array"
6025             },
6026             "encoding": {
6027                 "description": "A string specifying the encoding format of the data contained in CSR.",
6028                 "x-detail-desc": [
6029                     "oic.sec.encoding.pem - Encoding for PEM encoded CSR."
6030                 ],
6031                 "enum": [
6032                     "oic.sec.encoding.pem"
6033                 ],
6034                 "readOnly": true,
6035                 "type": "string"
6036             },
6037             "n": {
6038                 "$ref":
6039 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6040 schema.json#/definitions/n"
6041             },
6042             "id": {
6043                 "$ref":
6044 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6045 schema.json#/definitions/id"
6046             },
6047             "csr": {
6048                 "description": "Signed CSR in ASN.1 in the encoding specified by the encoding property.",
6049                 "maxLength": 3072,
6050                 "readOnly": true,
6051                 "type": "string"
6052             },
6053             "if": {
6054                 "description": "The interface set supported by this Resource.",
6055                 "items": {
6056                     "enum": [ "oic.if.baseline", "oic.if.rw" ],
6057                     "type": "string"
6058                 },
6059                 "minItems": 1,
6060                 "readOnly": true,

```

```

6061         "type": "array"
6062     },
6063 },
6064     "type": "object",
6065     "required": ["csr", "encoding"]
6066 }
6067 }
6068 }
6069

```

#### 6070 C.4.5 Property definition

6071 Table C-5 defines the Properties that are part of the "oic.r.csr" Resource Type.

6072 **Table C-5 – The Property definitions of the Resource with type "rt" = "oic.r.csr".**

| Property name | Value type                 | Mandatory | Access mode | Description   |
|---------------|----------------------------|-----------|-------------|---|
| rt            | array: see schema          | No        | Read Only   | Resource Type of the Resource.  |
| encoding      | string                     | Yes       | Read Only   | A string specifying the encoding format of the data contained in CSR.   |
| n             | multiple types: see schema | No        | Read Write  |   |
| id            | multiple types: see schema | No        | Read Write  |   |
| csr           | string                     | Yes       | Read Only   | Signed CSR in ASN.1 in the encoding specified by the encoding property. |
| if            | array: see schema          | No        | Read Only   | The interface set supported by this Resource.                           |

#### 6073 C.4.6 CRUDN behaviour

6074 Table C-6 defines the CRUDN operations that are supported on the "oic.r.csr" Resource Type.

6075 **Table C-6 – The CRUDN operations of the Resource with type "rt" = "oic.r.csr".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  |        |        | observe |

### 6076 C.5 Device Owner Transfer Method

#### 6077 C.5.1 Introduction

6078 This Resource specifies properties needed to establish a Device owner.

6079

#### 6080 C.5.2 Well-known URI

6081 /oic/sec/doxm

#### 6082 C.5.3 Resource type

6083 The Resource Type is defined as: "oic.r.doxm".

#### 6084 C.5.4 OpenAPI 2.0 definition

```

6085 {
6086     "swagger": "2.0",

```



```

6087     "info": {
6088         "title": "Device Owner Transfer Method",
6089         "version": "2020-10-19",
6090         "license": {
6091             "name": "OCF Data Model License",
6092             "url":
6093 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
6094 CENSE.md",
6095         "x-copyright": "copyright 2016-2017, 2019, 2020 Open Connectivity Foundation, Inc. All rights
6096 reserved."
6097     },
6098     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6099 },
6100 "schemes": ["http"],
6101 "consumes": ["application/json"],
6102 "produces": ["application/json"],
6103 "paths": {
6104     "/oic/sec/doxm" : {
6105         "get": {
6106             "description": "This Resource specifies properties needed to establish a Device owner.\n",
6107             "parameters": [
6108                 {"$ref": "#/parameters/interface"},
6109                 {"$ref": "#/parameters/owned"}
6110             ],
6111             "responses": {
6112                 "200": {
6113                     "description": "",
6114                     "x-example": {
6115                         "rt": ["oic.r.doxm"],
6116                         "oxms": [ 0, 2, 3 ],
6117                         "oxmsel": 0,
6118                         "sct": 16,
6119                         "owned": true,
6120                         "deviceuuid": "de305d54-75b4-431b-adb2-eb6b9e546014",
6121                         "devowneruuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
6122                         "rowneruuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
6123                     },
6124                     "schema": { "$ref": "#/definitions/Doxm" }
6125                 },
6126                 "400": {
6127                     "description": "The request is invalid."
6128                 }
6129             }
6130         },
6131         "post": {
6132             "description": "Updates the DOXM Resource data.\n",
6133             "parameters": [
6134                 {"$ref": "#/parameters/interface"},
6135                 {
6136                     "name": "body",
6137                     "in": "body",
6138                     "required": true,
6139                     "schema": { "$ref": "#/definitions/Doxm-Update" },
6140                     "x-example":
6141                     {
6142                         "oxmsel": 0,
6143                         "owned": true,
6144                         "deviceuuid": "de305d54-75b4-431b-adb2-eb6b9e546014",
6145                         "devowneruuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
6146                         "rowneruuid": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
6147                     }
6148                 }
6149             ],
6150             "responses": {
6151                 "400": {
6152                     "description": "The request is invalid."
6153                 },
6154                 "204": {
6155                     "description": "The DOXM entry is updated."
6156                 }
6157             }
6158         }
6159     }
6160 }

```

```

6159     }
6160   },
6161   "parameters": {
6162     "interface": {
6163       "in": "query",
6164       "name": "if",
6165       "type": "string",
6166       "enum": [ "oic.if.baseline", "oic.if.rw" ]
6167     },
6168     "owned": {
6169       "in": "query",
6170       "name": "owned",
6171       "type": "boolean"
6172     }
6173   },
6174   "definitions": {
6175     "Doxm": {
6176       "properties": {
6177         "rowneruuid": {
6178           "description": "Format pattern according to IETF RFC 4122.",
6179           "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$",
6180           "type": "string"
6181         },
6182         "oxms": {
6183           "description": "List of supported owner transfer methods.",
6184           "items": {
6185             "description": "The Device owner transfer methods that may be selected at Device on-boarding. Each value indicates a specific Owner Transfer method0 - Numeric OTM identifier for the Just-Works method (oic.sec.doxm.jw)1 - Numeric OTM identifier for the random PIN method (oic.sec.doxm.rdp)2 - Numeric OTM identifier for the manufacturer certificate method (oic.sec.doxm.mfgcert)3 - Numeric OTM identifier for the decap method (oic.sec.doxm.dcap) (deprecated).",
6186             "type": "integer"
6187           },
6188           "readOnly": true,
6189           "type": "array"
6190         },
6191         "devowneruuid": {
6192           "description": "Format pattern according to IETF RFC 4122.",
6193           "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$",
6194           "type": "string"
6195         },
6196         "deviceuuid": {
6197           "description": "The uuid formatted identity of the Device\nFormat pattern according to IETF RFC 4122.",
6198           "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$",
6199           "type": "string"
6200         },
6201         "owned": {
6202           "description": "Ownership status flag.",
6203           "type": "boolean"
6204         },
6205         "n": {
6206           "$ref": "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-schema.json#/definitions/n"
6207         },
6208         "id": {
6209           "$ref": "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-schema.json#/definitions/id"
6210         },
6211         "oxmsel": {
6212           "description": "The selected owner transfer method used during on-boarding\nThe Device owner transfer methods that may be selected at Device on-boarding. Each value indicates a specific Owner Transfer method0 - Numeric OTM identifier for the Just-Works method (oic.sec.doxm.jw)1 - Numeric OTM identifier for the random PIN method (oic.sec.doxm.rdp)2 - Numeric OTM identifier for the manufacturer certificate method (oic.sec.doxm.mfgcert)3 - Numeric OTM identifier for the decap method (oic.sec.doxm.dcap) (deprecated).",
6213           "type": "integer"
6214         }
6215       }
6216     }
6217   }
6218 }
6219

```

```

6231         "type": "integer"
6232     },
6233     "sct": {
6234         "description": "Bitmask encoding of supported credential types\nCredential Types -
6235 Cred type encoded as a bitmask.0 - Empty credential used for testing1 - Symmetric pair-wise key2 -
6236 Symmetric group key4 - Asymmetric signing key8 - Asymmetric signing key with certificate16 - PIN or
6237 password32 - Asymmetric encryption key.",
6238         "maximum": 511,
6239         "minimum": 0,
6240         "type": "integer",
6241         "readOnly": true
6242     },
6243     "rt": {
6244         "description": "Resource Type of the Resource.",
6245         "items": {
6246             "maxLength": 64,
6247             "type": "string",
6248             "enum": ["oic.r.doxm"]
6249         },
6250         "minItems": 1,
6251         "readOnly": true,
6252         "type": "array"
6253     },
6254     "if": {
6255         "description": "The OCF Interface set supported by this Resource.",
6256         "items": {
6257             "enum": [
6258                 "oic.if.baseline",
6259                 "oic.if.rw"
6260             ],
6261             "type": "string"
6262         },
6263         "minItems": 2,
6264         "readOnly": true,
6265         "type": "array"
6266     }
6267 },
6268 "type" : "object",
6269 "required": ["oxms", "oxmsel", "sct", "owned", "deviceuuid", "devowneruuid", "rowneruuid"]
6270 },
6271 "Doxm-Update" : {
6272     "properties": {
6273         "rowneruuid": {
6274             "description": "Format pattern according to IETF RFC 4122.",
6275             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
6276 9]{12}$",
6277             "type": "string"
6278         },
6279         "devowneruuid": {
6280             "description": "Format pattern according to IETF RFC 4122.",
6281             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
6282 9]{12}$",
6283             "type": "string"
6284         },
6285         "deviceuuid": {
6286             "description": "The uuid formatted identity of the Device\nFormat pattern according to
6287 IETF RFC 4122.",
6288             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
6289 9]{12}$",
6290             "type": "string"
6291         },
6292         "owned": {
6293             "description": "Ownership status flag.",
6294             "type": "boolean"
6295         },
6296         "oxmsel": {
6297             "description": "The selected owner transfer method used during on-boarding\nThe Device
6298 owner transfer methods that may be selected at Device on-boarding. Each value indicates a specific
6299 Owner Transfer method0 - Numeric OTM identifier for the Just-Works method (oic.sec.doxm.jw)1 -
6300 Numeric OTM identifier for the random PIN method (oic.sec.doxm.rdp)2 - Numeric OTM identifier for
6301 the manufacturer certificate method (oic.sec.doxm.mfgcert)3 - Numeric OTM identifier for the decap
6302 method (oic.sec.doxm.dcap) (deprecated).",

```

