Zigbee Direct Specification

Revision 1.1

Connectivity Standards Alliance Document 20-27688-041

October 16th, 2024

Sponsored by: Connectivity Standards Alliance™

Accepted by	This document has been accepted for release by the Alliance Board of Directors.
Abstract	This document describes the Zigbee Direct specification. Zigbee Direct allows for encrypted and authenticated access to Zigbee devices and networks from Bluetooth Low Energy ("BLE") capable devices. This enables devices like smartphones, tablets, smart speakers and hubs to commission and control devices on a Zigbee network.
Keywords	Zigbee Direct, ZVD, ZDD

Copyright © 2024 Connectivity Standards Alliance, Inc.
508 Second Street, Suite 206 Davis, CA 95616 - USA
www.csa-iot.org
All rights reserved.

Permission is granted to members of the Connectivity Standards Alliance to reproduce this document for their own use or the use of other Connectivity Standards Alliance members only, provided this notice is included. All other rights reserved. Duplication for sale, or for commercial or for-profit use is strictly prohibited without the prior written consent of the Connectivity Standards Alliance.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Notice and disclaimer

Copyright © 2024 Connectivity Standards Alliance. All rights Reserved. This information within this document is the property of the Connectivity Standards Alliance and its use and disclosure are restricted.

Elements of this document may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (such third party may or may not be a member of CSA). CSA is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third-party intellectual property rights.

No right to use any CSA name, logo or trademark is conferred herein. Use of any CSA name, logo or trademark requires membership in the CSA and compliance with the CSA Trademark and Logo Usage Guidelines and Terms and related CSA policies.

This document and the information contained herein are provided on an “AS IS” basis and CSA DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT. IN NO EVENT WILL CSA BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. All company, brand and product names may be trademarks that are the sole property of their respective owners.

This notice and disclaimer must be included on all copies of this document.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Revision history

Revisal	Date	Details	Editor
000	10/04/2019	First draft	Bożena Erdman
001	07/01/2020	Template change	Bozena Erdmann
002	07/16/2020	Changes to the input originally submitted to the Alliance, based on the BG discussion	Bozena Erdmann
003	08/05/2020	Included updates to the configuration cluster, conversion to asciidoc	Sander Raaijmakers
004	10/22/2020	GATT DB verification removed	Piotr Polak
005	10/23/2020	Moving away from Upper and Lower NWK: new architecture. NPDU Request and Indication now combined into NPDU Message.	Rob Houtepen
006	10/30/2020	Added annex with example Bluetooth advertisement	Robin Michielsen
007	11/06/2020	Add officially allocated Bluetooth SIG UUID	Sander Raaijmakers
008	11/11/2020	Key confirmation messages added to the secure session establishment flow	Piotr Polak
009	11/18/2020	Outdated (PoC 1) Type IDs and NPDU handle removed in Tunnel payload format	Robin Michielsen
010	12/08/2020	128-bit UUIDs moved from temporary annex to GATT service descriptions. Updated/aligned the intro of each GATT service.	Robin Michielsen
011	12/16/2020	Several changes to the Commissioning Service.	Robin Michielsen
012	01/11/2021	Some BLE ATT MTU requirements were added	Oleg Gladyshev
013	01/06/2021	Added functional description for ZVD	Sander Raaijmakers
014	01/06/2021	Added status code TLV and its use to the Commissioning Service	Marius Munder
015	01/13/2021	ZVD inband provisioning added, session establishment handshake updated, editorial changes to the security section.	Piotr Polak
016	01/14/2021	Update of functional description of Zigbee Direct Tunnel Service and proposed errata for R23	Rob Houtepen
017	05/10/2021	Added security test vectors for key agreement, key derivation and secure access	Arasch Honarbacht

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

018	05/26/2021	Added guidelines for ZVD EUI-64; added Trusted Link errata for R23; added sections on network key rotation and legacy network support;	Arasch Honarbacht
019	06/14/2021	Fixed (only) editorials after recirculation ballot	Sander Raaijmakers
020	07/07/2021	Security TLVs and constants were aligned with r23 or renamed	Oleg Gladyshev
021	07/07/2021	New commissioning use case was added to provide Link keys to a ZDD TC.	Oleg Gladyshev
022	07/21/2021	Various technical comments	Sander Raaijmakers
023	08/11/2021	Added details on software components	Marius Munder
024	09/06/2021	Amended Status Code TLV; Amended Link-Key TLV; Removed supplemental out-of-band commissioning with check for beacon (similar behavior was available with secure rejoin); Made consistent usage of Network Commissioning Request and mentioned Network Commissioning Response; Added prescriptive ZDD-initiated disconnect in case of network key rotation; Added R23 Errata concerning rejoins with the Trust Center address 0x0000; other minor editorials	Arasch Honarbacht
025	09/14/2021	Prepared specification package for ballot updating editorials, indexes and merging remaining pull requests	Sander Raaijmakers
026	10/18/2021	Replace all references to R23 by R22 where possible, and refer to subset of R23 included in Annex	Sander Raaijmakers
027	10/26/2021	Add clarification to suppress update device if TC = ZVD is reestablishing the connection; always use ZigBeeAlliance09 as the well-known global key between ZVD and ZDD, also when operating on a distributed security network	Arasch Honarbacht
028	11/15/2021	Version for recirculation ballot 0.9	Sander Raaijmakers
029	12/03/2021	Include configuration cluster anonymous join timeout	Sander Raaijmakers
030	02/01/2022	CCB 3604, 3606	Sander Raaijmakers
031	03/04/2022	Align R23 section with latest revision	Sander Raaijmakers
032	03/05/2022	Latest updates for SVE - update version appropriately	Sander Raaijmakers

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

033	07/06/2022	Rename Enhanced Provisioning Session to Limited Authorization Session; Clarifications/consolidation related to Limited Authorization Sessions; add section on routing on behalf of ZVD during Ephemeral Authorization Session; resolve editor's note concerning Ephemeral Authorization Session; further clarifications related to Ephemeral Authorization session; Add sequence diagram and guidance on Trust Center operation of ZVD	Arasch Honarbacht
034	08/22/2022	Further clarify, consolidate and simplify Ephemeral Session handling and APS key usage for transporting ephemeral authorization keys	Arasch Honarbacht
035	09/06/2022	Editorial updates	Sander Raaijmakers
036	10/20/2022	Editorial updates for release	Sander Raaijmakers
037	11/30/2022	Editorial updates for publication	Sander Raaijmakers
038	11/07/2023	Addressed CCBs 3962, 3841 and 3996	Marius Munder
039	02/01/2024	Addressed CCB 3972	Marius Munder
040	04/30/2024	Addressed CCBs 4010, 3868 and 3982	Marius Munder
041	06/12/2024	Editorial updates for release r1.1	Marius Munder

Participants

Thank you to the following participants who provided a small or large contribution in the creation of this specification, in no particular order:

Marius Munder	Arasch Honarbacht
Bożena Erdmann	Leslie Mulder
Vladimir Stepanchenko	Robin Michielsen
Gerhard Mekenkamp	Oleg Gladyshev
Sander Raaijmakers	Rob Houtepen
Rob Alexander	Piotr Polak
Wojciech Bober	Ian Winterburn
Bas Driesen	Chris Brandson
Remi Vicente	Peter Chung
Mohammed Suhel	Purnima Dutt
Nikolai Paukov	Aleksandar Ostojic

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Tomasz Chyrowicz	Jordan Wills
Chris DeCenzo	Jean Ferraton
Sergio de Paula	Martin Sanchez Revuelta
Artem Dubrovskikh	Martin Hernandez Palomares
Phil Jamieson	

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table of contents

Revision history . . . 5
Participants . . . 7
Table of contents . . . 9
List of Figures . . . 15
List of Tables . . . 17
1. Introduction . . . 20
1.1. Purpose . . . 20
1.2. Conformance testing . . . 20
1.3. Conformance levels . . . 20
1.4. Conventions . . . 21
1.4.1. Document organization . . . 21
1.4.2. Number formats . . . 21
2. References[1] . . . 22
2.1. Connectivity Standards Alliance documents . . . 22
2.2. IEEE documents . . . 22
2.3. IETF documents . . . 22
2.4. Bluetooth SIG documents . . . 22
3. Definitions . . . 24
4. Acronyms and abbreviations . . . 26
5. Architectural Overview . . . 28
5.1. Introduction . . . 28
5.2. Overview . . . 28
5.3. Zigbee Dependencies . . . 29
5.4. Bluetooth Low Energy Dependencies . . . 29
6. Security . . . 32
6.1. Introduction . . . 32
6.2. Zigbee Direct Security . . . 32
6.2.1. State diagram . . . 33
6.2.2. Un-provisioned ZDD . . . 33
6.2.3. Provisioned ZDD . . . 34
6.3. Session Establishment . . . 34
6.3.1. Provisioning session . . . 34
6.3.2. Authorization session . . . 36
6.3.2.1. Authorization keys . . . 36
6.3.3. BLE secure session . . . 38
6.4. Cryptographic operations . . . 38
6.4.1. Encryption . . . 38

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

6.4.2. Validation and decryption . . . 38
6.4.3. CCM Nonce . . . 39
6.4.4. Session key . . . 39
6.4.5. Unique address . . . 39
6.4.6. Counter handling . . . 40
6.4.7. Exception handling . . . 41
6.4.7.1. Packet validation exceptions . . . 41
6.4.7.2. Counter overflow exception . . . 41
6.4.7.3. Encryption failed exception . . . 41
6.4.7.4. Insufficient authentication exception . . . 42
6.5. Zigbee Direct Security Service . . . 42
6.5.1. Zigbee Direct Security Service characteristics . . . 42
6.5.1.1. Authenticate . . . 42
6.5.1.2. Authenticate using ECDHE-PSK/P-256/SHA-256/HMAC-SHA-256-128 . . . 44
6.5.1.2.1. Session Establishment Overview . . . 45
6.5.1.2.2. Session Establishment Message 1 . . . 47
6.5.1.2.3. Session Establishment Message 2 . . . 47
6.5.1.2.4. Session Establishment Message 3 . . . 48
6.5.1.2.5. Session Establishment Message 4 . . . 49
6.5.1.3. Authentication SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128 . . . 49
6.5.1.3.1. Session Establishment Overview . . . 49
6.5.1.4. Authentication: Behavior and Exception handling . . . 51
6.5.2. Local TLVs . . . 51
6.5.2.1. Zigbee Direct Key Negotiation Method TLV (ID=0) . . . 51
6.5.2.2. Zigbee Direct Key Negotiation P-256 Public Point TLV (ID=1) . . . 52
6.5.2.3. Zigbee Direct Key Negotiation Curve25519 Public Point TLV (ID=2) . . . 52
6.5.2.4. Network Key Sequence Number TLV (ID=3) . . . 53
6.5.2.5. MacTag TLV (ID=4) . . . 53
6.6. ZVD Provisioning . . . 53
7. ZDD Specification . . . 54
7.1. Introduction . . . 54
7.2. Basic device behavior . . . 54
7.3. Security . . . 54
7.4. BLE network parameters . . . 54
7.5. ZDD specific Zigbee network parameters . . . 55
7.5.1. NwkMacInterface Parameters . . . 56
7.6. Discovery over BLE . . . 56
7.6.1. Introduction . . . 56
7.6.2. Basic discovery . . . 56

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.6.3. Zigbee Direct extension to the BLE Advertisement . . . 57
7.6.4. ZDD additional information . . . 59
7.7. BLE Services . . . 60
7.7.1. Introduction . . . 60
7.7.2. Zigbee Direct Commissioning Service . . . 60
7.7.2.1. Introduction . . . 60
7.7.2.2. Security . . . 62
7.7.2.3. Zigbee Direct Commissioning Service characteristics . . . 62
7.7.2.4. TLV Items . . . 64
7.7.2.4.1. General TLV processing . . . 64
7.7.2.4.2. Local TLV IDs for Commissioning . . . 64
7.7.2.4.3. Device Type TLV . . . 66
7.7.2.4.4. Joining Method TLV . . . 66
7.7.2.4.5. Network Status Map TLV . . . 67
7.7.2.4.6. Status Code TLV . . . 67
7.7.2.4.7. Link Key TLV . . . 69
7.7.2.5. Form Network characteristic . . . 70
7.7.2.5.1. Form Network Write . . . 70
7.7.2.6. Permit Joining characteristic . . . 72
7.7.2.7. Join Network characteristic . . . 73
7.7.2.7.1. Common TLV item usage in Join Network . . . 73
7.7.2.7.2. Joining Method: MAC association . . . 74
7.7.2.7.3. Joining Method: Rejoin . . . 76
7.7.2.7.4. Joining Method: Out-of-band Commissioning . . . 77
7.7.2.8. Leave Network characteristic . . . 78
7.7.2.9. Commissioning Status characteristic . . . 78
7.7.2.10. Manage Joiners characteristic . . . 80
7.7.2.10.1. Local TLV IDs for Manage Joiners characteristic . . . 81
7.7.2.10.2. Manage Joiners Command TLV . . . 81
7.7.2.10.3. Drop All Joiners . . . 81
7.7.2.10.4. Add Joiner’s Provisional Link Key . . . 82
7.7.2.10.5. Remove Joiner’s Provisional Link Key . . . 82
7.7.2.11. Identify characteristic . . . 82
7.7.2.12. Finding&Binding characteristic . . . 83
7.7.3. Zigbee Direct Tunnel Service . . . 84
7.7.3.1. Introduction . . . 84
7.7.3.2. Security . . . 84
7.7.3.3. Zigbee Direct Tunnel Service characteristics . . . 85
7.7.3.4. Zigbee Direct Tunnel Service NPDU Characteristic . . . 85

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.7.3.5. Payload format . . . 86
7.7.3.5.1. TLV Length Field . . . 87
7.7.3.5.2. TLV Value Field . . . 87
7.7.3.6. NPDU Message TLV . . . 87
7.7.3.6.1. Value Field Format . . . 87
7.7.3.6.2. Security . . . 88
7.7.4. Functional Description . . . 89
7.7.4.1. Effect on Receipt of an NLME-TRUSTEDLINK-POSTPROCESSING.request . . . 89
7.7.4.2. Effect on Receipt of an NLME-TRUSTEDLINK-PREPROCESSING.request . . . 89
7.7.4.3. Joining centralized security network . . . 89
7.7.4.4. ZVD operating as trust center on a centralized network . . . 91
7.7.4.5. Joining distributed security network . . . 92
7.7.4.6. ZVD Rejoining . . . 94
7.7.4.7. ZVD Forming a network . . . 95
7.7.4.8. Additional Requirements on a ZDD . . . 95
8. ZVD Specification . . . 96
8.1. Introduction . . . 96
8.2. BLE Network Parameters . . . 96
8.3. Functional description . . . 97
8.3.1. Discovery of ZDD . . . 98
8.3.2. Security . . . 98
8.3.3. Connection re-establishment mechanism . . . 99
8.4. Zigbee Direct BLE Service clients . . . 100
8.4.1. Introduction . . . 100
8.4.2. Commissioning Client . . . 100
8.4.2.1. Introduction . . . 100
8.4.2.2. Security . . . 100
8.4.2.3. Client Functional Description . . . 100
8.4.3. Tunnel Client . . . 100
8.4.3.1. Introduction . . . 100
8.4.3.2. Security . . . 101
8.4.3.3. Client Functional Description . . . 101
8.5. Extended Unique Identifier Allocation . . . 101
9. Network Key Rotation . . . 102
9.1. Limited Authorization Session . . . 102
10. Support for Legacy Networks . . . 104
10.1. Ephemeral Authorization . . . 105
10.1.1. Routing on behalf of the ZVD . . . 106
10.2. Joining flows for various network classes . . . 107

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

10.2.1. Class I: Distributed Security Networks . . . 107
10.2.2. Class II: Networks not utilizing install codes . . . 108
10.2.3. Class III: Networks utilizing install codes . . . 109
11. Additions to Zigbee Cluster Library Specification . . . 111
11.1. Introduction . . . 111
11.2. Additions to Chapter 3 - General . . . 111
11.3. Zigbee Direct Configuration Cluster . . . 111
11.3.1. Overview . . . 111
11.3.2. Revision History . . . 111
11.3.3. Classification . . . 111
11.3.4. Cluster Identifiers . . . 111
11.3.5. Server . . . 111
11.3.5.1. Security . . . 112
11.3.5.2. Attributes . . . 112
11.3.5.3. Zigbee Direct Information Attribute set . . . 112
11.3.5.3.1. InterfaceState . . . 112
11.3.5.3.2. AnonymousJoinTimeout . . . 112
11.3.5.4. Commands . . . 113
11.3.5.4.1. Commands Received . . . 113
11.3.5.4.2. Generic Usage Notes . . . 113
11.3.5.4.3. Configure Zigbee Direct Interface Command . . . 113
11.3.5.4.4. Configure Zigbee Direct Anonymous Join Timeout Command . . . 115
11.3.5.5. Commands Generated . . . 116
11.3.5.5.1. Configure Zigbee Direct Interface Response Command . . . 116
11.3.6. Client . . . 117
Annex A: BLE Connectable Advertisement example . . . 119
Annex B: Security Test Vectors . . . 121
B.1. Key Agreement using ECDHE-PSK/P-256/SHA256/HMAC-SHA256-128 . . . 121
B.1.1. Round 1: Message 1 from ZVD to ZDD . . . 121
B.1.2. Round 1: Message 2 from the ZDD to the ZVD . . . 122
B.1.3. Round 2: Message 3 from ZVD to ZDD . . . 123
B.1.4. Round 2: Message 4 from the ZDD to the ZVD . . . 124
B.2. Key Agreement using SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128 . . . 124
B.2.1. Round 1: Message 1 from ZVD to ZDD . . . 125
B.2.2. Round 1: Message 2 from the ZDD to the ZVD . . . 125
B.2.3. Round 2: Message 3 from ZVD to ZDD . . . 127
B.2.4. Round 2: Message 4 from the ZDD to the ZVD . . . 127
B.3. Deriving authorization keys on a ZDD or TC . . . 128
B.3.1. Basic Authorization Key . . . 128

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

B.3.2. Administrative Authorization Key . . . 128
B.4. Accessing secure characteristics via AES-CCM-128 . . . 129
Annex C: Zigbee Direct Component Specifications . . . 131
C.1. Zigbee Direct Commissioning ZVD Component Specification [2] . . . 132
C.1.1. Testing and Certification . . . 132
C.1.2. Applicable Supported Operating Environments . . . 132
C.1.3. Next Higher Layer Component Interface Specification . . . 132
C.1.4. Lower Component Interface . . . 133
C.2. Application Framework Component Specification[3] . . . 133
C.2.1. Testing and Certification . . . 133
C.2.2. Applicable Supported Operating Environments . . . 134
C.2.3. Next Higher Layer Component Interface Specification . . . 134
C.2.4. Lower Component Interface . . . 134
C.3. Zigbee Stack Component Specification . . . 134
C.3.1. Testing and Certification . . . 135
C.3.2. Applicable Supported Operating Environments . . . 135
C.3.3. Next Higher Layer Component Interface Specification . . . 135
C.3.4. Lower Component Interface . . . 135
C.4. ZVD-TS Component Specification . . . 135
C.4.1. Testing and Certification . . . 136
C.4.2. Applicable Supported Operating Environments . . . 136
C.4.3. Next Higher Layer Component Interface Specification . . . 136
C.4.4. Lower Component Interface . . . 136

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

List of Figures

Figure 1	Typical Zigbee Direct Use-case	20
Figure 2	Protocol Stack Overview	29
Figure 3	Overview of security interactions between the ZDD and the ZVD	32
Figure 4	Security state diagram	33
Figure 5	Establishing provisioning session	35
Figure 6	Establishing authorization session	36
Figure 7	Payload format of BLE frames	38
Figure 8	Input to the CCM encryption operation	38
Figure 9	Input to the CCM validation and decryption operation	39
Figure 10	ECDHE PSK Session Establishment Sequence Diagram	46
Figure 11	SPEKE Session Establishment Sequence Diagram	50
Figure 12	Commissioning use case TC ZDD	80
Figure 13	Identify use case	82
Figure 14	ZVD ZDD message flow sequence	86
Figure 15	ZVD Joining Centralized Security Network	90
Figure 16	ZVD Operating as Trust Center on a Centralized Security Network	91
Figure 17	ZVD joining distributed security network	93
Figure 18	Zigbee Direct Virtual Device (ZVD) architecture	97
Figure 19	Legacy distributed security networks	107
Figure 20	Legacy networks not utilizing install codes	108
Figure 21	Legacy networks utilizing install codes	109

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

List of Tables

Table 1	Pre-shared Key IDs	35
Table 2	Authorization key levels	37
Table 3	CCM Nonce fields	39
Table 4	Unique address fields	40
Table 5	Zigbee Direct Security Service characteristics	42
Table 6	Payload of Session Establishment message	43
Table 7	Parameters for ECDHE using P-256	44
Table 8	PSK Items	44
Table 9	Session Establishment Message 1	47
Table 10	Session Establishment Message 2 for a factory new device	48
Table 11	Session Establishment Message 2 for a provisioned device	48
Table 12	Session Establishment Message 3	48
Table 13	Session Establishment Message 4	49
Table 14	Selected Key Negotiation Method	51
Table 15	Selected Key Negotiation Method Enumeration	51
Table 16	Selected Pre-shared Secret Enumeration	52
Table 17	P-256 Public Point TLV Format	52
Table 18	Fields of the Zigbee Direct Key Negotiation P-256 Public Point TLV	52
Table 19	Format of the Zigbee Direct Key Negotiation Curve25519 Public Point TLV	52
Table 20	Fields of the Zigbee Direct Key Negotiation Curve25519 Public Point TLV	53
Table 21	Format of the Network Key Sequence Number TLV	53
Table 22	Format of the MacTag TLV	53
Table 23	Recommended Bluetooth Low-Energy parameters	55
Table 24	Zigbee Direct extension to the BLE Advertisement	58
Table 25	Some UUIDs of BLE Device Information Characteristics	60
Table 26	Zigbee Direct Commissioning Service characteristics	62
Table 27	Zigbee Direct Commissioning Service TLV IDs	64

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 28	Device Type TLV values	66
Table 29	Joining Method TLV values	66
Table 30	Network Status Map TLV values	67
Table 31	Format of the Zigbee Direct Status Code TLV	67
Table 32	Fields of the Zigbee Direct Status Code TLV	68
Table 33	Status Code TLV Domain/Operation values	68
Table 34	Format of the Zigbee Direct Link-Key TLV	69
Table 35	Zigbee Direct APS Link-Key TLV Flags	69
Table 36	TLV usage in Form Network characteristic	71
Table 37	Permit Joining characteristic values	72
Table 38	Joining Method: MAC association ZDD behavior	75
Table 39	TLV usage for Joining Method: MAC association	75
Table 40	TLV usage for Joining Method: Rejoin	76
Table 41	TLV usage for Joining Method: Out-of-band Commissioning	78
Table 42	Leave Network characteristic payload	78
Table 43	TLV usage for Commissioning Status	79
Table 44	Zigbee Direct Manage Joiners TLV IDs	81
Table 45	Manage Joiners Command TLV values	81
Table 46	TLV usage for Manage Joiners commands	81
Table 47	Identify characteristic write values	83
Table 48	Identify characteristic read values	83
Table 49	Finding&Binding characteristic payload	83
Table 50	Zigbee Direct Tunnel Service characteristics	85
Table 51	Zigbee Direct Tunnel Service NPDU characteristic format	86
Table 52	ZDTS NPDU characteristic Type IDs	87
Table 53	Length field	87
Table 54	Value field	87
Table 55	Sub fields of NPDU Message TLV	88
Table 56	Zigbee Direct Attribute Sets	112
Table 57	Zigbee Direct Information Attribute set	112
Table 58	Zigbee Direct Configuration Cluster Received Command IDs	113

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 59	Format of the Configure Zigbee Direct Interface Command	114
Table 60	Values of the InterfaceState Field of the Configure Zigbee Direct Interface Command	114
Table 61	Format of the Configure Zigbee Direct Anonymous Join Timeout Command	115
Table 62	Zigbee Direct Configuration Cluster Generated Command IDs	116
Table 63	Payload Format	116

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

1. Introduction

1.1. Purpose

This document is the technical specification for Zigbee devices implementing the Zigbee Direct feature. The intent of Zigbee Direct is to allow for encrypted and authenticated access to Zigbee devices and networks from Bluetooth Low Energy ("BLE") capable devices. This enables devices like smartphones, tablets, smart speakers and hubs to commission and control devices on a Zigbee network.