```

6303         "type": "integer"
6304     },
6305 },
6306     "type" : "object"
6307 }
6308 }
6309 }
6310

```

### C.5.5 Property definition

Table C-7 defines the Properties that are part of the "oic.r.doxm" Resource Type.

**Table C-7 – The Property definitions of the Resource with type "rt" = "oic.r.doxm".**

| Property name | Value type                 | Mandatory | Access mode | Description   |
|---------------|----------------------------|-----------|-------------|---|
| rowneruuid    | string                     | Yes       | Read Write  | Format pattern according to IETF RFC 4122.  |
| oxms          | array: see schema          | Yes       | Read Only   | List of supported owner transfer methods.   |
| devowneruuid  | string                     | Yes       | Read Write  | Format pattern according to IETF RFC 4122.  |
| deviceuuid    | string                     | Yes       | Read Write  | The uuid formatted identity of the Device<br>Format pattern according to IETF RFC 4122.   |
| owned         | boolean                    | Yes       | Read Write  | Ownership status flag.  |
| n             | multiple types: see schema | No        | Read Write  |   |
| id            | multiple types: see schema | No        | Read Write  |   |
| oxmsel        | integer                    | Yes       | Read Write  | The selected owner transfer method used during on-boarding<br>The Device owner transfer methods that may be selected at Device on-boarding.<br>Each value indicates a specific Owner Transfer method<br>method0 - Numeric OTM identifier for the Just-Works method (oic.sec.doxm.jw)<br>1 - Numeric OTM identifier for the random PIN method (oic.sec.doxm.rdp)<br>2 - Numeric OTM identifier for the manufacturer certificate method (oic.sec.doxm.mfgcert)<br>3 - Numeric OTM identifier for the decap method (oic.sec.doxm.dcap) (deprecated). |
| sct           | integer                    | Yes       | Read Only   | Bitmask encoding of supported credential types<br>Credential Types - Cred   |

|              |                   |    |            |   |
|--------------|-------------------|----|------------|---|
|              |                   |    |            | type encoded as a bitmask.0 - Empty credential used for testing1 - Symmetric pair-wise key2 - Symmetric group key4 - Asymmetric signing key8 - Asymmetric signing key with certificate16 - PIN or password32 - Asymmetric encryption key.   |
| rt           | array: see schema | No | Read Only  | Resource Type of the Resource.  |
| if           | array: see schema | No | Read Only  | The OCF Interface set supported by this Resource.   |
| rowneruuid   | string            |    | Read Write | Format pattern according to IETF RFC 4122.  |
| devowneruuid | string            |    | Read Write | Format pattern according to IETF RFC 4122.  |
| deviceuuid   | string            |    | Read Write | The uuid formatted identity of the Device<br>Format pattern according to IETF RFC 4122.   |
| owned        | boolean           |    | Read Write | Ownership status flag.  |
| oxmsel       | integer           |    | Read Write | The selected owner transfer method used during on-boarding<br>The Device owner transfer methods that may be selected at Device on-boarding.<br>Each value indicates a specific Owner Transfer method<br>0 - Numeric OTM identifier for the Just-Works method (oic.sec.doxm.jw)<br>1 - Numeric OTM identifier for the random PIN method (oic.sec.doxm.rdp)<br>2 - Numeric OTM identifier for the manufacturer certificate method (oic.sec.doxm.mfgcert)<br>3 - Numeric OTM identifier for the decap method (oic.sec.doxm.dcap) (deprecated). |

### C.5.6 CRUDN behaviour

Table C-8 defines the CRUDN operations that are supported on the "oic.r.doxm" Resource Type.

6316 **Table C-8 – The CRUDN operations of the Resource with type "rt" = "oic.r.doxm".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   |        | observe |

## 6317 **C.6 Device Provisioning Status**

### 6318 **C.6.1 Introduction**

6319 This Resource specifies Device provisioning status.

6320

### 6321 **C.6.2 Well-known URI**

6322 /oic/sec/pstat

### 6323 **C.6.3 Resource type**

6324 The Resource Type is defined as: "oic.r.pstat".

### 6325 **C.6.4 OpenAPI 2.0 definition**

```

6326 {
6327   "swagger": "2.0",
6328   "info": {
6329     "title": "Device Provisioning Status",
6330     "version": "2019-10-01",
6331     "license": {
6332       "name": "OCF Data Model License",
6333       "url":
6334         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
6335         CENSE.md",
6336       "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
6337         reserved."
6338     },
6339     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
6340   },
6341   "schemes": ["http"],
6342   "consumes": ["application/json"],
6343   "produces": ["application/json"],
6344   "paths": {
6345     "/oic/sec/pstat" : {
6346       "get": {
6347         "description": "This Resource specifies Device provisioning status.\n",
6348         "parameters": [
6349           {"$ref": "#/parameters/interface"}
6350         ],
6351         "responses": {
6352           "200": {
6353             "description": "",
6354             "x-example":
6355               {
6356                 "rt": ["oic.r.pstat"],
6357                 "dos": {"s": 3, "p": true},
6358                 "isop": true,
6359                 "cm": 8,
6360                 "tm": 60,
6361                 "om": 2,
6362                 "sm": 7,
6363                 "rowneruuid": "de305d54-75b4-431b-adb2-eb6b9e546014"
6364               },
6365             "schema": { "$ref": "#/definitions/Pstat" }
6366           },
6367           "400": {
6368             "description": "The request is invalid."
6369           }
6370         }
6371       },
6372       "post": {

```

```

6373     "description": "Sets or updates Device provisioning status data.\n",
6374     "parameters": [
6375         { "$ref": "#/parameters/interface" },
6376         {
6377             "name": "body",
6378             "in": "body",
6379             "required": true,
6380             "schema": { "$ref": "#/definitions/Pstat-Update" },
6381             "x-example":
6382                 {
6383                     "dos": { "s": 3 },
6384                     "tm": 60,
6385                     "om": 2,
6386                     "rowneruuid": "de305d54-75b4-431b-adb2-eb6b9e546014"
6387                 }
6388         },
6389     ],
6390     "responses": {
6391         "400": {
6392             "description": "The request is invalid."
6393         },
6394         "204": {
6395             "description": "The PSTAT entry is updated."
6396         }
6397     }
6398 },
6399 },
6400 },
6401 "parameters": {
6402     "interface": {
6403         "in": "query",
6404         "name": "if",
6405         "type": "string",
6406         "enum": [ "oic.if.baseline", "oic.if.rw" ]
6407     }
6408 },
6409 "definitions": {
6410     "Pstat": {
6411         "properties": {
6412             "rowneruuid": {
6413                 "description": "The UUID formatted identity of the Resource owner\nFormat pattern
6414 according to IETF RFC 4122.",
6415                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
6416 9]{12}$",
6417                 "type": "string"
6418             },
6419             "rt": {
6420                 "description": "Resource Type of the Resource.",
6421                 "items": {
6422                     "maxLength": 64,
6423                     "type": "string",
6424                     "enum": [ "oic.r.pstat" ]
6425                 },
6426                 "minItems": 1,
6427                 "readOnly": true,
6428                 "type": "array"
6429             },
6430             "om": {
6431                 "description": "Current operational mode\nDevice provisioning operation may be server
6432 directed or client (aka provisioning service) directed. The value is a bitmask encoded as integer
6433 and indicates the provisioning operation modes1 - Server-directed utilizing multiple provisioning
6434 services2 - Server-directed utilizing a single provisioning service4 - Client-directed provisioning8
6435 - Unused16 - Unused32 - Unused64 - Unused128 - Unused.",
6436                 "maximum": 7,
6437                 "minimum": 1,
6438                 "type": "integer"
6439             },
6440             "cm": {
6441                 "description": "Current Device provisioning mode\nDevice provisioning mode maintains a
6442 bitmask of the possible provisioning states of a Device. The value can be either 8 or 16 character
6443 in length. If its only 8 characters it represents the lower byte value1 - Manufacturer reset state2 -
6444 Device pairing and owner transfer state4 - Unused8 - Provisioning of credential management

```

```

6445 services16 - Provisioning of access management services32 - Provisioning of local ACLs64 - Initiate
6446 Software Version Validation128 - Initiate Secure Software Update.",
6447     "maximum": 255,
6448     "minimum": 0,
6449     "type": "integer",
6450     "readOnly": true
6451 },
6452 "n": {
6453     "$ref":
6454 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6455 schema.json#/definitions/n"
6456 },
6457 "id": {
6458     "$ref":
6459 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6460 schema.json#/definitions/id"
6461 },
6462 "isop": {
6463     "description": "true indicates Device is operational.",
6464     "readOnly": true,
6465     "type": "boolean"
6466 },
6467 "tm": {
6468     "description": "Target Device provisioning mode\nDevice provisioning mode maintains a
6469 bitmask of the possible provisioning states of a Device. The value can be either 8 or 16 character
6470 in length. If its only 8 characters it represents the lower byte value1 - Manufacturer reset state2
6471 - Device pairing and owner transfer state4 - Unused8 - Provisioning of credential management
6472 services16 - Provisioning of access management services32 - Provisioning of local ACLs64 - Initiate
6473 Software Version Validation128 - Initiate Secure Software Update.",
6474     "maximum": 255,
6475     "minimum": 0,
6476     "type": "integer"
6477 },
6478 "sm": {
6479     "description": "Supported operational modes\nDevice provisioning operation may be server
6480 directed or client (aka provisioning service) directed. The value is a bitmask encoded as integer
6481 and indicates the provisioning operation modes1 - Server-directed utilizing multiple provisioning
6482 services2 - Server-directed utilizing a single provisioning service4 - Client-directed provisioning8
6483 - Unused16 - Unused32 - Unused64 - Unused128 - Unused.",
6484     "maximum": 7,
6485     "minimum": 1,
6486     "type": "integer",
6487     "readOnly": true
6488 },
6489 "dos": {
6490     "description": "Device on-boarding state\nDevice operation state machine.",
6491     "properties": {
6492         "p": {
6493             "default": true,
6494             "description": "'p' is TRUE when the 's' state is pending until all necessary changes
6495 to Device Resources are complete.",
6496             "readOnly": true,
6497             "type": "boolean"
6498         },
6499         "s": {
6500             "description": "The current or pending operational state.",
6501             "x-detail-desc": [
6502                 "0 - RESET - Device reset state.",
6503                 "1 - RFOTM - Ready for Device owner transfer method state.",
6504                 "2 - RFPRO - Ready for Device provisioning state.",
6505                 "3 - RFNOP - Ready for Device normal operation state.",
6506                 "4 - SRESET - The Device is in a soft reset state."
6507             ],
6508             "maximum": 4,
6509             "minimum": 0,
6510             "type": "integer"
6511         }
6512     },
6513     "required": [
6514         "s"
6515     ],
6516     "type": "object"

```



```

6517     },
6518     "if" : {
6519         "description": "The interface set supported by this Resource.",
6520         "items": {
6521             "enum": [ "oic.if.baseline", "oic.if.rw" ],
6522             "type": "string"
6523         },
6524         "minItems": 1,
6525         "readOnly": true,
6526         "type": "array"
6527     },
6528 },
6529 "type" : "object",
6530 "required": [ "dos", "isop", "cm", "tm", "om", "sm", "owneruuid" ]
6531 },
6532 "Pstat-Update" : {
6533     "properties": {
6534         "owneruuid": {
6535             "description": "The UUID formatted identity of the Resource owner\nFormat pattern
6536 according to IETF RFC 4122.",
6537             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
6538 9]{12}$",
6539             "type": "string"
6540         },
6541         "om": {
6542             "description": "Current operational mode\nDevice provisioning operation may be server
6543 directed or client (aka provisioning service) directed. The value is a bitmask encoded as integer
6544 and indicates the provisioning operation modes1 - Server-directed utilizing multiple provisioning
6545 services2 - Server-directed utilizing a single provisioning service4 - Client-directed provisioning8
6546 - Unused16 - Unused32 - Unused64 - Unused128 - Unused.",
6547             "maximum": 7,
6548             "minimum": 1,
6549             "type": "integer"
6550         },
6551         "tm": {
6552             "description": "Target Device provisioning mode\nDevice provisioning mode maintains a
6553 bitmask of the possible provisioning states of a Device. The value can be either 8 or 16 character
6554 in length. If its only 8 characters it represents the lower byte value1 - Manufacturer reset state2
6555 - Device pairing and owner transfer state4 - Unused8 - Provisioning of credential management
6556 services16 - Provisioning of access management services32 - Provisioning of local ACLs64 - Initiate
6557 Software Version Validation128 - Initiate Secure Software Update.",
6558             "maximum": 255,
6559             "minimum": 0,
6560             "type": "integer"
6561         },
6562         "dos": {
6563             "description": "Device on-boarding state\nDevice operation state machine.",
6564             "properties": {
6565                 "p": {
6566                     "default": true,
6567                     "description": "'p' is TRUE when the 's' state is pending until all necessary changes
6568 to Device Resources are complete.",
6569                     "readOnly": true,
6570                     "type": "boolean"
6571                 },
6572                 "s": {
6573                     "description": "The current or pending operational state.",
6574                     "x-detail-desc": [
6575                         "0 - RESET - Device reset state.",
6576                         "1 - RFOTM - Ready for Device owner transfer method state.",
6577                         "2 - RFPRO - Ready for Device provisioning state.",
6578                         "3 - RFNOP - Ready for Device normal operation state.",
6579                         "4 - SRESET - The Device is in a soft reset state."
6580                     ],
6581                     "maximum": 4,
6582                     "minimum": 0,
6583                     "type": "integer"
6584                 }
6585             },
6586             "required": [
6587                 "s"
6588             ]
6589         }
6590     }

```

```

6589         "type": "object"
6590     },
6591 },
6592 "type" : "object"
6593 }
6594 }
6595 }
6596

```

## 6597 C.6.5 Property definition

6598 Table C-9 defines the Properties that are part of the "oic.r.pstat" Resource Type.

6599 **Table C-9 – The Property definitions of the Resource with type "rt" = "oic.r.pstat".**

| Property name | Value type        | Mandatory | Access mode | Description   |
|---------------|-------------------|-----------|-------------|---|
| owneruuid     | string            | Yes       | Read Write  | The UUID formatted identity of the Resource owner<br>Format pattern according to IETF RFC 4122.   |
| rt            | array: see schema | No        | Read Only   | Resource Type of the Resource.  |
| om            | integer           | Yes       | Read Write  | Current operational mode<br>Device provisioning operation may be server directed or client (aka provisioning service) directed. The value is a bitmask encoded as integer and indicates the provisioning operation modes1 - Server-directed utilizing multiple provisioning services2 - Server-directed utilizing a single provisioning service4 - Client-directed provisioning8 - Unused16 - Unused32 - Unused64 - Unused128 - Unused. |
| cm            | integer           | Yes       | Read Only   | Current Device provisioning mode<br>Device provisioning mode maintains a bitmask of the possible provisioning states of a Device. The value can be either 8 or 16 character in length. If its only 8 characters it represents the lower byte value1 - Manufacturer reset state2 - Device pairing and owner  |

|      |                            |     |            |   |
|------|----------------------------|-----|------------|---|
|      |                            |     |            | transfer state4 - Unused8 - Provisioning of credential management services16 - Provisioning of access management services32 - Provisioning of local ACLs64 - Initiate Software Version Validation128 - Initiate Secure Software Update.   |
| n    | multiple types: see schema | No  | Read Write |   |
| id   | multiple types: see schema | No  | Read Write |   |
| isop | boolean                    | Yes | Read Only  | true indicates Device is operational.   |
| tm   | integer                    | Yes | Read Write | Target Device provisioning mode<br>Device provisioning mode maintains a bitmask of the possible provisioning states of a Device. The value can be either 8 or 16 character in length. If its only 8 characters it represents the lower byte value1 - Manufacturer reset state2 - Device pairing and owner transfer state4 - Unused8 - Provisioning of credential management services16 - Provisioning of access management services32 - Provisioning of local ACLs64 - Initiate Software Version Validation128 - Initiate Secure Software Update. |
| sm   | integer                    | Yes | Read Only  | Supported operational modes<br>Device provisioning operation may be server directed or client (aka provisioning service) directed. The value is a bitmask encoded as integer and indicates the provisioning operation modes1 - Server-directed utilizing multiple   |

|            |                    |     |            |   |
|------------|--------------------|-----|------------|---|
|            |                    |     |            | provisioning services2 - Server-directed utilizing a single provisioning service4 - Client-directed provisioning8 - Unused16 - Unused32 - Unused64 - Unused128 - Unused.  |
| dos        | object: see schema | Yes | Read Write | Device on-boarding state<br>Device operation state machine.   |
| if         | array: see schema  | No  | Read Only  | The interface set supported by this Resource.   |
| rowneruuid | string             | No  | Read Write | The UUID formatted identity of the Resource owner<br>Format pattern according to IETF RFC 4122.   |
| om         | integer            | No  | Read Write | Current operational mode<br>Device provisioning operation may be server directed or client (aka provisioning service) directed. The value is a bitmask encoded as integer and indicates the provisioning operation modes1 - Server-directed utilizing multiple provisioning services2 - Server-directed utilizing a single provisioning service4 - Client-directed provisioning8 - Unused16 - Unused32 - Unused64 - Unused128 - Unused. |
| tm         | integer            | No  | Read Write | Target Device provisioning mode<br>Device provisioning mode maintains a bitmask of the possible provisioning states of a Device. The value can be either 8 or 16 character in length. If its only 8 characters it represents the lower byte value1 - Manufacturer reset state2 - Device   |

|     |                    |    |            |   |
|-----|--------------------|----|------------|---|
|     |                    |    |            | pairing and owner transfer state4 - Unused8 - Provisioning of credential management services16 - Provisioning of access management services32 - Provisioning of local ACLs64 - Initiate Software Version Validation128 - Initiate Secure Software Update. |
| dos | object: see schema | No | Read Write | Device on-boarding state<br>Device operation state machine.   |

## C.6.6 CRUDN behaviour

Table C-10 defines the CRUDN operations that are supported on the "oic.r.pstat" Resource Type.

**Table C-10 – The CRUDN operations of the Resource with type "rt" = "oic.r.pstat".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   |        | observe |

## C.7 Asserted Roles

### C.7.1 Introduction

This Resource specifies roles that have been asserted.

### C.7.2 Well-known URI

/oic/sec/roles

### C.7.3 Resource type

The Resource Type is defined as: "oic.r.roles".

### C.7.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Asserted Roles",
    "version": "2017-03-23",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/sec/roles" : {
      "get": {
```

```

6633     "description": "This Resource specifies roles that have been asserted.\n",
6634     "parameters": [
6635         { "$ref": "#/parameters/interface" }
6636     ],
6637     "responses": {
6638         "200": {
6639             "description": "",
6640             "x-example":
6641                 {
6642                     "roles": [
6643                         {
6644                             "credid": 1,
6645                             "credtype": 8,
6646                             "subjectuuid": "00000000-0000-0000-0000-000000000000",
6647                             "publicdata":
6648                                 {
6649                                     "encoding": "oic.sec.encoding.pem",
6650                                     "data": "PEMENCODEDROLECERT"
6651                                 },
6652                             "optionaldata":
6653                                 {
6654                                     "revstat": false,
6655                                     "encoding": "oic.sec.encoding.pem",
6656                                     "data": "PEMENCODEDISSUERCERT"
6657                                 }
6658                         },
6659                         {
6660                             "credid": 2,
6661                             "credtype": 8,
6662                             "subjectuuid": "00000000-0000-0000-0000-000000000000",
6663                             "publicdata":
6664                                 {
6665                                     "encoding": "oic.sec.encoding.pem",
6666                                     "data": "PEMENCODEDROLECERT"
6667                                 },
6668                             "optionaldata":
6669                                 {
6670                                     "revstat": false,
6671                                     "encoding": "oic.sec.encoding.pem",
6672                                     "data": "PEMENCODEDISSUERCERT"
6673                                 }
6674                         }
6675                     ],
6676                     "rt": [ "oic.r.roles" ],
6677                     "if": [ "oic.if.rw" ]
6678                 }
6679             ,
6680             "schema": { "$ref": "#/definitions/Roles" }
6681         },
6682         "400": {
6683             "description": "The request is invalid."
6684         }
6685     }
6686 },
6687 "post": {
6688     "description": "Update the roles Resource, i.e., assert new roles to this server.\n\nNew
6689 role certificates that match an existing certificate (i.e., publicdata\nand optionaldata are the
6690 same) are not added to the Resource (and 204 is\nreturned).\n\nThe provided credid values are
6691 ignored, the Resource assigns its own.\n",
6692     "parameters": [
6693         { "$ref": "#/parameters/interface" },
6694         {
6695             "name": "body",
6696             "in": "body",
6697             "required": true,
6698             "schema": { "$ref": "#/definitions/Roles-update" },
6699             "x-example":
6700                 {
6701                     "roles": [
6702                         {
6703                             "credid": 1,
6704                             "credtype": 8,

```

```

6705         "subjectuuid": "00000000-0000-0000-0000-000000000000",
6706         "publicdata":
6707         {
6708             "encoding": "oic.sec.encoding.pem",
6709             "data": "PEMENCODEDROLECERT"
6710         },
6711         "optionaldata":
6712         {
6713             "revstat": false,
6714             "encoding": "oic.sec.encoding.pem",
6715             "data": "PEMENCODEDISSUERCERT"
6716         }
6717     },
6718     {
6719         "credid": 2,
6720         "credtype": 8,
6721         "subjectuuid": "00000000-0000-0000-0000-000000000000",
6722         "publicdata":
6723         {
6724             "encoding": "oic.sec.encoding.pem",
6725             "data": "PEMENCODEDROLECERT"
6726         },
6727         "optionaldata":
6728         {
6729             "revstat": false,
6730             "encoding": "oic.sec.encoding.pem",
6731             "data": "PEMENCODEDISSUERCERT"
6732         }
6733     }
6734 ]
6735 }
6736 }
6737 ],
6738 "responses": {
6739     "400": {
6740         "description": "The request is invalid."
6741     },
6742     "204": {
6743         "description": "The roles entry is updated."
6744     }
6745 },
6746 },
6747 "delete": {
6748     "description": "Deletes roles Resource entries.\nWhen DELETE is used without query
6749 parameters, all the roles entries are deleted.\nWhen DELETE is used with a query parameter, only the
6750 entries matching\nthe query parameter are deleted.\n",
6751     "parameters": [
6752         {"$ref": "#/parameters/interface"},
6753         {"$ref": "#/parameters/roles-filtered"}
6754     ],
6755     "responses": {
6756         "200": {
6757             "description": "The specified or all roles Resource entries have been successfully
6758 deleted."
6759         },
6760         "400": {
6761             "description": "The request is invalid."
6762         }
6763     }
6764 }
6765 }
6766 },
6767 "parameters": {
6768     "interface": {
6769         "in": "query",
6770         "name": "if",
6771         "type": "string",
6772         "enum": [ "oic.if.baseline", "oic.if.rw" ]
6773     },
6774     "roles-filtered": {
6775         "in": "query",
6776         "name": "credid",

```

```

6777         "required" : false,
6778         "type" : "integer",
6779         "description" : "Only applies to the credential with the specified credid.",
6780         "x-example" : 2112
6781     }
6782 },
6783 "definitions": {
6784     "Roles" : {
6785         "properties": {
6786             "rt": {
6787                 "description": "Resource Type of the Resource.",
6788                 "items": {
6789                     "maxLength": 64,
6790                     "type": "string",
6791                     "enum": ["oic.r.roles"]
6792                 },
6793                 "minItems": 1,
6794                 "readOnly": true,
6795                 "type": "array"
6796             },
6797             "n": {
6798                 "$ref":
6799 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6800 schema.json#/definitions/n"
6801             },
6802             "id": {
6803                 "$ref":
6804 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
6805 schema.json#/definitions/id"
6806             },
6807             "roles": {
6808                 "description": "List of role certificates.",
6809                 "items": {
6810                     "properties": {
6811                         "credid": {
6812                             "description": "Local reference to a credential Resource.",
6813                             "type": "integer"
6814                         },
6815                         "credtype": {
6816                             "description": "Representation of this credential's type\nCredential Types - Cred
6817 type encoded as a bitmask.0 - Empty credential used for testing1 - Symmetric pair-wise key2 -
6818 Symmetric group key4 - Asymmetric signing key8 - Asymmetric signing key with certificate16 - PIN or
6819 password32 - Asymmetric encryption key.",
6820                             "maximum": 63,
6821                             "minimum": 0,
6822                             "type": "integer"
6823                         },
6824                         "credusage": {
6825                             "description": "A string that provides hints about how/where the cred is used\nThe
6826 type of credusage.oic.sec.cred.trustca - Trust certificateoic.sec.cred.cert -
6827 Certificateoic.sec.cred.rolecert - Role Certificateoic.sec.cred.mfgtrustca - Manufacturer
6828 Certificate Trust Anchoroic.sec.cred.mfgcert - Manufacturer Certificate.",
6829                             "enum": [
6830                                 "oic.sec.cred.trustca",
6831                                 "oic.sec.cred.cert",
6832                                 "oic.sec.cred.rolecert",
6833                                 "oic.sec.cred.mfgtrustca",
6834                                 "oic.sec.cred.mfgcert"
6835                             ],
6836                             "type": "string"
6837                         },
6838                         "crms": {
6839                             "description": "The refresh methods that may be used to update this credential.",
6840                             "items": {
6841                                 "description": "Each enum represents a method by which the credentials are
6842 refreshed.oic.sec.crm.pro - Credentials refreshed by a provisioning serviceoic.sec.crm.rdp -
6843 Credentials refreshed by a key agreement protocol and random PINoic.sec.crm.psk - Credentials
6844 refreshed by a key agreement protocoloic.sec.crm.skdc - Credentials refreshed by a key distribution
6845 serviceoic.sec.crm.pk10 - Credentials refreshed by a PKCS#10 request to a CA.",
6846                                 "enum": [
6847                                     "oic.sec.crm.pro",
6848                                     "oic.sec.crm.psk",