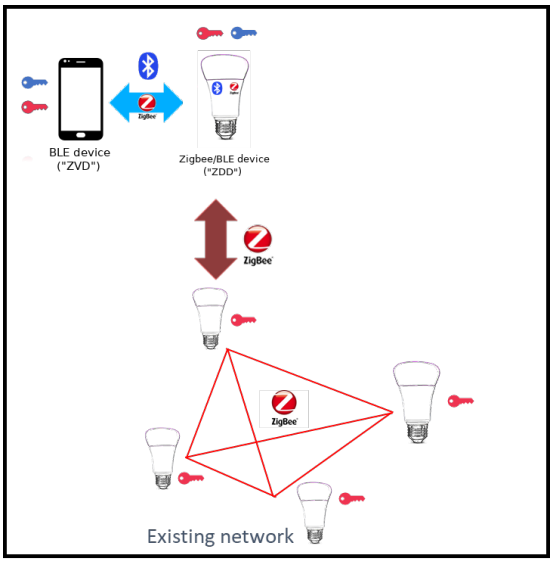


Figure 1. Typical Zigbee Direct Use-case

This document defines the corresponding Zigbee Direct services: Security Service, Tunnel Service and Commissioning Service. New device types/roles have been defined as well to address those use cases. This document is based on requirements captured in the Technical Requirements Document 19-75233-002 ([R0]).

Backward compatibility to existing Zigbee devices and networks was one of the design principles.

1.2. Conformance testing

Implementations required to demonstrate conformity to this specification SHALL follow the Zigbee Direct Test Specification [R2].

1.3. Conformance levels

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED" and "MAY" in this document are to be interpreted as described in [R1].

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

1.4. Conventions

1.4.1. Document organization

The specification below has been split into different categories: Zigbee Direct Device (ZDD) and Zigbee Virtual Device (ZVD), each subdivided into sub categories. The specification refers to other specifications as needed for implementing the base device, particularly the Zigbee PRO Core and BDB specifications. Where BLE specific behavior is concerned, or where said specifications do not offer clarity on the subject, this specification provides all the relevant details.

1.4.2. Number formats

In this specification hexadecimal numbers are prefixed with the designation "0x" and binary numbers are prefixed with the designation "0b". All other numbers are assumed to be decimal unless indicated otherwise within the associated text.

Binary numbers are specified as successive groups of 4 bits, separated by a space (" ") character from the most significant bit (next to the 0b prefix and left most on the page) to the least significant bit (rightmost on the page), e.g. the binary number 0b0000 1111 represents the decimal number 15. Where individual bits are indicated (e.g. bit 3) the bit numbers are relative to the least significant bit (i.e. bit 0).

When a bit is specified as having a value of either 0 or 1 it is specified with an "x", e.g. "0b0000 0xxx" indicates that the lower 3 bits can take any value but the upper 5 bits must each be set to 0.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

2. References[1]

2.1. Connectivity Standards Alliance documents

[R0] Zigbee 3.0/BLE Proxying Technical Requirements Document (TRD) 19-75233-002

[R2] Zigbee Direct Test Specification 20-27538-22

[R4] Zigbee Core Specification 05-3474-22

[R12] Zigbee Core Specification r23.1 05-3474-23.1

[R8] Zigbee Cluster Library Specification 07-5123-08

[R9] CSA certification Policy 07-4842-13

[R10] Base-Device-Behavior-Specification v 3.0.1 16-02828-011

[R11] Certification Policy 07-4842-20

2.2. IEEE documents

[R7] Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID)

2.3. IETF documents

[R1] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF 236 RFC 2119, March 1997.

2.4. Bluetooth SIG documents

[R3] Bluetooth Core Specification Version 4.2, CS 4.2, Bluetooth SIG, Dec 2014

[R5] Supplement to the Bluetooth Core Specification, Revision v9, Bluetooth SIG, Dec 2019

[R6] 16-bit UUID Numbers Document, Bluetooth SIG, Nov 2020

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

3. Definitions

Zigbee Direct Device (ZDD):

A Zigbee device (e.g. End Device, Router, Coordinator, Trust Center) capable of concurrently running Zigbee and BLE stacks, implementing functionality which enables the exchange of data between certain BLE-based devices (ZVDs) and a Zigbee device or network.

Zigbee Direct Virtual Device (ZVD):

A BLE device performing a certain role (e.g. Trust Center, commissioning tool) or adopting a certain device type on a Zigbee network (e.g. End Device, Router, Coordinator, Trust Center) in conjunction with a ZDD. The software stack of a ZVD may be constituted of the ZVD application, Zigbee PRO networking stack, BDB and ZCL software components, and BLE stack with appropriate BLE services.

Zigbee Direct Tunnel Service:

A BLE GATT service as exposed by a ZDD that allows a ZVD to send and receive data to and from a Zigbee network as required by the role the ZVD is performing.

Zigbee Direct Commissioning Service:

A BLE GATT service as exposed by a ZDD that enables a ZVD to commission a device onto the Zigbee network.

Maintenance:

Allows a ZVD to perform basic diagnostics (maintenance) on the Zigbee network by means of receiving status of the ZDD on the Zigbee network. Maintenance may be items such as obtaining status of the 802.15.4 interface, obtaining status of the Zigbee network state, interrogating performance counters, or triggering operations that may repair problems such as a Network Rejoin.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

4. Acronyms and abbreviations

APS Application support sub-layer

BDB Base Device Behaviour

BLE Bluetooth Low Energy

ECDHE Elliptic Curve Diffie-Hellman

GATT Generic Attribute Profile

IC Install Code

ID Identifier

IEEE Institute of Electrical and Electronic Engineers

MAC Medium access control

MIC Message Integrity Code

MTU Maximum Transmission Unit

NWK Network

OOB Out-of-band

PAN Personal area network

PSK Pre-shared key

PHY Physical Layer

TC Trust Center

TCLK Trust Center Link Key

UUID Universally Unique Identifier

ZC Zigbee coordinator

ZCL Zigbee cluster library

ZD Zigbee Direct

ZDD Zigbee Direct Device

ZDO Zigbee device objects

ZDTS Zigbee Direct Tunnel Service

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

ZED Zigbee end device

ZR Zigbee router

ZVD Zigbee Virtual Device

ZVD-TS Zigbee Virtual Device implementing the client side of the Zigbee Direct Tunnel Service

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

5. Architectural Overview

5.1. Introduction

This chapter gives an overview of the architecture of Zigbee Direct and describes general elements that apply to any device that supports the feature.

5.2. Overview

A Zigbee Direct based system holds two kinds of devices: Zigbee Direct Device (ZDD) and a Zigbee Virtual Device (ZVD). The ZDD is characterized by having both an IEEE 802.15.4 radio and a Bluetooth Low Energy (BLE) radio, and running BLE in a Peripheral role. The ZVD is characterized by having at least a BLE radio and running BLE in a Central role. In practice, a ZDD is typically a regular Zigbee device, and a ZVD is typically a smart phone or a smart speaker. Note that it is not per se excluded that the ZVD has an IEEE 802.15.4 radio, but it is not very likely in practice.

There are two main use-cases where Zigbee Direct plays a role:

- a ZVD is used to commission a ZDD, using the Zigbee Direct Commissioning Service, on its intended Zigbee network and does not need to interact with Zigbee devices on that network, and
- a ZVD is part of a Zigbee network and is able to interact with Zigbee devices on that network through a Zigbee Direct Tunnel Service provided by the ZDD.

In the first use-case, the Zigbee Direct tunnel functionality (and therefore having the need for a Zigbee stack) MAY be supported.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

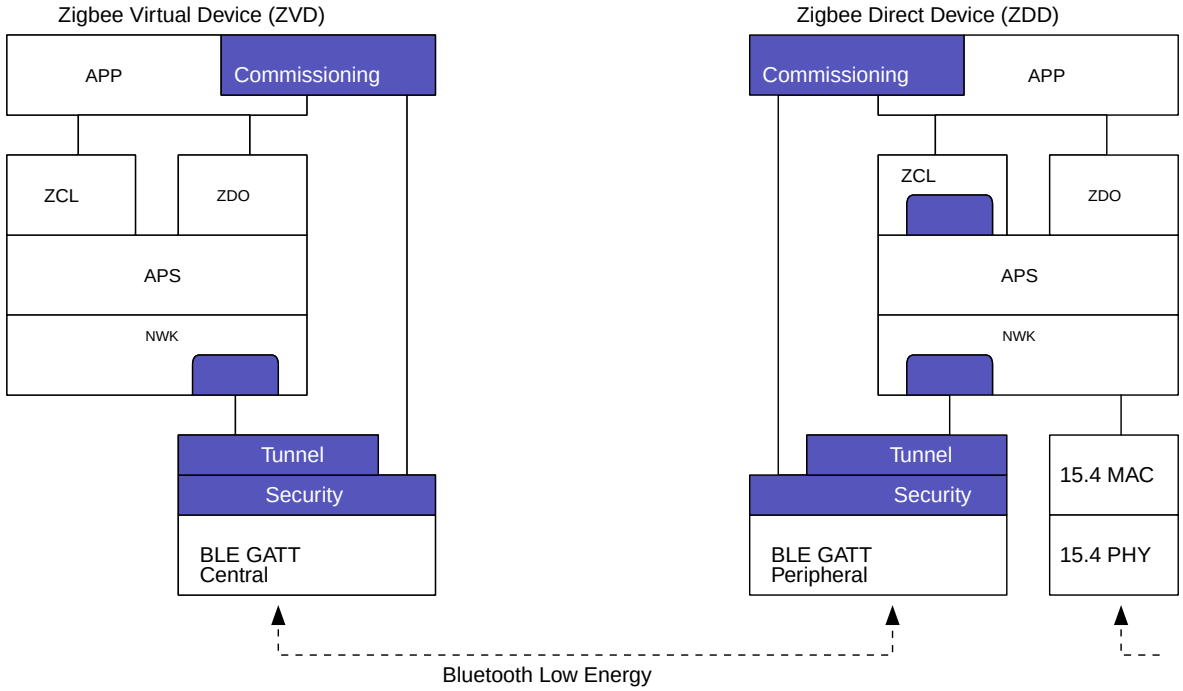


Figure 2. Protocol Stack Overview

Figure 2 gives an overview of the protocol stacks of both a ZDD and a ZVD. The areas highlighted in blue indicate new technical items in scope of this specification.

These include:

- Zigbee Direct Commissioning - Used to commission a ZDD onto the Zigbee network via BLE.
- Zigbee Direct Tunnel - Serves as a Trusted Link to transfer Zigbee network packets over BLE.
- Zigbee Direct Security - Used to protect data transferred over BLE.
- Zigbee Direct Cluster - Used to set the behavior of Zigbee Direct in the ZDD.
- Zigbee Direct NWK Layer Enhancements - Specific Zigbee Direct additions such as an additional MAC interface description (see section 7.5) .