```



```

6849         "oic.sec.crm.rdp",
6850         "oic.sec.crm.skdc",
6851         "oic.sec.crm.pk10"
6852     ],
6853     "type": "string"
6854 },
6855     "type": "array"
6856 },
6857     "optionaldata": {
6858         "description": "Credential revocation status information\nOptional credential
6859 contents describes revocation status for this credential.",
6860         "properties": {
6861             "data": {
6862                 "description": "This is the encoded structure.",
6863                 "type": "string"
6864             },
6865             "encoding": {
6866                 "description": "A string specifying the encoding format of the data contained in
6867 the optdata.",
6868                 "x-detail-desc": [
6869                     "oic.sec.encoding.jwt - RFC7517 JSON web token (JWT) encoding.",
6870                     "oic.sec.encoding.cwt - RFC CBOR web token (CWT) encoding.",
6871                     "oic.sec.encoding.base64 - Base64 encoded object.",
6872                     "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain.",
6873                     "oic.sec.encoding.der - Encoding for DER encoded certificate.",
6874                     "oic.sec.encoding.raw - Raw hex encoded data."
6875                 ],
6876                 "enum": [
6877                     "oic.sec.encoding.jwt",
6878                     "oic.sec.encoding.cwt",
6879                     "oic.sec.encoding.base64",
6880                     "oic.sec.encoding.pem",
6881                     "oic.sec.encoding.der",
6882                     "oic.sec.encoding.raw"
6883                 ],
6884                 "type": "string"
6885             },
6886             "revstat": {
6887                 "description": "Revocation status flag - true = revoked.",
6888                 "type": "boolean"
6889             }
6890         },
6891         "required": [
6892             "revstat"
6893         ],
6894         "type": "object"
6895     },
6896     "period": {
6897         "description": "String with RFC5545 Period.",
6898         "type": "string"
6899     },
6900     "privatedata": {
6901         "description": "Private credential information\nCredential Resource non-public
6902 contents.",
6903         "properties": {
6904             "data": {
6905                 "description": "The encoded value.",
6906                 "maxLength": 3072,
6907                 "type": "string"
6908             },
6909             "encoding": {
6910                 "description": "A string specifying the encoding format of the data contained in
6911 the privdata.",
6912                 "x-detail-desc": [
6913                     "oic.sec.encoding.jwt - RFC7517 JSON web token (JWT) encoding.",
6914                     "oic.sec.encoding.cwt - RFC CBOR web token (CWT) encoding.",
6915                     "oic.sec.encoding.base64 - Base64 encoded object.",
6916                     "oic.sec.encoding.uri - URI reference.",
6917                     "oic.sec.encoding.handle - Data is contained in a storage sub-system
6918 referenced using a handle.",
6919                     "oic.sec.encoding.raw - Raw hex encoded data."
6920                 ],

```

```

6921         "enum": [
6922             "oic.sec.encoding.jwt",
6923             "oic.sec.encoding.cwt",
6924             "oic.sec.encoding.base64",
6925             "oic.sec.encoding.uri",
6926             "oic.sec.encoding.handle",
6927             "oic.sec.encoding.raw"
6928         ],
6929         "type": "string"
6930     },
6931     "handle": {
6932         "description": "Handle to a key storage Resource.",
6933         "type": "integer"
6934     }
6935 },
6936 "required": [
6937     "encoding"
6938 ],
6939 "type": "object"
6940 },
6941 "publicdata": {
6942     "description": "Public credential information.",
6943     "properties": {
6944         "data": {
6945             "description": "This is the encoded value.",
6946             "maxLength": 3072,
6947             "type": "string"
6948         },
6949         "encoding": {
6950             "description": "A string specifying the encoding format of the data contained in
6951 the pubdata.",
6952             "x-detail-desc": [
6953                 "oic.sec.encoding.jwt - RFC7517 JSON web token (JWT) encoding.",
6954                 "oic.sec.encoding.cwt - RFC CBOR web token (CWT) encoding.",
6955                 "oic.sec.encoding.base64 - Base64 encoded object.",
6956                 "oic.sec.encoding.uri - URI reference.",
6957                 "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain.",
6958                 "oic.sec.encoding.der - Encoding for DER encoded certificate.",
6959                 "oic.sec.encoding.raw - Raw hex encoded data."
6960             ],
6961             "enum": [
6962                 "oic.sec.encoding.jwt",
6963                 "oic.sec.encoding.cwt",
6964                 "oic.sec.encoding.base64",
6965                 "oic.sec.encoding.uri",
6966                 "oic.sec.encoding.pem",
6967                 "oic.sec.encoding.der",
6968                 "oic.sec.encoding.raw"
6969             ],
6970             "type": "string"
6971         }
6972     },
6973     "type": "object"
6974 },
6975 "roleid": {
6976     "description": "The role this credential possesses\nSecurity role specified as an
6977 <Authority> & <Rolename>. A NULL <Authority> refers to the local entity or Device.",
6978     "properties": {
6979         "authority": {
6980             "description": "The Authority component of the entity being identified. A NULL
6981 <Authority> refers to the local entity or Device.",
6982             "type": "string"
6983         },
6984         "role": {
6985             "description": "The ID of the role being identified.",
6986             "type": "string"
6987         }
6988     },
6989     "required": [
6990         "role"
6991     ],
6992     "type": "object"

```

```

6993     },
6994     "subjectuuid": {
6995         "anyOf": [
6996             {
6997                 "description": "The id of the Device, which the cred entry applies to or \"*\
6998 for wildcard identity.",
6999                 "pattern": "^\\*$",
7000                 "type": "string"
7001             },
7002             {
7003                 "description": "Format pattern according to IETF RFC 4122.",
7004                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-
7005 F0-9]{12}$",
7006                 "type": "string"
7007             }
7008         ]
7009     },
7010 },
7011 "type": "object"
7012 },
7013 "type": "array"
7014 },
7015 "if": {
7016     "description": "The interface set supported by this Resource.",
7017     "items": {
7018         "enum": [ "oic.if.baseline", "oic.if.rw" ],
7019         "type": "string"
7020     },
7021     "minItems": 1,
7022     "readOnly": true,
7023     "type": "array"
7024 }
7025 },
7026 "type": "object",
7027 "required": [ "roles" ]
7028 },
7029 "Roles-update" : {
7030     "properties": {
7031         "roles": {
7032             "description": "List of role certificates.",
7033             "items": {
7034                 "properties": {
7035                     "credid": {
7036                         "description": "Local reference to a credential Resource.",
7037                         "type": "integer"
7038                     },
7039                     "credtype": {
7040                         "description": "Representation of this credential's type\nCredential Types - Cred
7041 type encoded as a bitmask.0 - Empty credential used for testing1 - Symmetric pair-wise key2 -
7042 Symmetric group key4 - Asymmetric signing key8 - Asymmetric signing key with certificatel6 - PIN or
7043 password32 - Asymmetric encryption key.",
7044                         "maximum": 63,
7045                         "minimum": 0,
7046                         "type": "integer"
7047                     },
7048                     "credusage": {
7049                         "description": "A string that provides hints about how/where the cred is used\nThe
7050 type of credusage.oic.sec.cred.trustca - Trust certificateoic.sec.cred.cert -
7051 Certificateoic.sec.cred.rolecert - Role Certificateoic.sec.cred.mfgtrustca - Manufacturer
7052 Certificate Trust Anchoroic.sec.cred.mfgcert - Manufacturer Certificate.",
7053                         "enum": [
7054                             "oic.sec.cred.trustca",
7055                             "oic.sec.cred.cert",
7056                             "oic.sec.cred.rolecert",
7057                             "oic.sec.cred.mfgtrustca",
7058                             "oic.sec.cred.mfgcert"
7059                         ],
7060                         "type": "string"
7061                     },
7062                     "crms": {
7063                         "description": "The refresh methods that may be used to update this credential.",
7064                         "items": {

```

```

7065         "description": "Each enum represents a method by which the credentials are
7066 refreshed.oic.sec.crm.pro - Credentials refreshed by a provisioning serviceoic.sec.crm.rdp -
7067 Credentials refreshed by a key agreement protocol and random PINoic.sec.crm.psk - Credentials
7068 refreshed by a key agreement protocol.oic.sec.crm.skdc - Credentials refreshed by a key distribution
7069 serviceoic.sec.crm.pk10 - Credentials refreshed by a PKCS#10 request to a CA.",
7070         "enum": [
7071             "oic.sec.crm.pro",
7072             "oic.sec.crm.psk",
7073             "oic.sec.crm.rdp",
7074             "oic.sec.crm.skdc",
7075             "oic.sec.crm.pk10"
7076         ],
7077         "type": "string"
7078     },
7079     "type": "array"
7080 },
7081 "optionaldata": {
7082     "description": "Credential revocation status information\nOptional credential
7083 contents describes revocation status for this credential.",
7084     "properties": {
7085         "data": {
7086             "description": "This is the encoded structure.",
7087             "type": "string"
7088         },
7089         "encoding": {
7090             "description": "A string specifying the encoding format of the data contained in
7091 the optdata.",
7092             "x-detail-desc": [
7093                 "oic.sec.encoding.jwt - RFC7517 JSON web token (JWT) encoding.",
7094                 "oic.sec.encoding.cwt - RFC CBOR web token (CWT) encoding.",
7095                 "oic.sec.encoding.base64 - Base64 encoded object.",
7096                 "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain.",
7097                 "oic.sec.encoding.der - Encoding for DER encoded certificate.",
7098                 "oic.sec.encoding.raw - Raw hex encoded data."
7099             ],
7100             "enum": [
7101                 "oic.sec.encoding.jwt",
7102                 "oic.sec.encoding.cwt",
7103                 "oic.sec.encoding.base64",
7104                 "oic.sec.encoding.pem",
7105                 "oic.sec.encoding.der",
7106                 "oic.sec.encoding.raw"
7107             ],
7108             "type": "string"
7109         },
7110         "revstat": {
7111             "description": "Revocation status flag - true = revoked.",
7112             "type": "boolean"
7113         }
7114     },
7115     "required": [
7116         "revstat"
7117     ],
7118     "type": "object"
7119 },
7120 "period": {
7121     "description": "String with RFC5545 Period.",
7122     "type": "string"
7123 },
7124 "privatedata": {
7125     "description": "Private credential information\nCredential Resource non-public
7126 contents.",
7127     "properties": {
7128         "data": {
7129             "description": "The encoded value.",
7130             "maxLength": 3072,
7131             "type": "string"
7132         },
7133         "encoding": {
7134             "description": "A string specifying the encoding format of the data contained in
7135 the privdata.",
7136             "x-detail-desc": [

```

```

7137         "oic.sec.encoding.jwt - RFC7517 JSON web token (JWT) encoding.",
7138         "oic.sec.encoding.cwt - RFC CBOR web token (CWT) encoding.",
7139         "oic.sec.encoding.base64 - Base64 encoded object.",
7140         "oic.sec.encoding.uri - URI reference.",
7141         "oic.sec.encoding.handle - Data is contained in a storage sub-system
7142 referenced using a handle.",
7143         "oic.sec.encoding.raw - Raw hex encoded data."
7144     ],
7145     "enum": [
7146         "oic.sec.encoding.jwt",
7147         "oic.sec.encoding.cwt",
7148         "oic.sec.encoding.base64",
7149         "oic.sec.encoding.uri",
7150         "oic.sec.encoding.handle",
7151         "oic.sec.encoding.raw"
7152     ],
7153     "type": "string"
7154 },
7155 "handle": {
7156     "description": "Handle to a key storage Resource.",
7157     "type": "integer"
7158 },
7159 },
7160 "required": [
7161     "encoding"
7162 ],
7163 "type": "object"
7164 },
7165 "publicdata": {
7166     "description": "Public credential information.",
7167     "properties": {
7168         "data": {
7169             "description": "The encoded value.",
7170             "maxLength": 3072,
7171             "type": "string"
7172         },
7173         "encoding": {
7174             "description": "A string specifying the encoding format of the data contained in
7175 the pubdata.",
7176             "x-detail-desc": [
7177                 "oic.sec.encoding.jwt - RFC7517 JSON web token (JWT) encoding.",
7178                 "oic.sec.encoding.cwt - RFC CBOR web token (CWT) encoding.",
7179                 "oic.sec.encoding.base64 - Base64 encoded object.",
7180                 "oic.sec.encoding.uri - URI reference.",
7181                 "oic.sec.encoding.pem - Encoding for PEM encoded certificate or chain.",
7182                 "oic.sec.encoding.der - Encoding for DER encoded certificate.",
7183                 "oic.sec.encoding.raw - Raw hex encoded data."
7184             ],
7185             "enum": [
7186                 "oic.sec.encoding.jwt",
7187                 "oic.sec.encoding.cwt",
7188                 "oic.sec.encoding.base64",
7189                 "oic.sec.encoding.uri",
7190                 "oic.sec.encoding.pem",
7191                 "oic.sec.encoding.der",
7192                 "oic.sec.encoding.raw"
7193             ],
7194             "type": "string"
7195         }
7196     },
7197     "type": "object"
7198 },
7199 "roleid": {
7200     "description": "The role this credential possesses\nSecurity role specified as an
7201 <Authority> & <Rolename>. A NULL <Authority> refers to the local entity or Device.",
7202     "properties": {
7203         "authority": {
7204             "description": "The Authority component of the entity being identified. A NULL
7205 <Authority> refers to the local entity or Device.",
7206             "type": "string"
7207         },
7208         "role": {

```

```

7209         "description": "The ID of the role being identified.",
7210         "type": "string"
7211     },
7212 },
7213     "required": [
7214         "role"
7215     ],
7216     "type": "object"
7217 },
7218     "subjectuuid": {
7219         "anyOf": [
7220             {
7221                 "description": "The id of the Device, which the cred entry applies to or \"*\
7222 for wildcard identity.",
7223                 "pattern": "^\\*$",
7224                 "type": "string"
7225             },
7226             {
7227                 "description": "Format pattern according to IETF RFC 4122.",
7228                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-
7229 F0-9]{12}$",
7230                 "type": "string"
7231             }
7232         ]
7233     },
7234 },
7235     "type": "object"
7236 },
7237     "type": "array"
7238 },
7239 },
7240     "type": "object",
7241     "required": ["roles"]
7242 }
7243 }
7244 }
7245

```

## 7246 C.7.5 Property definition

7247 Table C-11 defines the Properties that are part of the "oic.r.roles" Resource Type.

7248 **Table C-11 – The Property definitions of the Resource with type "rt" = "oic.r.roles".**

| Property name | Value type                 | Mandatory | Access mode | Description                                   |
|---------------|----------------------------|-----------|-------------|---|
| rt            | array: see schema          | No        | Read Only   | Resource Type of the Resource.                |
| n             | multiple types: see schema | No        | Read Write  |   |
| id            | multiple types: see schema | No        | Read Write  |   |
| roles         | array: see schema          | Yes       | Read Write  | List of role certificates.                    |
| if            | array: see schema          | No        | Read Only   | The interface set supported by this Resource. |
| roles         | array: see schema          | Yes       | Read Write  | List of role certificates.                    |

## 7249 C.7.6 CRUDN behaviour

7250 Table C-12 defines the CRUDN operations that are supported on the "oic.r.roles" Resource Type.

7251 **Table C-12 – The CRUDN operations of the Resource with type "rt" = "oic.r.roles".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   | delete | observe |

## 7252 **C.8 Security Profile**

### 7253 **C.8.1 Introduction**

7254 Resource specifying supported and active security profile(s).

7255

### 7256 **C.8.2 Well-known URI**

7257 /oic/sec/sp

### 7258 **C.8.3 Resource type**

7259 The Resource Type is defined as: "oic.r.sp".

### 7260 **C.8.4 OpenAPI 2.0 definition**

```

7261 {
7262   "swagger": "2.0",
7263   "info": {
7264     "title": "Security Profile",
7265     "version": "2019-02-08",
7266     "license": {
7267       "name": "OCF Data Model License",
7268       "url":
7269         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7270         CENSE.md",
7271       "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
7272         reserved."
7273     },
7274     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7275   },
7276   "schemes": ["http"],
7277   "consumes": ["application/json"],
7278   "produces": ["application/json"],
7279   "paths": {
7280     "/oic/sec/sp" : {
7281       "get": {
7282         "description": "Resource specifying supported and active security profile(s).\n",
7283         "parameters": [
7284           {"$ref": "#/parameters/interface"}
7285         ],
7286         "responses": {
7287           "200": {
7288             "description": "",
7289             "x-example":
7290               {
7291                 "rt": ["oic.r.sp"],
7292                 "supportedprofiles" : ["1.3.6.1.4.1.51414.0.0.1.0", " 1.3.6.1.4.1.51414.0.0.2.0"],
7293                 "currentprofile" : "1.3.6.1.4.1.51414.0.0.1.0"
7294               },
7295             "schema": { "$ref": "#/definitions/SP" }
7296           },
7297           "400": {
7298             "description": "The request is invalid."
7299           }
7300         }
7301       },
7302       "post": {
7303         "description": "Sets or updates Device provisioning status data.\n",
7304         "parameters": [
7305           {"$ref": "#/parameters/interface"},
7306           {
7307             "name": "body",

```

```

7308         "in": "body",
7309         "required": true,
7310         "schema": { "$ref": "#/definitions/SP-Update" },
7311         "x-example":
7312         {
7313             "supportedprofiles" : ["1.3.6.1.4.1.51414.0.0.1.0", " 1.3.6.1.4.1.51414.0.0.2.0"],
7314             "currentprofile" : "1.3.6.1.4.1.51414.0.0.1.0"
7315         }
7316     },
7317 ],
7318 "responses": {
7319     "200": {
7320         "description" : "",
7321         "x-example":
7322         {
7323             "rt": ["oic.r.sp"],
7324             "supportedprofiles" : ["1.3.6.1.4.1.51414.0.0.1.0", " 1.3.6.1.4.1.51414.0.0.2.0"],
7325             "currentprofile" : "1.3.6.1.4.1.51414.0.0.1.0"
7326         },
7327         "schema": { "$ref": "#/definitions/SP" }
7328     },
7329     "400": {
7330         "description" : "The request is invalid."
7331     }
7332 }
7333 },
7334 },
7335 },
7336 "parameters": {
7337     "interface" : {
7338         "in" : "query",
7339         "name" : "if",
7340         "type" : "string",
7341         "enum" : [ "oic.if.baseline", "oic.if.rw" ]
7342     }
7343 },
7344 "definitions": {
7345     "SP" : {
7346         "properties": {
7347             "rt": {
7348                 "description": "Resource Type of the Resource.",
7349                 "items": {
7350                     "maxLength": 64,
7351                     "type": "string",
7352                     "enum": ["oic.r.sp"]
7353                 },
7354                 "minItems": 1,
7355                 "readOnly": true,
7356                 "type": "array"
7357             },
7358             "n": {
7359                 "$ref":
7360                 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7361                 schema.json#/definitions/n"
7362             },
7363             "id": {
7364                 "$ref":
7365                 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7366                 schema.json#/definitions/id"
7367             },
7368             "currentprofile": {
7369                 "description": "Security Profile currently active.",
7370                 "type": "string"
7371             },
7372             "supportedprofiles": {
7373                 "description": "Array of supported Security Profiles.",
7374                 "items": {
7375                     "type": "string"
7376                 },
7377                 "type": "array"
7378             },
7379             "if": {

```



```

7380         "description": "The interface set supported by this Resource.",
7381         "items": {
7382             "enum": [ "oic.if.baseline", "oic.if.rw" ],
7383             "type": "string"
7384         },
7385         "minItems": 1,
7386         "readOnly": true,
7387         "type": "array"
7388     },
7389 },
7390 "type" : "object",
7391 "required": ["supportedprofiles", "currentprofile"]
7392 },
7393 "SP-Update" : {
7394     "properties": {
7395         "currentprofile": {
7396             "description": "Security Profile currently active.",
7397             "type": "string"
7398         },
7399         "supportedprofiles": {
7400             "description": "Array of supported Security Profiles.",
7401             "items": {
7402                 "type": "string"
7403             },
7404             "type": "array"
7405         }
7406     },
7407     "type" : "object"
7408 }
7409 }
7410 }
7411

```

### C.8.5 Property definition

Table C-13 defines the Properties that are part of the "oic.r.sp" Resource Type.

**Table C-13 – The Property definitions of the Resource with type "rt" = "oic.r.sp".**

| Property name     | Value type                 | Mandatory | Access mode | Description                                   |
|-------------------|----------------------------|-----------|-------------|---|
| rt                | array: see schema          | No        | Read Only   | Resource Type of the Resource.                |
| n                 | multiple types: see schema | No        | Read Write  |   |
| id                | multiple types: see schema | No        | Read Write  |   |
| currentprofile    | string                     | Yes       | Read Write  | Security Profile currently active.            |
| supportedprofiles | array: see schema          | Yes       | Read Write  | Array of supported Security Profiles.         |
| if                | array: see schema          | No        | Read Only   | The interface set supported by this Resource. |
| currentprofile    | string                     |           | Read Write  | Security Profile currently active.            |
| supportedprofiles | array: see schema          |           | Read Write  | Array of supported Security Profiles.         |

### C.8.6 CRUDN behaviour

Table C-14 defines the CRUDN operations that are supported on the "oic.r.sp" Resource Type.

7417 **Table C-14 – The CRUDN operations of the Resource with type "rt" = "oic.r.sp".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   |        | observe |

## 7418 **C.9 Auditable Event List**

### 7419 **C.9.1 Introduction**

7420 This Resource contains the Auditable Events that have been logged on the Device.

### 7421 **C.9.2 Well-known URI**

7422 /oic/sec/ael

### 7423 **C.9.3 Resource type**

7424 The Resource Type is defined as: "oic.r.ael".

### 7425 **C.9.4 OpenAPI 2.0 definition**

```

7426 {
7427     "swagger": "2.0",
7428     "info": {
7429         "title": "Auditable Event List",
7430         "version": "2019-10-03",
7431         "license": {
7432             "name": "OCF Data Model License",
7433             "url":
7434 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7435 CENSE.md",
7436             "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights
7437 reserved."
7438         },
7439         "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7440     },
7441     "schemes": ["http"],
7442     "consumes": ["application/json"],
7443     "produces": ["application/json"],
7444     "paths": {
7445         "/AelResURI": {
7446             "get": {
7447                 "description": "This Resource contains the Auditable Events that have
7448 been logged on the Device.",
7449                 "parameters": [{" $ref": "#/parameters/interface"}],
7450                 "responses": {
7451                     "200": {
7452                         "description": "Example response payload. In this
7453 example, 'oic.d.light' Device has logged 2 Auditable Event Entries: Update attempt against
7454 '/room1/led1' Resource was denied, and Delete attempt against '/room1/led1' Resource was denied.
7455 Both Auditable Event Entries belong to 'AccessControl (0x01)' category and 'WARN' priority (2).",
7456                         "x-example": {
7457                             "rt": [ "oic.r.ael" ],
7458                             "events": [
7459                                 {
7460                                     "aeid": "AC-1",
7461                                     "category": 1,
7462                                     "priority": 2,
7463                                     "timestamp": "2018-11-
7464 13T20:22:39+00:00",
7465                                     "message": "Access Denied",
7466                                     "auxiliaryinfo":
7467 [ "[2001::1]:1234", "0f33887b-f7d6-4fdb-9125-dd4b60d5aaae", "/room1/led1", "UPDATE", "RFNOP", "No
7468 roles asserted" ]
7469                                 },
7470                                 {
7471                                     "aeid": "AC-1",
7472                                     "category": 1,
7473                                     "priority": 2,

```

```

7474                                     "timestamp": "2018-11-
7475 13T20:20:00+00:00",
7476                                     "message": "Access Denied",
7477                                     "auxiliaryinfo":
7478 [ "[2001::1]:1234", "0f33887b-f7d6-4fdb-9125-dd4b60d5aaae", "/room1/led1", "DELETE", "RFNOP", "No
7479 roles asserted" ]
7480                                     }
7481                                     },
7482                                     "usedspace": 2,
7483                                     "maxspace": 5,
7484                                     "categoryfilter": 3,
7485                                     "priorityfilter": 1
7486                                     },
7487                                     "schema": { "$ref": "#/definitions/Ael" }
7488                                     }
7489                                     },
7490                                     },
7491                                     "post": {
7492                                     "description": "An UPDATE operation may set the 'categoryfilter'
7493 and/or 'priorityfilter' Properties.",
7494                                     "parameters": [
7495                                     {
7496                                     "$ref": "#/parameters/interface"
7497                                     },
7498                                     {
7499                                     "in": "body",
7500                                     "name": "body",
7501                                     "required": true,
7502                                     "schema": { "$ref": "#/definitions/Ael-Update" },
7503                                     "x-example": {
7504                                     "categoryfilter": 3,
7505                                     "priorityfilter": 1
7506                                     }
7507                                     }
7508                                     ],
7509                                     "responses": {
7510                                     "204": {
7511                                     "description": "The new categoryfilter and
7512 priorityfilter were set."
7513                                     }
7514                                     }
7515                                     }
7516                                     },
7517                                     },
7518                                     "parameters": {
7519                                     "interface": {
7520                                     "in": "query",
7521                                     "name": "if",
7522                                     "type": "string",
7523                                     "enum": [ "oic.if.baseline", "oic.if.rw" ]
7524                                     }
7525                                     },
7526                                     "definitions": {
7527                                     "Aee": {
7528                                     "description": "Auditable Event Entry logged by a Device",
7529                                     "type": "object",
7530                                     "properties": {
7531                                     "aeid": {
7532                                     "description": "Identity of the logged event",
7533                                     "type": "string",
7534                                     "readOnly": true
7535                                     },
7536                                     "category": {
7537                                     "description": "Category of this Auditable Event: 0x01
7538 (Access Control), 0x02 (Onboarding), 0x04 (Device), 0x08 (Authentication), 0x10 (SVR Modification),
7539 0x20 (Cloud), 0x40 (Communication), 0x80 (Reserved)",
7540                                     "type": "integer",
7541                                     "enum": [
7542                                     1, 2, 4, 8, 16, 32, 64, 128
7543                                     ],
7544                                     "readOnly": true
7545                                     }
7546                                     },