The specification also defines the portion of the BLE advertisement that is necessary for correct Zigbee Direct device discovery and operation.

5.3. Zigbee Dependencies

A ZDD SHALL support a Zigbee stack. A ZVD MAY support a Zigbee stack. When a Zigbee stack is supported, it SHALL be based on revision 23.1 of the Zigbee Specification [R12].

5.4. Bluetooth Low Energy Dependencies

Both a ZDD and a ZVD SHALL support Bluetooth Low Energy with at least version 4.0, as certified by the Bluetooth SIG. The following features SHALL be supported:

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

- Bluetooth Low Energy

The following features SHOULD be supported:

- LE Data Packet Length Extension ("Data Length Extension", see [R3] Vol 6, Part B, Section 4.6)

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

6. Security

6.1. Introduction

This chapter describes the general security of Zigbee Direct. Any extensions or changes to the security mechanisms described of previous versions of the Zigbee specification ([R4]) are included in [R12]. Specific security extensions that apply only to specific device types or services are described in their respective sections.

6.2. Zigbee Direct Security

Figure 3 gives an overview of the interactions between the ZDD and the ZVD.

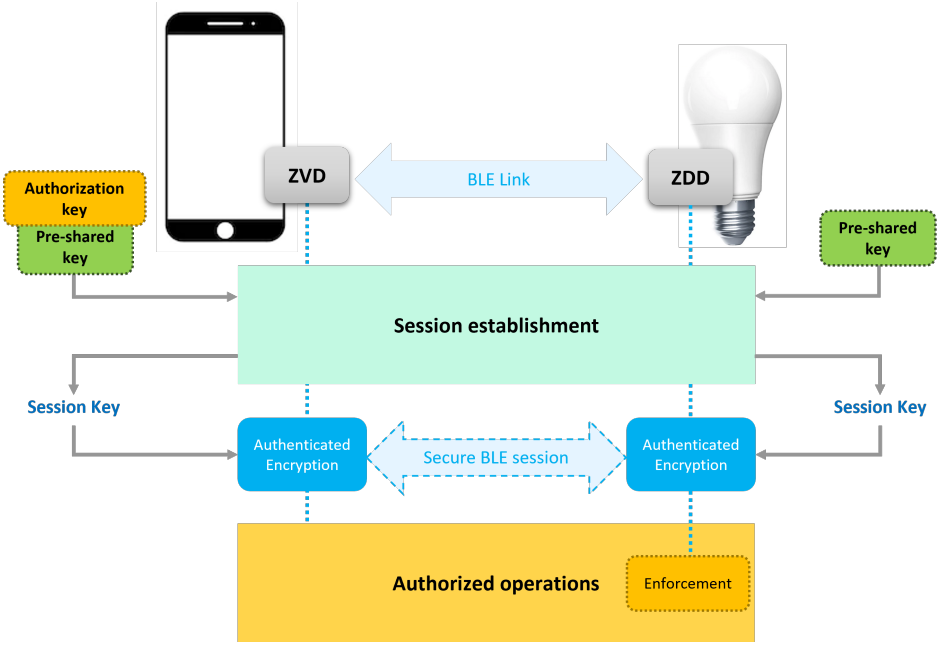


Figure 3. Overview of security interactions between the ZDD and the ZVD

A ZVD and a ZDD establish a BLE connection with security mode 1 and security level 1 (no security). Next, a secure session handshake is performed using a shared credential section 6.3.2.1. The resulting session key is used to establish a secure BLE session between the ZVD and the ZDD.

It is required for a ZVD to be authorized to join a Zigbee network via a ZDD. The authorization is implemented via a shared key that is used as a credential to establish the secure BLE session.

Zigbee Direct allows a ZVD to establish a secure BLE session with Basic or Admin authorization level. The level of access for a ZVD, defined by the authorization key used, is enforced by a ZDD.

The Zigbee Direct Security solution does not rely on Bluetooth Low Energy specific security, and as such does not mandate any specific pairing or bonding methods as defined in [R3] Vol 3, Part H. ZVDs and ZDDs SHOULD avoid (direct or indirect) triggering of Bluetooth Low Energy device pairing. It is recommended for a ZDD to not return a BLE ATT Error Response with the error code

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

<<Insufficient Encryption>> (as defined in [R3] Vol 3, Part F, Section 3), as this results in a Bluetooth link layer pairing request on some systems (such as iOS and Android).

6.2.1. State diagram

The Figure 4 shows a prototypical ZDD state diagram for provisioning.

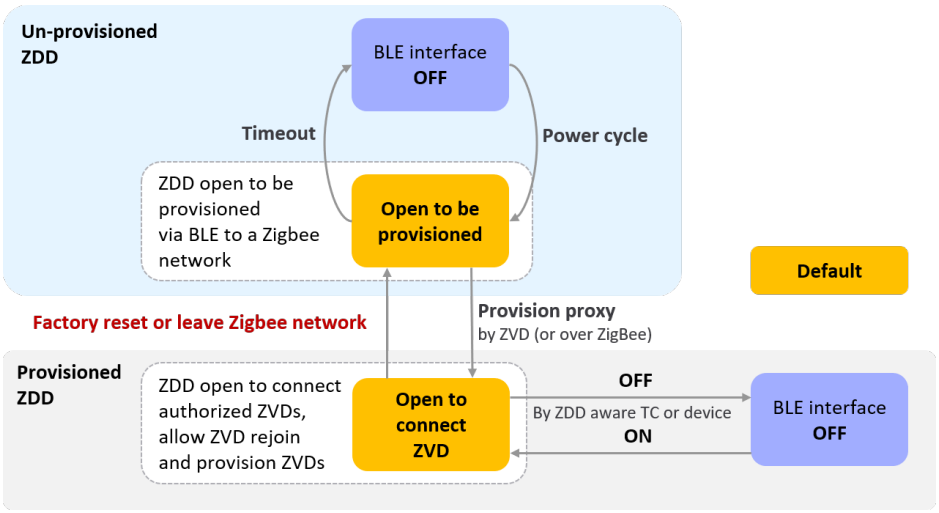


Figure 4. Security state diagram

6.2.2. Un-provisioned ZDD

In un-provisioned (“factory new”) state the BLE interface of a ZDD SHALL BE open to allow for provisioning using a ZVD; referred to as **Open to be provisioned** state. In this state the ZDD will send the connectable advertisement outlined in section section 7.6.3.

Any ZVD can claim the ownership of the ZDD and provision it into a Zigbee network by establishing a provisioning session, see section 6.3.1. The provisioning session allows the ZVD to configure the ZDD for operation on the Zigbee network, including an option to securely transfer all required network parameters and keys to the ZDD, see section 7.7.2. The provisioning session established to provision the ZDD remains open once the ZDD has been configured for operation on a Zigbee network granting the ZVD with access to the network.

If, after a timeout of *NewZddProvisioningTimeout* seconds, there is no provisioning session established, the ZDD SHALL transition into the (Zigbee Direct) **BLE interface OFF** state. *NewZddProvisioningTimeout* is the interval during which it is possible to connect to a ZDD which has not been provisioned to a Zigbee Network (e.g. is factory new) in seconds. A manufacturer defined value for *NewZddProvisioningTimeout* SHALL be no smaller than 60 seconds. In this state the connectable advertisement of Zigbee Direct SHALL be disabled. If there are manufacturer specific extensions advertised, advertisements MAY still be sent, but the Zigbee Direct specific advertisement data (section 7.6.3) SHALL be removed from the advertisement and the advertisement frequency SHOULD be set to the lowest possible value for the device. The ZDD can transition to **Open to be provisioned** state again by power cycling, or any other manufacturer defined user interaction.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

6.2.3. Provisioned ZDD

Once on a Zigbee network, the BLE interface of the ZDD goes into the **Open to connect ZVD** state that allows any authorized ZVD to access the Zigbee network via the ZDD if the Zigbee network has distributed security or if the Zigbee network has centralized security and the Trust Center is Zigbee Direct aware. This state is also enabling a new ZVD to be provisioned over the BLE interface with basic authorization or an authorized ZVD to rejoin the Zigbee network.

Once the ZDD is active on a Zigbee network, to check whether the centralized security Zigbee network is Zigbee Direct aware, it SHALL follow the following procedure regardless of how the ZDD became active on the Zigbee network:

- Discover the Zigbee Direct Configuration client cluster (section 11.3.6) at all endpoints of the Trust Center
- If the Zigbee Direct Configuration client cluster exists, the Zigbee network is Zigbee Direct aware. No further steps need to be processed
- If the Zigbee Direct Configuration client cluster does not exist, a new ZVD SHALL be provisioned by ZDD as described in section 10.

A TC or another device implementing the Zigbee Direct Configuration cluster section 11.3.5.4.3 may turn the Zigbee Direct BLE Interface OFF or ON bringing it in and out of state (Zigbee Direct) **BLE Interface OFF.**

A ZVD MAY check whether the centralized security network is Zigbee Direct aware using the procedure as indicated for the ZDD, by discovering the Zigbee Direct Configuration client cluster (section 11.3.6) at all endpoints of the Trust Center.

Any action making the ZDD leave the Zigbee network SHALL bring the ZDD into the **Open to be provisioned** state.

6.3. Session Establishment

The BLE session key is derived using one of the two key agreement algorithms used by the mechanism to negotiate a dynamic link key as described in [R12]:

- ECDHE-PSK/P-256/SHA-256/HMAC-SHA-256-128, Refer to section 6.5.1.2 and [R12]
- SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128, see Section "ECDHE/SPEKE using Curve25519" in Annex J "Cryptographic Processing for ECDHE" in [R12] and also annex C.7 of [R12]

The key agreement algorithms are implemented as two Authenticate characteristics of the Zigbee Direct Security Service, see section 6.5.

6.3.1. Provisioning session

A provisioning session is established using one of the following pre-shared keys (PSK) as

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

described in [R12], Annex J “Cryptographic Processing for ECDHE”:

Table 1. Pre-shared Key IDs

PSK ID	PSK
n/a	"ZigbeeAlliance18" (Anonymous Key Exchange)
0x00	Installation Code
0x01	Variable-length Passcode

The "ZigbeeAlliance18" key SHOULD only be utilized in case neither Install Code nor Variable-length Passcode of the ZDD can be provided out of band to the ZVD.

The "ZigbeeAlliance18" and Installation Code keys SHALL be supported by all ZDD implementations. The Variable-length Passcode is optional and SHALL only be used for the ZDDs implementing SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128 exchange.

It is up to the ZVD initiating the provisioning session to select the PSK used to establish the session. The selected PSK ID is indicated to the ZDD in the first message of the session establishment.