```

```

7546         "priority": {
7547             "definitions": "Priority of this Auditable Event: 0 (CRIT), 1
7548 (ERR), 2 (WARN), 3 (INFO), 4 (DEBUG)",
7549             "type": "integer",
7550             "enum": [
7551                 0, 1, 2, 3, 4
7552             ],
7553             "readOnly": true
7554         },
7555         "timestamp": {
7556             "description": "Time when this Auditable Event occurred",
7557             "type": "string",
7558             "format": "date-time",
7559             "readOnly": true
7560         },
7561         "message": {
7562             "description": "Description for this Auditable Event",
7563             "type": "string",
7564             "readOnly": true
7565         },
7566         "auxiliaryinfo": {
7567             "description": "Supplementary info for Auditable Event
7568 message. (e.g. URI of specific Resource in ACE2 for 'Access Denied' message)",
7569             "type": "array",
7570             "minItems": 0,
7571             "items": {
7572                 "type": "string"
7573             },
7574             "readOnly": true
7575         }
7576     },
7577     "required": [
7578         "aeid", "message", "auxiliaryinfo", "category", "priority",
7579         "timestamp"
7580     ],
7581 },
7582
7583 "Ael": {
7584     "description": "Resource for storing Auditable Events List",
7585     "type": "object",
7586     "properties": {
7587         "rt": {
7588             "description": "Resource Type",
7589             "type": "array",
7590             "minItems": 1,
7591             "uniqueItems": true,
7592             "items": {
7593                 "maxLength": 64,
7594                 "type": "string",
7595                 "enum": [ "oic.r.ael" ]
7596             },
7597             "readOnly": true
7598         },
7599         "n": {
7600             "$ref":
7601 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7602 schema.json#/definitions/n"
7603         },
7604         "id": {
7605             "$ref":
7606 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7607 schema.json#/definitions/id"
7608         },
7609         "if": {
7610             "description": "The OCF Interface set supported by this
7611 Resource",
7612             "type": "array",
7613             "minItems": 2,
7614             "uniqueItems": true,
7615             "items": {
7616                 "type": "string",
7617                 "enum": [ "oic.if.baseline", "oic.if.rw" ]

```

```

7618         },
7619         "readOnly": true
7620     },
7621     "events": {
7622         "description": "This list stores AEEs whose 'category'
7623 Property value is filtered by 'categoryfilter' Property and 'priority' Property value is equal or
7624 less than the value of 'priorityfilter' Property.",
7625         "type": "array",
7626         "uniqueItems": true,
7627         "items": {
7628             "$ref": "#/definitions/Aee"
7629         }
7630     },
7631     "usedspace": {
7632         "description": "Current used space for logged AEEs. The
7633 Device updates this Property whenever new AEEs are logged.",
7634         "type": "integer",
7635         "default": 0,
7636         "readOnly": true
7637     },
7638     "maxspace": {
7639         "description": "This means the maximum allowable storage size
7640 for AEEs that can be stored in 'events' list. The Manufacturer chooses this value.",
7641         "type": "integer",
7642         "readOnly": true
7643     },
7644     "unit": {
7645         "description": "The unit for 'usedspace' and 'maxspace'
7646 Properties. The Manufacturer chooses this value.",
7647         "type": "string",
7648         "enum": [
7649             "Kbyte",
7650             "Byte"
7651         ],
7652         "default": "Byte",
7653         "readOnly": true
7654     },
7655     "categoryfilter": {
7656         "description": "This value decides what categories of AEEs
7657 are to be logged. Meaning of each bit: 0x01 (Access Control), 0x02 (Onboarding), 0x04 (Device), 0x08
7658 (Authentication), 0x10 (SVR Modification), 0x20 (Cloud), 0x40 (Communication), 0x80 (Reserved).
7659 e.g.) if categoryfilter == 0xff: log all events of all categories, e.g.) if categoryfilter == 0x03:
7660 log all events of 'AC (== 0x01)' and 'OB (==0x02)' categories ",
7661         "type": "integer",
7662         "default": 255
7663     },
7664     "priorityfilter": {
7665         "description": "The AEEs whose 'priority' values are equal to
7666 or smaller than this value are logged. A smaller value means a higher priority. Meaning of each
7667 value: 0 (CRIT), 1 (ERR), 2 (WARN), 3 (INFO), 4 (DEBUG). e.g.) if priorityfilter is set to DEBUG
7668 (==4) all AEEs will be logged, e.g.) if priorityfilter is set to 1, CRIT (==0) and ERR (==1) AEEs
7669 will be logged ",
7670         "type": "integer",
7671         "default": 4,
7672         "enum": [
7673             0, 1, 2, 3, 4
7674         ]
7675     },
7676     },
7677     "required": [
7678         "events", "usedspace", "maxspace", "categoryfilter", "priorityfilter"
7679     ],
7680     },
7681     "Ael-Update": {
7682         "type": "object",
7683         "properties": {
7684             "categoryfilter": {
7685                 "description": "This value decides what categories of AEEs
7686 are to be logged. Meaning of each bit: 0x01 (Access Control), 0x02 (Onboarding), 0x04 (Device), 0x08
7687 (Authentication), 0x10 (SVR Modification), 0x20 (Cloud), 0x40 (Communication). e.g.) if
7688 categoryfilter == 0xff: log all events of all categories, e.g.) if categoryfilter == 0x03: log all
7689 events of 'AC (== 0x01)' and 'OB (==0x02)' categories ",

```

```

7690         "type": "integer",
7691         "default": 255
7692     },
7693     "priorityfilter": {
7694         "description": "The AEEs whose 'priority' values are equal to
7695 or smaller than this value are logged. A smaller value means a higher priority. Meaning of each
7696 value: 0 (CRIT), 1 (ERR), 2 (WARN), 3 (INFO), 4 (DEBUG). e.g.) if priorityfilter is set to DEBUG
7697 (==4) all AEEs will be logged, e.g.) if priorityfilter is set to 1, CRIT (==0) and ERR (==1) AEEs
7698 will be logged ",
7699         "type": "integer",
7700         "default": 4,
7701         "enum": [
7702             0, 1, 2, 3, 4
7703         ]
7704     },
7705 },
7706 "required": [
7707     "categoryfilter", "priorityfilter"
7708 ],
7709 },
7710 },
7711 },
7712 },
7713 }

```

### 7714 C.9.5 Property definition

7715 Table C-15 defines the Properties that are part of the "oic.r.ael" Resource Type.

7716 **Table C-15 – The Property definitions of the Resource with type "rt" = "oic.r.ael".**

| Property name | Value type                 | Mandatory | Access mode | Description  |
|---------------|----------------------------|-----------|-------------|--|
| aeid          | string                     | Yes       | Read Only   | Identity of the logged event   |
| category      | integer                    | Yes       | Read Only   | Category of this Auditable Event: 0x01 (Access Control), 0x02 (Onboarding), 0x04 (Device), 0x08 (Authentication), 0x10 (SVR Modification), 0x20 (Cloud), 0x40 (Communication), 0x80 (Reserved) |
| priority      | integer                    | Yes       | Read Only   |  |
| timestamp     | string                     | Yes       | Read Only   | Time when this Auditable Event occurred  |
| message       | string                     | Yes       | Read Only   | Description for this Auditable Event   |
| auxiliaryinfo | array: see schema          | Yes       | Read Only   | Supplementary info for Auditable Event message. (e.g. URI of specific Resource in ACE2 for 'Access Denied' message)  |
| rt            | array: see schema          | No        | Read Only   | Resource Type  |
| n             | multiple types: see schema | No        | Read Write  |  |
| id            | multiple types: see schema | No        | Read Write  |  |

|                |                   |     |            |   |
|----------------|-------------------|-----|------------|---|
| if             | array: see schema | No  | Read Only  | The OCF Interface set supported by this Resource  |
| events         | array: see schema | Yes | Read Write | This list stores AEEs whose 'category' Property value is filtered by 'categoryfilter' Property and 'priority' Property value is equal or less than the value of 'priorityfilter' Property.  |
| usedspace      | integer           | Yes | Read Only  | Current used space for logged AEEs. The Device updates this Property whenever new AEEs are logged.  |
| maxspace       | integer           | Yes | Read Only  | This means the maximum allowable storage size for AEEs that can be stored in 'events' list. The Manufacturer chooses this value.  |
| unit           | string            | No  | Read Only  | The unit for 'usedspace' and 'maxspace' Properties. The Manufacturer chooses this value.  |
| categoryfilter | integer           | Yes | Read Write | This value decides what categories of AEEs are to be logged. Meaning of each bit: 0x01 (Access Control), 0x02 (Onboarding), 0x04 (Device), 0x08 (Authentication), 0x10 (SVR Modification), 0x20 (Cloud), 0x40 (Communication), 0x80 (Reserved). e.g.) if categoryfilter == 0xff: log all events of all categories, e.g.) if categoryfilter == 0x03: log all events of 'AC (== 0x01)' and 'OB (==0x02)' categories |
| priorityfilter | integer           | Yes | Read Write | The AEEs whose 'priority' values are equal to or smaller than this value are logged. A smaller value means a higher priority. Meaning of each value: 0 (CRIT), 1 (ERR), 2 (WARN), 3   |

|                |         |     |            |  |
|----------------|---------|-----|------------|--|
|                |         |     |            | (INFO), 4 (DEBUG). e.g.) if priorityfilter is set to DEBUG (==4) all AEEs will be logged, e.g.) if priorityfilter is set to 1, CRIT (==0) and ERR (==1) AEEs will be logged  |
| categoryfilter | integer | Yes | Read Write | This value decides what categories of AEEs are to be logged. Meaning of each bit: 0x01 (Access Control), 0x02 (Onboarding), 0x04 (Device), 0x08 (Authentication), 0x10 (SVR Modification), 0x20 (Cloud), 0x40 (Communication). e.g.) if categoryfilter == 0xff: log all events of all categories, e.g.) if categoryfilter == 0x03: log all events of 'AC (== 0x01)' and 'OB (==0x02)' categories |
| priorityfilter | integer | Yes | Read Write | The AEEs whose 'priority' values are equal to or smaller than this value are logged. A smaller value means a higher priority. Meaning of each value: 0 (CRIT), 1 (ERR), 2 (WARN), 3 (INFO), 4 (DEBUG). e.g.) if priorityfilter is set to DEBUG (==4) all AEEs will be logged, e.g.) if priorityfilter is set to 1, CRIT (==0) and ERR (==1) AEEs will be logged                                  |

## C.9.6 CRUDN behaviour

Table C-16 defines the CRUDN operations that are supported on the "oic.r.ael" Resource Type.

**Table C-16 – The CRUDN operations of the Resource with type "rt" = "oic.r.ael".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   |        | observe |

## C.10 Security Domain Information

### C.10.1 Introduction

This Resource contains the information that identifies the OCF Security Domain to which the device belongs.