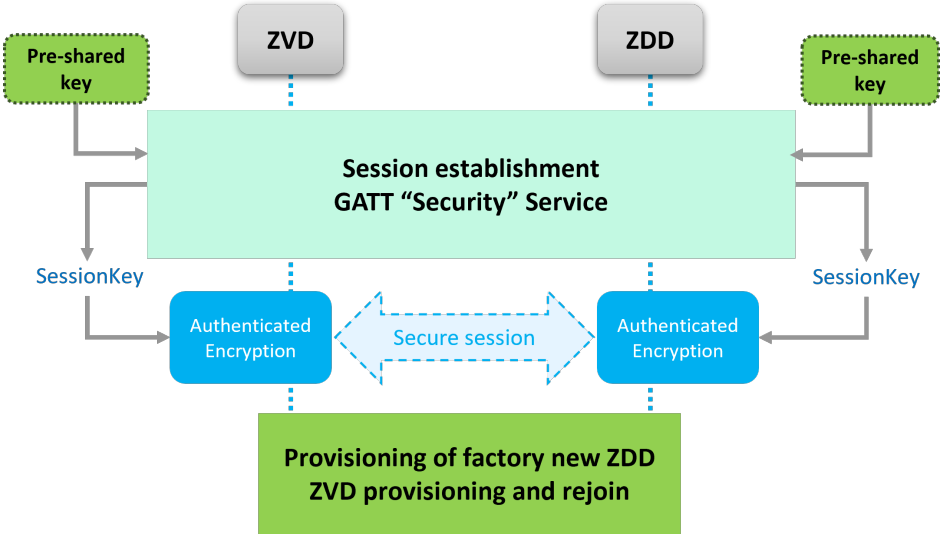


Figure 5. Establishing provisioning session

The successful establishment of provisioning session enables the ZVD to:

- Get access to the Zigbee Direct Commissioning Service of an un-provisioned ZDD, see section 7.7.2 for more details. The Commissioning Service of a provisioned ZDD SHALL only be accessible to a ZVD that has been provisioned with the Admin authorization key, see section 6.3.2.1, except for a Read Request to the Commissioning Status characteristic that SHALL be accessible using a provisioning session, the Basic Key or the Admin Key [4].
- Request an authorization key to access the network, see section 6.6 for more details.

Two authorization levels can be granted to the ZVD as a result of successful provisioning session establishment:

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

- **Admin access** to the Commisioning Service of a factory new ZDD
- **ZVD provisioning access** to the ZDD that is connected to a Zigbee network allowing ZVD provisioning

When a ZDD is connected to a Zigbee network allowing ZVD provisioning and the ZVD tries to establish a provisioning session using the Anonymous Well-Known Secret, the ZDD SHALL verify the value of the Anonymous Join Countdown Timer and SHALL only allow the provisioning session when the countdown timer is set to a value higher than 0 seconds. See section 6.6 for more details. The ZDD Anonymous Join Countdown Timer can be configured using the Zigbee Direct Configuration cluster section 11.3.

The provisioning session SHALL NOT provide the ZVD with access to the Zigbee network. To be granted with access to the Zigbee network the ZVD SHALL establish a new secure authorization session using either the basic authorization key or the admin authorization key as described in section 6.3.2.

The ZDD SHALL only allow a provisioning session to be establish with a ZVD if there is no active authorization session between both devices.

6.3.2. Authorization session

A ZVD that is authorized to connect to a ZDD establishes an authorization session using authorization keys, see Figure 6.

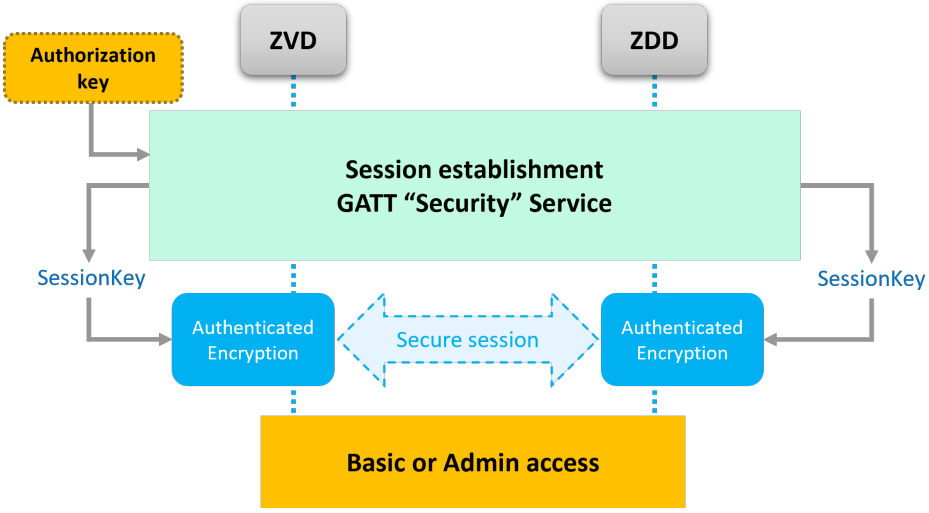


Figure 6. Establishing authorization session

6.3.2.1. Authorization keys

Two authorization levels are defined: the Basic level and the Admin level. The levels are enforced using two shared keys as defined in Table 2.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 2. Authorization key levels

Key	Bits	Derivation formula	Description
Basic key	128	• *KDF(s,{0x03}) where s = H(ZVD IEEE || Zigbee NWK key)*	**Basic authorization** key. The Basic key grants the ZVD with access to the network.
Admin key	128	• *KDF(s,{0x04}) where s = H(ZVD IEEE || ZDD TC Link key)* • The key is generated by a ZVD and provided to a ZDD over a secure provisioning session	**Admin authorization** key. The Admin key grants the ZVD with access to the Commissioning Service of a provisioned ZDD and access to the network.

The keys can be derived by the ZDD as well as the Zigbee network's TC. Furthermore they can be derived by a ZVD which is either provisioning a ZDD to an existing network, or establishing a new network and setting the network and ZDD link keys. To derive the keys the ZVD's IEEE address (discovered during secure BLE session establishment) and corresponding Zigbee network keys are concatenated together and hashed using H(x) hashing function based on AES-MMO as defined in [R12]: Section "ECDHE/SPEKE using Curve25519" in Annex J "Cryptographic Processing for ECDHE". The output of the hashing function is used as shared secret input to the KDF based on HMAC-AES-MMO as defined in [R12]: Section "ECDHE/SPEKE using Curve25519" in Annex J "Cryptographic Processing for ECDHE".

The Zigbee NWK key is used to derive the Basic key and the ZDD's TC Link key is used to derive the Admin key. The TC Link key may be provided by a ZVD provisioning a ZDD to a network using the Commissioning Service, see section 7.7.2. If not provided by the ZVD, the preprogrammed link key is used if available, otherwise the default keys are used as described in section 7.7.2.7.

The Admin key can be generated by a ZVD and provided to a ZDD with the Zigbee network parameters over a secure BLE provisioning session using the Commissioning Service, see section 7.7.2. Only a ZVD provisioning a ZDD onto a Zigbee network can provision the Admin key. If provided during ZDD provisioning, the Admin key is stored by the ZDD and SHALL be used by any ZVD operation requiring admin access. The Admin key can be a single key for all ZDDs provisioned to a network or an individual key per ZDD. It is a choice of a commissioner provisioning the ZDD to a network. If the Admin key is not provided during the ZDD provisioning, the key is derived by the ZDD participating in a centralized security network using the active TCLK of the ZDD and the IEEE address of the ZVD. There is no Admin access possible if no Admin key is provided during provisioning of a ZDD to a distributed security network.

To change the Admin key for a ZDD participating in a Zigbee network the ZDD SHALL be factory reset first and re-provisioned to the network with a new key.

The TCLK can also be provided during ZDD provisioning as a separate parameter (if applicable) and does not influence the Admin key if the key is also provided.

The authorization key can be provisioned into a ZVD via an out of band method (Basic or Admin authorization key) or via the BLE interface (Basic authorization key only), see section 6.6.

The Zigbee NWK key SHALL never be exposed by a ZDD to a ZVD over BLE session regardless of

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

authorization key used by the ZVD to establish the session.

6.3.3. BLE secure session

The payload format of BLE frames exchanged over the BLE link between ZVD and ZDD is shown in Figure 7.

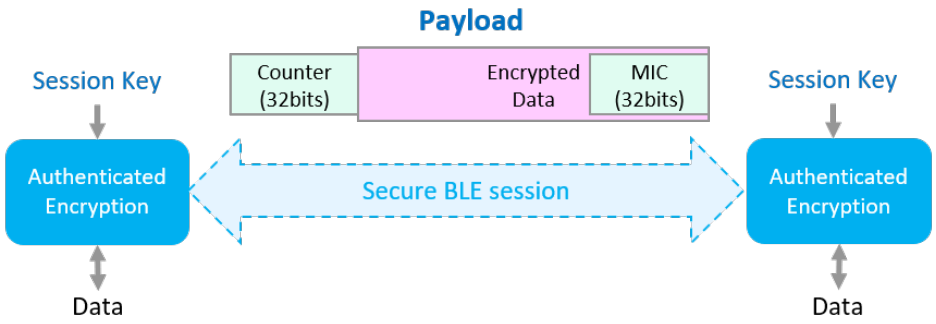


Figure 7. Payload format of BLE frames

6.4. Cryptographic operations

6.4.1. Encryption

The Figure 8 shows the input to the CCM encryption operation.

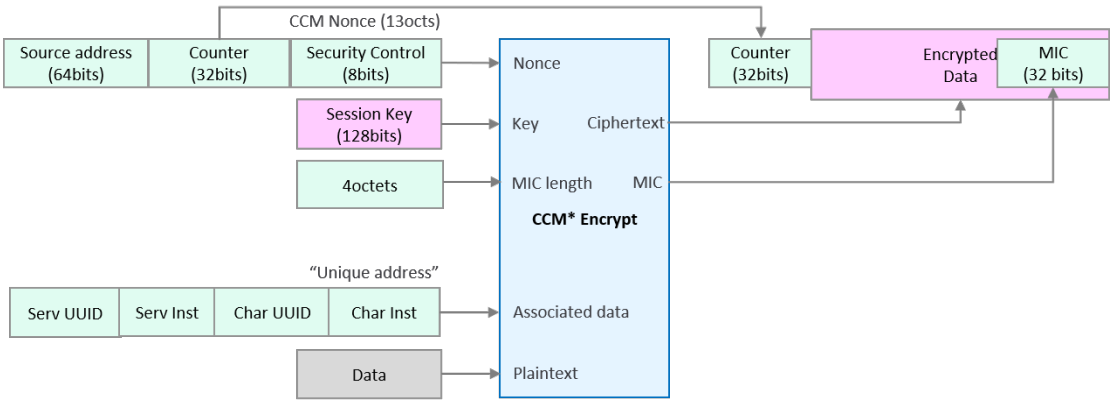


Figure 8. Input to the CCM encryption operation

6.4.2. Validation and decryption

The Figure 9 shows the input to the CCM validation and decryption operation.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

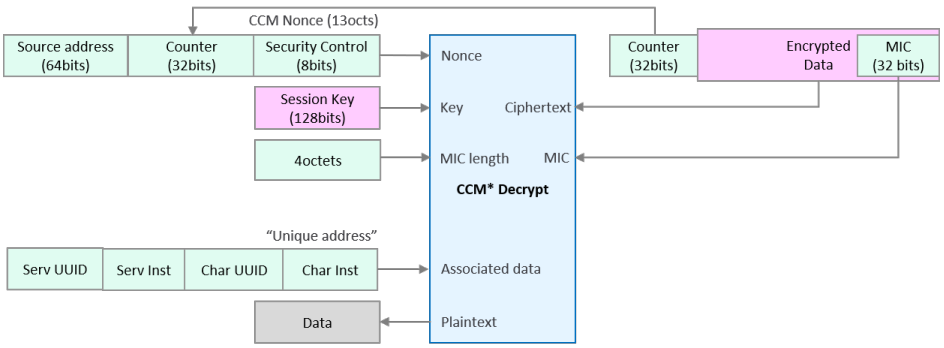


Figure 9. Input to the CCM validation and decryption operation

6.4.3. CCM Nonce

The nonce that is provided as input to the "CCM Encrypt" operation is 104 bits long and consists of a concatenation of the fields that described in Table 3.

CCM Nonce = Source address || Counter || Security Control

Table 3. CCM Nonce fields

Field	Bits	Description
Source address	64	IEEE address of the source device in little-endian representation
Counter	32	Counter value in little-endian representation, see section 6.4.6 for detailed description
Security Control	8	The 3 least significant bits (bits 0..2) are the 'Security Level Indicator' as defined by Zigbee specification [R12] indicating security level 5 (encrypted, 32bit MIC), the 5 most significant bits (bits 3..7) are 'Reserved' (0)

6.4.4. Session key

The session key is 128 bits long and is provided as the result of the session establishment, see section 6.3.

6.4.5. Unique address

To protect against swapping of characteristics (swapping of GATT database's ATT handles) a characteristic's unique address is provided as associated data to the CCM encrypt and decrypt operations.
This unique address is 34 bytes long and consists of the fields that are listed in Table 4

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 4. Unique address fields

Field	Octets	Description
Service UUID	0..15	128-bit UUID of the service that includes the characteristic, see RFC4122 section 4.1.2 Layout and Byte order In case the service uses a 16-bit UUID, then that 16-bit UUID will be mapped to a 128-bit UUID via the Bluetooth Base UUID.[2]
Service Instance	16	Instance ID of the GATT service. In this revision of the specification the instance ID SHALL be 0.
Characteristic UUID	17..32	128-bit UUID of the characteristic, see RFC4122 section 4.1.2 Layout and Byte order In case the characteristic uses a 16-bit UUID, then that 16-bit UUID will be mapped to a 128-bit UUID via the Bluetooth Base UUID.[5]
Characteristic Instance	33	Instance ID of the GATT characteristic. In this revision of the specification the instance ID SHALL be 0.

6.4.6. Counter handling

Each communication side SHALL keep two 32-bit counters per BLE connection: “incoming counter” and “outgoing counter”. Both counters are set to 0 after a new BLE secure session is established (since each session creates a fresh session key).

Sender device SHALL:

- Increment its outgoing counter before it composes the nonce for the "CCM Encrypt" operation.
- Use this newly incremented outgoing counter for encryption of the payload and include that counter in the packet that is composed for sending (*Counter* field of message to be transmitted SHALL be equal to outgoing counter).

Receiver device SHALL:

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

- Validate message freshness by testing that the *Counter* field from the received message is greater than the previously stored incoming counter.
- If the received message passes freshness validation, continue with checking the authenticity of the message.
- If the received message passes authenticity validation (the MIC of the received message is validated), process (decrypt) the encrypted data of the received message.
- If packet freshness or authenticity validations fail, then the receiver SHALL ignore the received message and the receiver SHALL NOT update the incoming counter.
- If the received message passes freshness and authenticity validation, then the receiver SHALL update its incoming counter by setting it to the value of the Counter field from the received message.

6.4.7. Exception handling

6.4.7.1. Packet validation exceptions

Packet validation SHALL fail in the following cases:

- Counter from the received packet is smaller than or equal to the receiver's stored incoming counter (counter from the last received message that was successfully validated).
- CCM validation fails: calculated MIC does not match the MIC in the received packet.

In case a validation exception occurs, the receiver SHALL ignore the received packet. This includes not updating the incoming counter.
In case the received message was sent via BLE ATT Write Request, then (next to ignoring the received packet) the receiver SHALL respond with a BLE ATT Error Response with the error code 0x80 (*Zigbee Direct Application Error*), (see [R3] Vol 3, Part F, Section 3).

With the remote end seemingly running into continuous packet validation exceptions, the sender may decide to disconnect the BLE connection and initiate a rejoin as described in section 7.7.4.6.

6.4.7.2. Counter overflow exception

The sender SHALL disconnect the BLE connection when its 32-bits outgoing counter reaches its maximum value of 0xffffffff. To re-connect the ZVD can chose to rejoin using the method described in section 7.7.4.6, which will result in a new session key and a reset of both outgoing and incoming counters.

6.4.7.3. Encryption failed exception

When an encryption failed exception occurs, it likely is a bug in the implementation. However, if the exception occurs during encryption of the BLE ATT Read Response payload, then the device SHALL return a BLE ATT Error Response with the error code 0x80 (*Zigbee Direct Application Error*) in response to the BLE ATT Read Request (see [R3] Vol 3, Part F, Section 3).

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

6.4.7.4. Insufficient authentication exception

An insufficient authentication exception occurs in case a ZVD attempts to read or write a characteristic for which its current authentication level is insufficient.

In case an insufficient authentication exception occurs, then the receiver ignores the received request. In case the received request is a BLE ATT Write Request or BLE ATT Read Request, then (next to ignoring the received packet) the receiver SHALL respond with a BLE ATT Error Response with the error code 0x80 (*Zigbee Direct Application Error*) (see [R3] Vol 3, Part F, Section 3).

6.5. Zigbee Direct Security Service

The Zigbee Direct Security Service is a BLE GATT service that is used for the establishment of a secure BLE session. The Zigbee Direct Security Service is a mandatory service, which SHALL have exactly one instance on a ZDD. The service SHALL be instantiated as a primary GATT service with the 128-bit UUID: 29144AF4-0000-4481-BFE9-6D0299B429E3. Bluetooth Low Energy security for the Zigbee Direct Security Service SHALL be set to LE Security Mode 1, Level 1 (no authentication and no encryption) per [R3] Vol 3, Part C, Section 10.2.1.

The purpose of this BLE GATT service is to establish a secure session over BLE.

6.5.1. Zigbee Direct Security Service characteristics

Table 5. Zigbee Direct Security Service characteristics

Characteristic Name	Characteristic UUID	Supported BLE Procedures	BLE Format Type[6]	Authentication Level
Authenticate SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128	29144AF4-0001-4481-BFE9-6D0299B429E3	• Write Request • Indication	struct	*None*
Authenticate SPEKE/Curve25519/SHA-256/HMAC-SHA-256-128	29144AF4-0002-4481-BFE9-6D0299B429E3	• Write Request • Indication	struct	*None*
Authenticate ECDHE-PSK/P-256/SHA-256/HMAC-SHA-256-128	29144AF4-0003-4481-BFE9-6D0299B429E3	• Write Request • Indication	struct	*None*

6.5.1.1. Authenticate

The Authenticate characteristic is used to exchange messages during the secure session establishment. The messages are exchanged in a strict sequence, therefore the device SHALL

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

maintain the state of the session establishment per connection and SHALL update this state according to the messages that have been exchanged. The state is maintained as long as the secure session establishment exchange is active.

The session key and authorization level resulting from the successful secure session exchange SHALL be maintained for as long as the session established between the ZDD and ZVD is active.

Multiple Authenticate characteristics with distinctive UUIDs can be exposed by the ZDD to a ZVD during GATT database discovery. The ZVD SHALL support all authentication methods as defined by the Authenticate characteristics specified below. The ZDD SHALL implement at least one. This specification defines three Authenticate characteristics:

1. SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128
2. Reserved for SPEKE/Curve25519/SHA-256/HMAC-SHA-256-128
3. ECDHE-PSK/P-256/SHA-256/HMAC-SHA-256-128

The characteristic Curve25519/SHA-256/HMAC-SHA-256-128 SHALL be reserved for future use but is not mandatory to implement for both ZVD and ZDD in this version of the specification.

The payload of each Session Establishment message is formatted as shown in Table 6.

Table 6. Payload of Session Establishment message

<table>
<tr><th>Octet #</th><th>Field</th><th>Data Type</th><th>Description</th></tr>
<tr><td>0</td><td>Opcode</td><td>uint8</td><td>Opcode that identifies the Session Establishment message.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Session Establishment Message 1</td></tr><tr><td>2</td><td>Session Establishment Message 2</td></tr><tr><td>3</td><td>Session Establishment Message 3</td></tr><tr><td>4</td><td>Session Establishment Message 4</td></tr><tr><td>0, 5..255</td><td>Reserved</td></tr></table></td></tr>
<tr><td>1..N</td><td>Data</td><td>uint8 array</td><td>Variable</td></tr>
</table>

The *Data* field of the Session Establishment messages makes use of parameters that are formatted as TLV items.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

6.5.1.2. Authenticate using ECDHE-PSK/P-256/SHA-256/HMAC-SHA-256-128

This section covers ECDHE-PSK using P-256.

Table 7. Parameters for ECDHE using P-256

Item	Description
i	Initiator, the device initiating the key exchange (generally the joining device)
r	Responder, the device answering the key exchange (generally the Trust Center)
PSK	Pre-shared Key, see Table PSK Items
H(x)	Hashing Function. This SHALL be SHA-256.
KDF(key, instance)	Key Derivation Function with key, instance as input. This SHALL be HMAC-SHA-256-128 (HMAC-SHA-256 truncated to the first 128 bits)
Ai	Initiator Identity, IEEE EUI-64 of Initiator. For purposes of hashing, the IEEE EUI-64 is assumed to be stored in little-endian order
Ar	Responder Identity, IEEE EUI-64 of Responder. For purposes of hashing, the IEEE EUI-64 is assumed to be stored in little-endian order
I	Session identifier
di	Initiator private key
dr	Responder private key
Qi	Initiator Public Key Point where Qi = diG (notice: only the x-coordinate is relevant)
Qr	Responder Public Key Point where Qr = drG (notice: only the x-coordinate is relevant)
s	Shared secret
DK	Derived key

Table 8. PSK Items

n/a	Anonymous Key Exchange	“ZigbeeAlliance18” (no quotes)
PSK ID 0x00	Key Exchange with Installation Code	Pre-configured link-key derived from Device Installation Code
PSK ID 0x01	Key Exchange with Passcode	Not supported for ECDHE-PSK
PSK ID 0x02	Basic Authorization Key (Zigbee Direct)	Details defined in Zigbee Direct specification
PSK ID 0x03	Administrative Authorization Key (Zigbee Direct)	Details defined in Zigbee Direct specification

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

PSK ID 0x04 – PSK ID 0x07	Reserved for future use	

Operation:

1. Initiator generates di and Qi
2. Initiator sends Qi and Ai
3. Responder generates dr and Qr
4. Responder computes point (xk, yk) = drQi
5. Responder sends Qr and Ar
6. Initiator computes point (xk, yk) = diQr
7. Initiator/Responder determines session identifier I = Ar || Qr || Ai || Qi for Ar ⇐ Ai or Ai || Qi || Ar || Qr for Ar > Ai
8. Initiator/Responder derives shared secret, s = H(xk || I || PSK)
9. Derived Key = KDF(s, {0x01})

6.5.1.2.1. Session Establishment Overview

The following sequence diagram shows a detailed view of all operations performed and messages exchanged between the ZVD (initiator) and ZDD (responder) during the session establishment.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