## 7725 C.10.2 Well-known URI

7726 /oic/sec/sdi

## 7727 C.10.3 Resource type

7728 The Resource Type is defined as: "oic.r.sdi".

## 7729 C.10.4 OpenAPI 2.0 definition

```
7730 {
7731   "swagger": "2.0",
7732   "info": {
7733     "title": "Security Domain Information",
7734     "version": "2019-10-01",
7735     "license": {
7736       "name": "OCF Data Model License",
7737       "url":
7738         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
7739         CENSE.md",
7740       "x-copyright": "copyright 2016-2017, 2019 Open Connectivity Foundation, Inc. All rights
7741         reserved."
7742     },
7743     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
7744   },
7745   "schemes": ["http"],
7746   "consumes": ["application/json"],
7747   "produces": ["application/json"],
7748   "paths": {
7749     "/oic/sec/sdi" : {
7750       "get": {
7751         "description": "This Resource contains the information that identifies the OCF Security
7752         Domain to which the device belongs.\n",
7753         "parameters": [
7754           {"$ref": "#/parameters/interface"}
7755         ],
7756         "responses": {
7757           "200": {
7758             "description": "Success",
7759             "x-example":
7760               {
7761                 "rt": ["oic.r.sdi"],
7762                 "uuid": "de305d54-75b4-431b-adb2-eb6b9e546014",
7763                 "name": "Home",
7764                 "priv": true
7765               },
7766             "schema": { "$ref": "#/definitions/Sdi" }
7767           },
7768           "400": {
7769             "description": "The request is invalid."
7770           }
7771         }
7772       },
7773       "post": {
7774         "description": "Provision the OCF Security Domain information.\n",
7775         "parameters": [
7776           {"$ref": "#/parameters/interface"},
7777           {
7778             "name": "body",
7779             "in": "body",
7780             "required": true,
7781             "schema": { "$ref": "#/definitions/Sdi-Update" },
7782             "x-example": {
7783               "uuid": "de305d54-75b4-431b-adb2-eb6b9e546014",
7784               "name": "Home",
7785               "priv": false
7786             }
7787           }
7788         ],
7789         "responses": {
7790           "400": {
7791             "description": "The request is invalid."
7792           }
7793         }
7794       }
7795     }
7796   }
```

```

7792         },
7793         "204": {
7794             "description": "The SDI is updated.",
7795             "schema": { "$ref": "#/definitions/Sdi-Update" },
7796             "x-example": {
7797                 "uuid": "de305d54-75b4-431b-adb2-eb6b9e546014",
7798                 "name": "Home",
7799                 "priv": false
7800             }
7801         }
7802     }
7803 }
7804 },
7805 },
7806 "parameters": {
7807     "interface": {
7808         "in": "query",
7809         "name": "if",
7810         "type": "string",
7811         "enum": [ "oic.if.rw", "oic.if.baseline" ]
7812     }
7813 },
7814 "definitions": {
7815     "Sdi": {
7816         "properties": {
7817             "uuid": {
7818                 "$ref": "https://openconnectivityfoundation.github.io/core/schemas/oic.types-
7819 schema.json#/definitions/uuid"
7820             },
7821             "name": {
7822                 "description": "Human-friendly name for the Security Domain, set by DOTS during
7823 onboarding.",
7824                 "type": "string"
7825             },
7826             "rt": {
7827                 "description": "Resource Type of the Resource.",
7828                 "items": {
7829                     "maxLength": 64,
7830                     "type": "string",
7831                     "enum": [ "oic.r.sdi" ]
7832                 },
7833                 "minItems": 1,
7834                 "readOnly": true,
7835                 "type": "array"
7836             },
7837             "n": {
7838                 "$ref":
7839 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7840 schema.json#/definitions/n"
7841             },
7842             "id": {
7843                 "$ref":
7844 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
7845 schema.json#/definitions/id"
7846             },
7847             "priv": {
7848                 "description": "Flag to indicate whether the Security Domain Information is copied to
7849 "/oic/res", and thus, whether it is publicly visible or private.",
7850                 "type": "boolean"
7851             },
7852             "if": {
7853                 "description": "The interface set supported by this Resource.",
7854                 "items": {
7855                     "enum": [ "oic.if.rw", "oic.if.baseline" ],
7856                     "type": "string"
7857                 },
7858                 "minItems": 1,
7859                 "readOnly": true,
7860                 "type": "array"
7861             }
7862         },
7863         "type": "object",

```

```

7864     "required": [ "uuid", "name", "priv" ]
7865 },
7866
7867     "Sdi-Update" : {
7868         "properties": {
7869             "uuid": {
7870                 "$ref": "https://openconnectivityfoundation.github.io/core/schemas/oic.types-
7871 schema.json#/definitions/uuid"
7872             },
7873             "name": {
7874                 "description": "Human-friendly name for the Security Domain, set by DOTS during
7875 onboarding.",
7876                 "type": "string"
7877             },
7878             "priv": {
7879                 "description": "Flag to indicate whether the Security Domain Information is copied to
7880 "/oic/res", and thus, whether it is publicly visible or private.",
7881                 "type": "boolean"
7882             }
7883         },
7884         "type" : "object",
7885         "required": [ "name", "priv" ]
7886     }
7887 }
7888 }
7889

```

## 7890 C.10.5 Property definition

7891 Table C-17 defines the Properties that are part of the "oic.r.sdi" Resource Type.

7892 **Table C-17 – The Property definitions of the Resource with type "rt" = "oic.r.sdi".**

| Property name | Value type                 | Mandatory | Access mode | Description  |
|---------------|----------------------------|-----------|-------------|--|
| uuid          | multiple types: see schema | Yes       | Read Write  |  |
| name          | string                     | Yes       | Read Write  | Human-friendly name for the Security Domain, set by DOTS during onboarding.  |
| rt            | array: see schema          | No        | Read Only   | Resource Type of the Resource.   |
| n             | multiple types: see schema | No        | Read Write  |  |
| id            | multiple types: see schema | No        | Read Write  |  |
| priv          | boolean                    | Yes       | Read Write  | Flag to indicate whether the Security Domain Information is copied to "/oic/res", and thus, whether it is publicly visible or private. |
| if            | array: see schema          | No        | Read Only   | The interface set supported by this Resource.  |
| uuid          | multiple types: see schema | No        | Read Write  |  |
| name          | string                     | Yes       | Read Write  | Human-friendly name for the Security Domain, set by DOTS during onboarding.  |

|      |         |     |            |  |
|------|---------|-----|------------|--|
| priv | boolean | Yes | Read Write | Flag to indicate whether the Security Domain Information is copied to "/oic/res", and thus, whether it is publicly visible or private. |
|------|---------|-----|------------|--|

7893 **C.10.6 CRUDN behaviour**

7894 Table C-18 defines the CRUDN operations that are supported on the "oic.r.sdi" Resource Type.

7895 **Table C-18 – The CRUDN operations of the Resource with type "rt" = "oic.r.sdi".**

| Create | Read | Update | Delete | Notify  |
|--------|------|--------|--------|---------|
|        | get  | post   |        | observe |

7896

7897

## Annex D (informative)

### OID definitions

This annex captures the OIDs defined throughout the document. The OIDs listed are intended to be used within the context of an X.509 v3 certificate. MAX is an upper bound for SEQUENCES of UTF8Strings and OBJECT IDENTIFIERS and should not exceed 255.

```
id-OCF OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1)
    private(4) enterprise(1) OCF(51414) }
```

```
-- OCF Security specific OIDs
```

```
id-ocfSecurity OBJECT IDENTIFIER ::= { id-OCF 0 }
id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }
```

```
-- OCF Security Categories
```

```
id-ocfSecurityProfile ::= { id-ocfSecurity 0 }
id-ocfCertificatePolicy ::= { id-ocfSecurity 1 }
```

```
-- OCF Security Profiles
```

```
sp-unspecified ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 0 }
sp-baseline ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 1 }
sp-black ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 2 }
sp-blue ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 3 }
sp-purple ::= OBJECT IDENTIFIER { id-ocfSecurityProfile 4 }
```

```
sp-unspecified-v0 ::= ocfSecurityProfileOID {id-sp-unspecified 0}
sp-baseline-v0 ::= ocfSecurityProfileOID {id-sp-baseline 0}
sp-black-v0 ::= ocfSecurityProfileOID {id-sp-black 0}
sp-blue-v0 ::= ocfSecurityProfileOID {id-sp-blue 0}
sp-purple-v0 ::= ocfSecurityProfileOID {id-sp-purple 0}
```

```
ocfSecurityProfileOID ::= UTF8String
```

```
-- OCF Security Certificate Policies
```

```
ocfCertificatePolicy-v1 ::= { id-ocfCertificatePolicy 2 }
```

```
-- OCF X.509v3 Extensions
```

```
id-ocfX509Extensions OBJECT IDENTIFIER ::= { id-OCF 1 }
id-ocfCompliance OBJECT IDENTIFIER ::= { id-ocfX509Extensions 0 }
id-ocfSecurityClaims OBJECT IDENTIFIER ::= { id-ocfX509Extensions 1 }
id-ocfCPLAttributes OBJECT IDENTIFIER ::= { id-ocfX509Extensions 2 }
```

```
ocfVersion ::= SEQUENCE {
    major    INTEGER,
    minor    INTEGER,
    build    INTEGER}
```

```
ocfCompliance ::= SEQUENCE {
    version        ocfVersion,
    securityProfile SEQUENCE SIZE (1..MAX) OF ocfSecurityProfileOID,
    deviceName     UTF8String,
    deviceManufacturer UTF8String}
```

```
claim-secure-boot ::= ocfSecurityClaimsOID { id-ocfSecurityClaims 0 }
claim-hw-backed-cred-storage ::= ocfSecurityClaimsOID { id-ocfSecurityClaims 1 }
```

```
7958
7959 ocfSecurityClaimsOID ::= OBJECT IDENTIFIER
7960
7961 ocfSecurityClaims ::= SEQUENCE SIZE (1..MAX) of ocfSecurityClaimsOID
7962
7963 cpl-at-IANAPen ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 0 }
7964 cpl-at-model ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 1 }
7965 cpl-at-version ::= OBJECT IDENTIFIER { id-ocfCPLAttributes 2 }
7966
7967 ocfCPLAttributes ::= SEQUENCE {
7968     cpl-at-IANAPen UTF8String,
7969     cpl-at-model UTF8String,
7970     cpl-at-version UTF8String}
```

## Annex E (informative)

### Security considerations specific to Bridged Protocols

The text in this Annex is provided for information only. This Annex has no normative impact. This information is applicable at the time of initial publication and may become out of date.

#### E.1 Security Considerations specific to the AllJoyn Protocol

This clause intentionally left empty.

#### E.2 Security Considerations specific to the Bluetooth LE Protocol

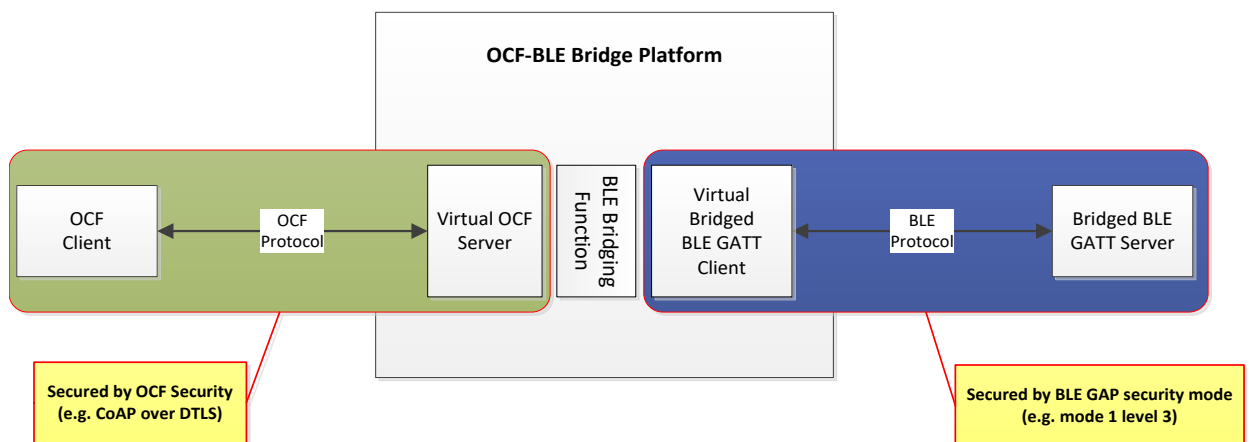
BLE GAP supports two security modes, security mode 1 and security mode 2. Each security mode has several security levels (see Table E.1)

Security mode 1 and Security level 2 or higher would typically be considered secure from an OCF perspective. The appropriate selection of security mode and level is left to the vendor.

**Table E.1 GAP security mode**

| GAP security mode | security level  |
|-------------------|---|
| Security mode 1   | 1 (no security)   |
|                   | 2 (Unauthenticated pairing with encryption)                     |
|                   | 3 (Authenticated pairing with encryption)                       |
|                   | 4 (Authenticated LE Secure Connections pairing with encryption) |
| Security mode 2   | 1 (Unauthenticated pairing with data signing)                   |
|                   | 2 (Authenticated pairing with data signing)                     |

Figure E-1 shows how communications in both ecosystems of OCF-BLE Bridge Platform are secured by their own security.