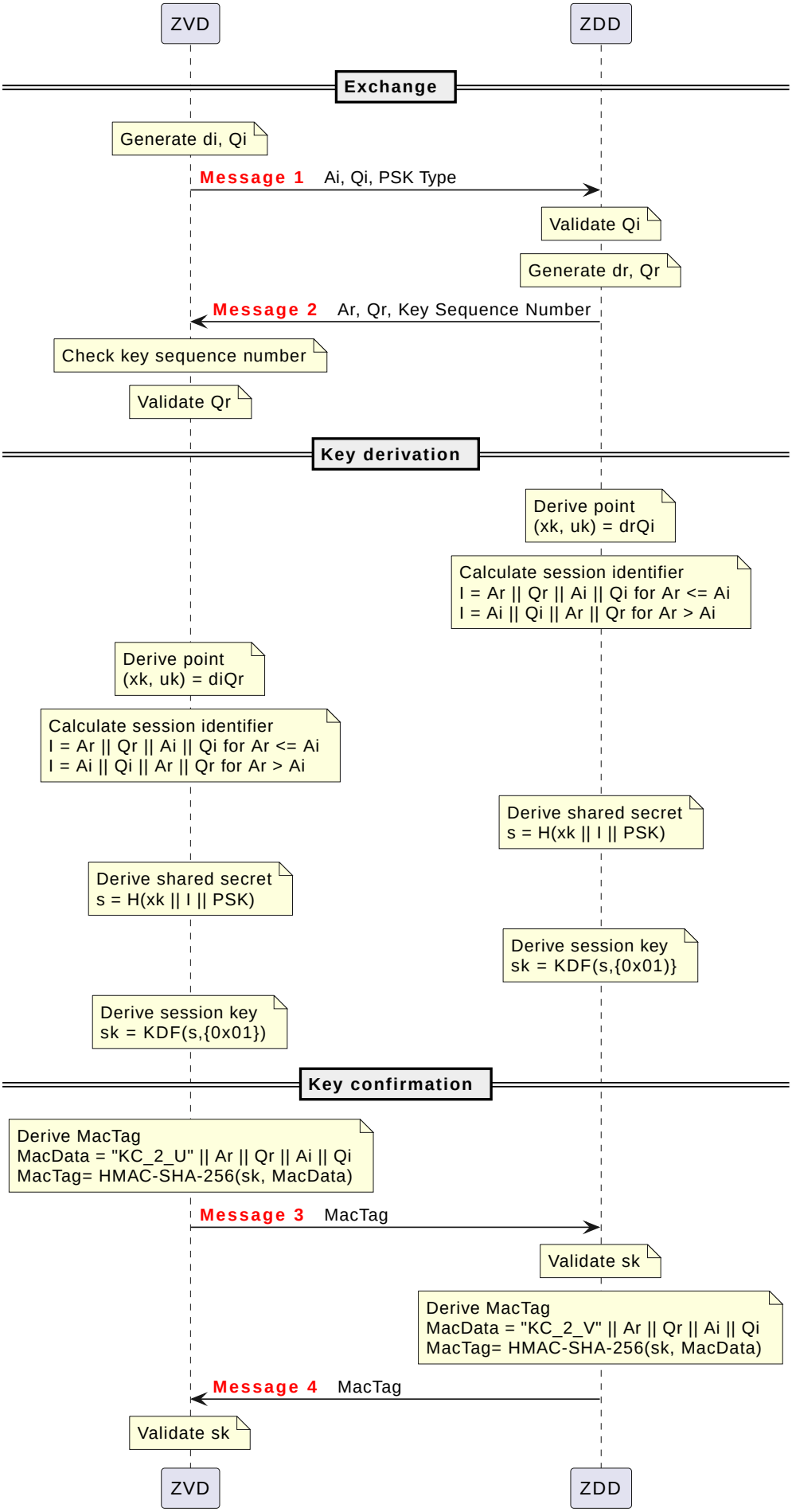


Figure 10. ECDHE PSK Session Establishment Sequence Diagram

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

The session key SHALL be verified to be correctly derived on both sides. The verification is performed by validating the MacTag values exchanged using Session Establishment messages 3 and 4, which were calculated using the derived session key.

None of the messages exchanged during the session establishment are encrypted.

6.5.1.2.2. Session Establishment Message 1

Session Establishment Message 1 is sent from ZVD to ZDD and is used to initiate the Session Establishment protocol and send PSK Type (Pre-shared Secrets Bitmask) and ephemeral public key Qi.

Table 9. Session Establishment Message 1

Opcode	Data	Data	Data
0x01	*Zigbee Direct Key Negotiation Method TLV* PSK used to authenticate Selected negotiation method	*Zigbee Direct Key Negotiation P-256 Public Point TLV* Public key of ZVD (Qi)	*Network Key Sequence Number* Network Key Sequence Number TLV (optional)

Parameters for Message 1:

- Zigbee Direct Key Negotiation Method TLV (See section 6.5.2.1)

  The *Selected Pre-shared Secret Enumeration* octet is used to determine the PSK type used by ZVD to authenticate.

  Selected Key Enumeration octet SHALL be validated by the ZDD to verify if it corresponds with the selected Authenticate characteristic key negotiation method

- Zigbee Direct Key Negotiation P-256 Public Point TLV (See its definition in section 6.5.2.2).
- Network Key Sequence Number TLV (optional)

The Network Key Sequence Number TLV SHALL be included for a PSK type denoting Basic Authorization if key sequence number is non-zero; it MAY be omitted if it is zero. If the value included is different than the ZDD's current network key sequence number in the NIB, the ZVD needs to renew its Basic Authorization key after a network key rotation occurred on a centralized network. In this case, a Basic Authorization key derived from a prior network key, with the key sequence number as indicated in the TLV, is selected for the session establishment. Such a session will provide restricted permissions only, as specified in section 9.1.

6.5.1.2.3. Session Establishment Message 2

Session Establishment Message 2 is sent from the ZDD to the ZVD and is used to send the ephemeral public key Qr and the NWK key sequence number.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 10. Session Establishment Message 2 for a factory new device

Opcode	Data
0x02	*Zigbee Direct Key Negotiation P-256 Public Point TLV* + Public key of ZDD (Qr)

Table 11. Session Establishment Message 2 for a provisioned device

Opcode	Data	Data
0x02	*Network Key Sequence Number TLV* Active network key sequence number (Optional)	*Zigbee Direct Key Negotiation P-256 Public Point TLV* Public key of ZDD (Qr)

Parameters for Message 2:

- Network Key Sequence Number TLV (See its definition in section 6.5.2.4) The Active Network Key Sequence Number TLV SHALL be included when the device is not factory new and the value is different from 0.
- Zigbee Direct Key Negotiation P-256 Public Point TLV (See its definition in section 6.5.2.2).

6.5.1.2.4. Session Establishment Message 3

Session Establishment Message 3 is sent from ZVD to ZDD and is used by ZDD to confirm the same session key was derived on both sides. The message payload containing MacTag as defined below, see NIST Special Publication 800-56A Revision 3, Section 6.1.1.5.3: C(2e, 2s) Scheme with Bilateral Key Confirmation for more details.

```
MacData =  "KC_2_U" || ZDD IEEE Address (Ar) ||  EphPubKey ZDD (Qr) || ZVD IEEE
Address (Ai) || EphPubKey ZVD (Qi)
```

```
MacTag = HMAC-SHA-256(SessionKey, MacData)
```

Note: The Session Key is 128 bits.

Table 12. Session Establishment Message 3

Opcode	Data
0x03	*MacTag TLV* MacTag

Parameters for Session Establishment Message 3:

- MacTag TLV (See its definition section 6.5.2.5).

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

6.5.1.2.5. Session Establishment Message 4

Session Establishment Message 4 is sent from the ZDD to the ZVD and is used by the ZVD to confirm the same session key was derived on both sides. The message payload containing the MacTag as defined below, see NIST Special Publication 800-56A Revision 3, Section 6.1.1.5.3: C(2e, 2s) Scheme with Bilateral Key Confirmation for more details.

```
MacData =  "KC_2_V" || ZDD IEEE Address (Ar) || EphPubKey ZDD (Qr) || ZVD IEEE
Address (Ai) ||  EphPubKey ZVD (Qi)
```

```
MacTag = HMAC-SHA-256(SessionKey, MacData)
```

Note: The Session Key is 128 bits.

Table 13. Session Establishment Message 4

Opcode	Data
0x04	*MacTag TLV* MacTag

Parameters for Session Establishment Message 4:

- MacTag TLV (See its definition in section 6.5.2.5).

6.5.1.3. Authentication SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128

This section covers SPEKE using Curve25519 as described in [R12].

6.5.1.3.1. Session Establishment Overview

The following sequence diagram shows a detailed view of all operations performed and messages exchanged between the ZVD (initiator) and ZDD (responder) during the session establishment.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

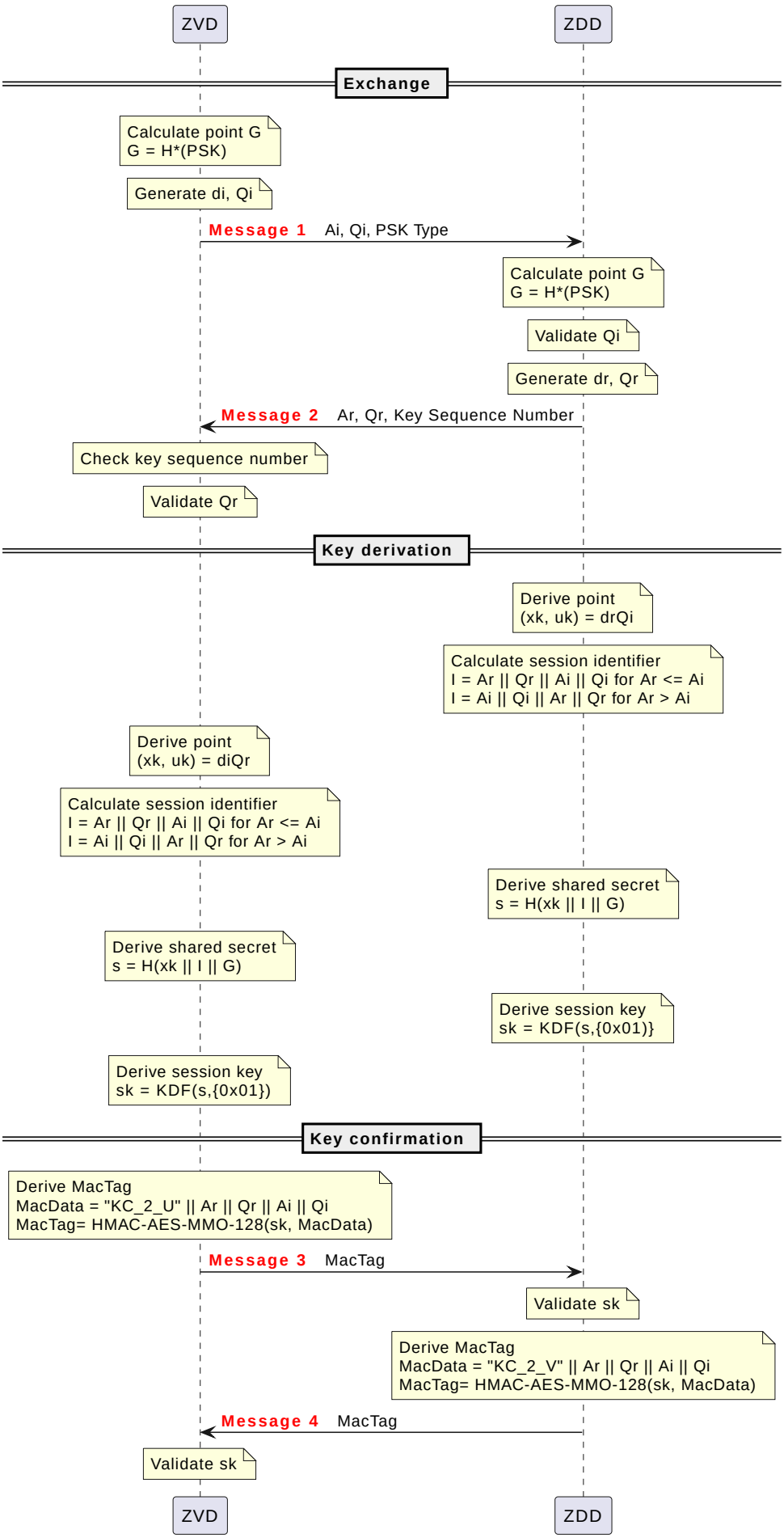


Figure 11. SPEKE Session Establishment Sequence Diagram

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

The session key SHALL be verified to be correctly derived on both sides. The verification is performed by validating the MacTag values exchanged using Session Establishment messages 3 and 4, which were calculated using the derived session key.

None of the messages exchanged during the session establishment are encrypted.

The authentication using the 25519 curve uses the same sequence and format of messages as authentication using P-256 curve, only the Zigbee Direct Key Negotiation Curve25519 Public Point TLV is used instead of its P-256 counterpart.

6.5.1.4. Authentication: Behavior and Exception handling

In case of any validation error detected during the secure session establishment, the device detecting it SHALL disconnect the peer device immediately.

If, after a timeout of *SecureSessionTimeout* seconds, there is no response message sent by the peer device, the device detecting it SHALL disconnect the peer device immediately. *SecureSessionTimeout* SHALL be set to 15 seconds by default[7].