**Figure E-1 Security Considerations for BLE Bridge**

#### E.3 Security Considerations specific to the oneM2M Protocol

This clause intentionally left empty.

#### E.4 Security Considerations specific to the U+ Protocol

A U+ server supports one of the TLS 1.2 cipher suites as in Table E.2 defined in IETF RFC 5246.  
Copyright Open Connectivity Foundation, Inc. © 2016-2020. All rights Reserved

Table E.2 TLS 1.2 Cipher Suites used by U+

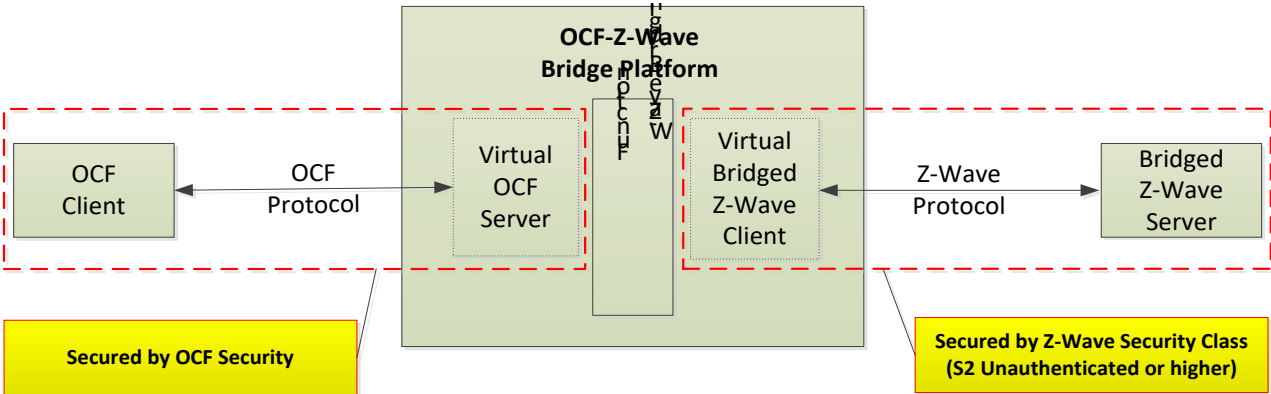
| Cipher Suite                            |
|---|
| TLS_RSA_WITH_AES_128_CBC_SHA256         |
| TLS_RSA_WITH_AES_256_CBC_SHA256         |
| TLS_RSA_WITH_AES_256_CCM                |
| TLS_RSA_WITH_AES_256_CCM_8              |
| TLS_RSA_WITH_AES_256_GCM_SHA384         |
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA256     |
| TLS_DHE_RSA_WITH_AES_256_GCM_SHA384     |
| TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384  |
| TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384  |
| TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384    |
| TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384    |
| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 |
| TLS_ECDHE_ECDSA_WITH_AES_256_CCM        |
| TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8      |
| TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 |
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384   |
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384   |
| TLS_DHE_RSA_WITH_AES_256_CCM            |
| TLS_DHE_RSA_WITH_AES_256_CCM_8          |

7994 The security of the Haier U+ Protocol is proprietary, and further details are presently unavailable.

7995 **E.5 Security Considerations specific to the Z-Wave Protocol**

7996 Z-Wave currently supports two kinds of security class which are S0 Security Class and S2 Security Class, as shown in Table E.3. Bridged Z-wave Servers using S2 Security Class for communication  
7997 with a Virtual Bridged Client would typically be considered secure from an OCF perspective. The  
7998 appropriate selection for S2 Security Class and Class Name is left to the vendor.  
7999

8000 Figure E-2 presents how OCF Client and Bridged Z-Wave Server communicate based upon their  
8001 own security.



8002

8003

Figure E-2 Security Considerations for Z-Wave Bridge



8004 All 3 types of S2 Security Class such as S2 Access Control, S2 Authenticated and S2  
8005 Unauthenticated provides the following advantages from the security perspective;

- 8006 – The unique device specific key for every secure device enables validation of device identity and  
8007 prevents man-in-the-middle compromises to security
- 8008 – The Secure cryptographic key exchange methods during inclusion achieves high level of  
8009 security between the Virtual Z-Wave Client and the Bridged Z-Wave Server.
- 8010 – Out of band key exchange for product authentication which is combined with device specific  
8011 key prevents eavesdropping and man-in-the-middle attack vectors.

8012 See Table E.3 for a summary of Z-Wave Security Classes.

8013 **Table E.3 Z-Wave Security Class**

| Security Class | Class Name         | Validation of device identity | Key Exchange                      | Message Encapsulation          |
|----------------|--------------------|-------------------------------|-----------------------------------|--------------------------------|
| S2             | S2 Access Control  | Device Specific key           | Out-of-band inclusion             | Encrypted command transmission |
|                | S2 Authenticated   | Device Specific key           | Out-of-band inclusion             | Encrypted command transmission |
|                | S2 Unauthenticated | Device Specific key           | Z-wave RF band used for inclusion | Encrypted command transmission |
| S0             | S0 Authenticated   | N/A                           | Z-wave RF band used for inclusion | Encrypted command transmission |

8014 On the other hand, S0 Security Class has the vulnerability of security during inclusion by  
8015 exchanging of temporary 'well-known key' (e.g. 1234). As a result of that, it could lead the  
8016 disclosure of the network key if the log of key exchange methods is captured, so Z-Wave devices  
8017 might be no longer secure in that case.

## 8018 **E.6 Security Considerations specific to the Zigbee Protocol**

8019 The Zigbee 3.0 stack supports multiple security levels. A security level is supported by both the  
8020 network (NWK) layer and application support (APS) layer. A security attribute in the Zigbee 3.0  
8021 stack, "nwkSecurityLevel", represents the security level of a device.

8022 The security level nwkSecurityLevel > 0x04 provides message integrity code (MIC) and/or AES128-  
8023 CCM encryption (ENC). Zigbee Servers using nwkSecurityLevel > 0x04 would typically be  
8024 considered secure from an OCF perspective. The appropriate selection for nwkSecurityLevel is left  
8025 to the vendor.

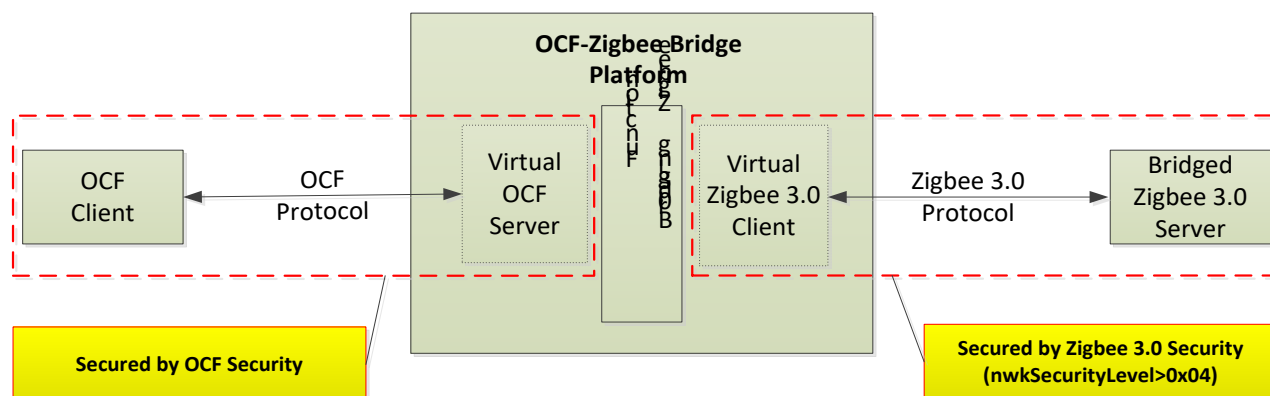
8026 See Table E.4 for a summary of the Zigbee Security Levels.

8027 **Table E.4 Zigbee 3.0 Security Levels to the Network, and Application Support layers**

| Security Level Identifier | Security Level Sub-Field | Security Attributes | Data Encryption | Frame Integrity (Length of M of MIC, in Number of Octets) |
|---------------------------|--------------------------|---------------------|-----------------|---|
| 0x00                      | '000'                    | None                | OFF             | NO (M=0)  |
| 0x01                      | '001'                    | MIC-32              | OFF             | YES(M=4)  |
| 0x02                      | '010'                    | MIC-64              | OFF             | YES(M=8)  |
| 0x03                      | '011'                    | MIC-128             | OFF             | YES(M=16)   |

|      |       |             |    |           |
|------|-------|-------------|----|-----------|
| 0x04 | '100' | ENC         | ON | NO(M=0)   |
| 0x05 | '101' | ENC-MIC-32  | ON | YES(M=4)  |
| 0x06 | '110' | ENC-MIC-64  | ON | YES(M=8)  |
| 0x07 | '111' | ENC-MIC-128 | ON | YES(M=16) |

Figure E-3 shows how communications in both ecosystems of OCF-Zigbee Bridge Platform are secured by their own security.



**Figure E-3 Security Considerations for Zigbee Bridge**

## E.7 Security Considerations specific to the the EnOcean Radio Protocol

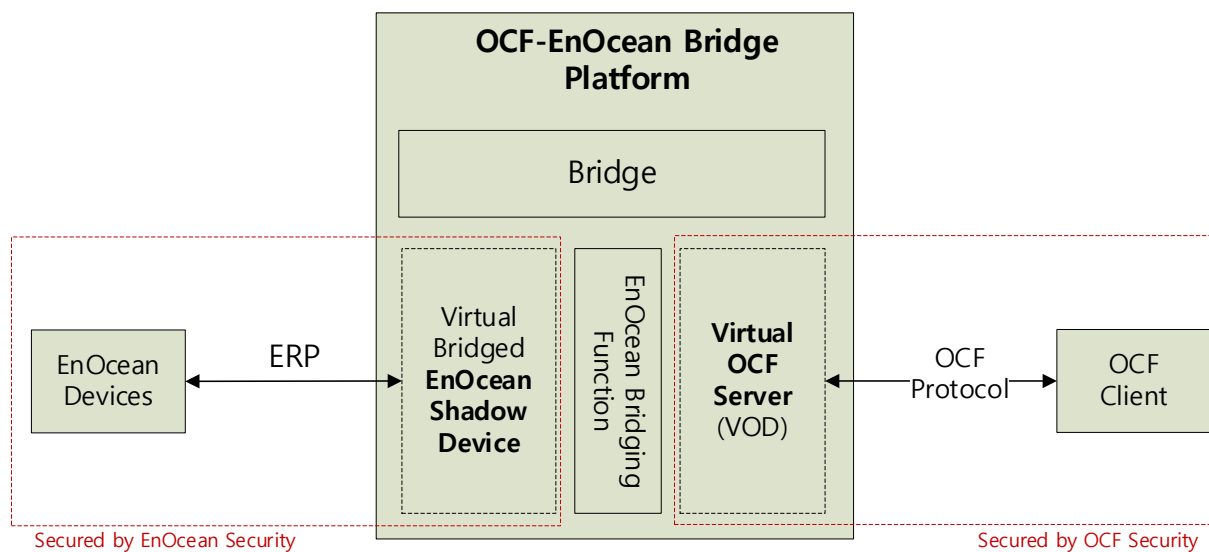
The EnOcean Radio Protocol supports four different security levels. The security level depends on which security mechanisms are used. Table E.5 defines them

**Table E.5 EnOcean Radio Protocol security levels**

| Level | Features                                 | Replay Attack Vulnerability | Eavesdropping Vulnerability |
|-------|--|-----------------------------|-----------------------------|
| 0     | No Features (Unsecure)                   | Yes                         | Yes                         |
| 1     | With Encryption only                     | Yes                         | No                          |
| 2     | Without Encryption but with RLC and CMAC | No                          | Yes                         |
| 3     | With Encryption, RLC and CMAC            | No                          | No                          |

The security levels 1 and 2 have been declared deprecated and shall not longer be used. Security level 3 uses Variable AES Encryption, Rolling Code (RLC) and a cipher-based message authentication code (CMAC) with private keys and public vectors. Technically each feature can be combined with every other feature, even if it is obsolete or unreasonable.

Figure E-4 shows how communications in both ecosystems of OCF- EnOcean Bridge Platform are secured by their own security



**Figure E-4 Security Considerations for EnOcean Bridge**