6.5.2. Local TLVs

6.5.2.1. Zigbee Direct Key Negotiation Method TLV (ID=0)

The Table 14 indicates the format of the value field of the Zigbee Direct Key Negotiation Method TLV.

Table 14. Selected Key Negotiation Method

Octets: 1	**Octets: 1**
Selected Key Negotiation Method Enumeration	Selected Pre-shared Secret Enumeration

The Table 15 defines the selected pre-shared key negotiation method enumeration.

Table 15. Selected Key Negotiation Method Enumeration

Enumerated Value	**Description**
0	Reserved
1	ECDHE using Curve25519 with Hash AES-MMO-128
2	ECDHE using Curve25519 with Hash SHA-256
3	ECDHE using curve P-256 with Hash SHA-256
4 – 255	Reserved

The Table 16 defines the Selected Pre-shared Secret Enumeration.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 16. Selected Pre-shared Secret Enumeration

Enumerated Value	Description
0	Symmetric Authentication Token
1	Pre-configured link-key derived from installation code
2	Variable-length pass code (for PAKE protocols)
3	Basic Authorization Key
4	Administrative Authorization Key
5 – 254	Reserved
255	Anonymous Well-Known Secret

6.5.2.2. Zigbee Direct Key Negotiation P-256 Public Point TLV (ID=1)

The Table 17 indicates the format of value field of the Zigbee Direct Key Negotiation P-256 Public Point TLV.

Table 17. P-256 Public Point TLV Format

Octets: 8	Octets: 64
Device EUI64	Public Point

The fields of the Zigbee Direct Key Negotiation P-256 Public Point TLV are in Table 18.

Table 18. Fields of the Zigbee Direct Key Negotiation P-256 Public Point TLV

Field	Description
Device EUI64	This indicates the EUI64 of the device that generated the public point.
Public Point	The 64-byte P-256 public point.

6.5.2.3. Zigbee Direct Key Negotiation Curve25519 Public Point TLV (ID=2)

The Table 19 indicates the format of the value field of the Zigbee Direct Key Negotiation Curve25519 Public Point TLV.

Table 19. Format of the Zigbee Direct Key Negotiation Curve25519 Public Point TLV

Octets: 8	Octets: 32
Device EUI64	Public Point

The fields of the Zigbee Direct Key Negotiation Curve25519 Public Point TLV are in Table 20.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 20. Fields of the Zigbee Direct Key Negotiation Curve25519 Public Point TLV

Field	Description
Device EUI64	This indicates the EUI64 of the device that generated the public point.
Public Point	The 32-byte Curve25519 public point.

6.5.2.4. Network Key Sequence Number TLV (ID=3)

The Table 21 indicates the format of the value field of the Network Key Sequence Number TLV.

Table 21. Format of the Network Key Sequence Number TLV

Octets: 1
Network Key Sequence Number

6.5.2.5. MacTag TLV (ID=4)

The Table 22 indicates the format of the value field of the MacTag TLV. The message authentication code (MAC) derived using a cryptographic hash function and a secret key and is 32-bytes in length for HMAC-SHA256 function and 16-bytes in length for HMAC-AES-MMO-128.

Table 22. Format of the MacTag TLV

Octets: 32/16
MacTag

6.6. ZVD Provisioning

A new ZVD - i.e., a ZVD that is accessing this Zigbee network for the first time - MAY be authorized to access a Zigbee network either via an out-of-band method (out of scope of this specification) or via the BLE interface provided by a ZDD device, which is already provisioned to the network. The latter option SHALL only be possible in case the Zigbee network has been opened by a node on the Zigbee network issuing a *Mgmt_Permit_Joining_req* command, as defined in the Zigbee specification [R12] or later, such that the network's constituent nodes respond with Beacons with the Association Permit bit set when requested. Only the basic authorization key SHALL be granted to ZVDs provisioned via the BLE interface.

When the ZVD tries to establish a provisioning session using the Anonymous Well-Known Secret to a ZDD which is already provisioned to the network, the ZDD SHALL verify the value of the Anonymous Join Countdown Timer. When the value of the Anonymous Join Countdown Timer is set to 0 seconds, the ZDD SHALL NOT allow the provisioning session establishment with the ZVD and close the connection.

Additionally, a ZVD can form both a centralized as well as a distributed security Zigbee network.

For more details on these operations please refer to section 7.7.4.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7. ZDD Specification

7.1. Introduction

A ZDD is a Zigbee device capable of concurrently running both Zigbee PRO and BLE stacks, implementing functionality which enables the exchange of data between certain BLE-based devices (ZVDs) and a Zigbee PRO network. This section specifies the behavior of the ZDD.

7.2. Basic device behavior

A ZDD SHALL support at minimum Zigbee specification as defined in [R12].

7.3. Security

Security is handled as described in section 6.

7.4. BLE network parameters

The parameters shown in Table 23 SHOULD be configured with the provided default values for a ZDD[8].

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 23. Recommended Bluetooth Low-Energy parameters

Parameter	Recommended value (range)	Remarks
Advertising interval [R3] Vol 3, Part C, Section 9.3.11	152.5 ms	recommended
Connection interval [R3] Vol 3, Part C, Section 9.3.12	7.5 - 100 ms	
Data packet length extension[R3] Vol 6, Part B, Section 4.6	supported	
ATT_MTU [R3] Vol 3, Part F, Section 3.2.8	*MinAttMtuSize* = 185 octets *RecommendedAttMtuSize* = 247 octets	
Connections maximum	*MaxZvdConnections*	MaxZvdConnections is defined as the maximum number of concurrent Zigbee Direct (BLE) connections supported on the device, this SHALL be at least 1.

7.5. ZDD specific Zigbee network parameters

The following contains a set of extensions to the Zigbee Specification [R12]. These do not fundamentally change the stack behavior and are optional for the stack software, but mandatory for a ZDD implementing the Zigbee Direct Tunnel Service.

The ZDD behaves similar to a Multi-MAC Switch Device where each BLE connection to an authenticated ZVD represents its own individual MAC interface.

To support this, the ZDD SHALL have free capacity for a minimum of *MaxZvdConnections* entries to be added to its *nwkMacInterfaceTable*. In addition to this the ZDD SHALL have a minimum *nwkNeighborTable* size of 16 + *MaxZvdConnections*, where 16 is the value defined by Zigbee Specification [R12]. *MaxZvdConnections* entries are permanently reserved for ZVD entries. Only connected ZVDs that operate as routers SHALL be included in the link status message sent out by the ZDD. In the link status messages both the incoming and outgoing cost indicated by the ZDD for a link to the ZVD SHALL be set to 1 (best). In case the connected ZVD's device type is ZR or ZC, a ZDD SHALL also send link status messages including all relevant neighbors to the ZVD and expect link status messages back from it.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Additionally, a ZDD SHALL utilize the MAC Interface Index field present in every entry of the *nwkNeighborTable* to point to the corresponding entry in the *nwkMacInterfaceTable*.

7.5.1. NwkMacInterface Parameters

After creation of a Zigbee Direct BLE connection and successful authentication of a ZVD, an entry in the *nwkMacInterfaceTable* with the Type field set to 1 (Trusted Link), the BLE Connection Handle field set to the 16-bit BLE connection identifier and the State field set to TRUE SHALL be created. For more information about the newly added Type and BLE Connection Handle fields refer to [new-mac-interface-table-fields]. In case the *nwkMacInterfaceTable* is full, the oldest entry for a Trusted Link type MAC Interface with the State field set to FALSE SHALL be overwritten.

In case there are a number of *MaxZvdConnections* active and authenticated Zigbee Direct BT connections (i.e. *MaxZvdConnections* entries with both the Type field set to 1 (Trusted Link) and the State field set to TRUE) the ZDD SHALL stop sending the connectable Zigbee Direct advertisement (see section 7.7.3). If there are manufacturer specific extensions advertised, advertisements MAY still be sent, but the Zigbee Direct specific advertisement data (section 7.6.3) SHALL be removed from the advertisement and the advertisement frequency SHOULD be set to the lowest possible value for the device.

Note that if the Type field of a given entry in the *nwkMacInterfaceTable* is set to 1 (Trusted Link), both the Supported Channels as well as the Channel In Use fields are undefined and SHALL be ignored by the stack.

In case a Zigbee Direct Bluetooth connection is terminated for any reason, the State field of the corresponding entry in the *nwkMacInterfaceTable* SHALL be set to FALSE. In addition to this, any entries in the ZDDs *nwkNeighborTable* which are listed against this particular MacInterface SHALL be deleted. If connectable Zigbee Direct advertisements as per section 7.6.2 were stopped because *MaxZvdConnections* had been previously exceeded, they SHALL be resumed.

7.6. Discovery over BLE

7.6.1. Introduction

The discovery protocol as described in this section enables a ZVD to discover a ZDD and retrieve some basic properties from the ZDD.

7.6.2. Basic discovery

A ZDD SHALL implement a BLE connectable advertisement that includes the Zigbee Direct BLE Advertisement Extension (see section 7.6.3), enabling a ZVD to detect ZDD-capable devices in its vicinity. The ZDD MAY transition to a non-connectable advertisement under application control.

The ZDD SHALL send these BLE connectable advertisements when in a factory new state. By default, the ZDD SHALL send connectable advertisements when configured for a Zigbee network

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

– as long as the number of established BLE connections is lower than *MaxZvdConnections*. The ZDDs BLE interface can be disabled using the Zigbee Direct Configuration cluster section 11.3.

7.6.3. Zigbee Direct extension to the BLE Advertisement

The BLE advertisement SHALL conform to [R3] Vol 3, Part C, Section 11. The advertisement SHALL be unencrypted.

The BLE advertisement MAY include manufacturer specific data. Any manufacturer specific data SHALL NOT impede the function or visibility of Zigbee defined Bluetooth GATT services.

The BLE connectable advertisement of the ZDD SHALL include records shown in Table 24.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 24. Zigbee Direct extension to the BLE Advertisement

<table>
<tr><th>AD field</th><th>Value / Field Name</th><th>Field size</th><th>Description</th></tr>
<tr><td>AD Length</td><td>0x08</td><td>1 octet</td><td>Length of this Advertising Data (AD) field</td></tr>
<tr><td>AD Type</td><td>0x16</td><td>1 octet</td><td>Advertising Data Type set to the Service Data data type <<Service Data - 16 bit UUID>> as defined in [R5]</td></tr>
<tr><td rowspan="8">AD Data</td><td>16-bit UUID (low byte)</td><td>1 octet</td><td rowspan="2">16-bit UUID of Zigbee Direct Commissioning Service</td></tr>
<tr><td>16-bit UUID (high byte)</td><td>1 octet</td></tr>
<tr><td>ZD version</td><td>4 bits (lsb)</td><td>Version of this Zigbee Direct extension to the BLE Advertisement. SHALL have a value of 0x1 for ZDDs that conform to this specification.</td></tr>
<tr><td>ZD flags</td><td>4 bits (msb)</td><td>Flags that indicate the ZDD’s capabilities and status.
<table>
<tr><th>Bit</th><th>Flag name</th><th>Description</th></tr>
<tr><td>4</td><td>ZDTS</td><td>Indicates if ZDD supports the Zigbee Direct Tunnel Service.</td></tr>
<tr><td>5</td><td>PermitJoin</td><td>Indicates the ZDD’s permit joining status. SHALL be set to 0 when the ZDD is currently not joined to a Zigbee network.</td></tr>
<tr><td>6-7</td><td>Reserved</td><td>Reserved, set to 0</td></tr>
</table></td></tr>
<tr><td>Zigbee PAN ID (low byte)</td><td>1 octet</td><td rowspan="2">The 16-bit PAN identifier of the Zigbee network. SHALL be set to 0xFFFF when the ZDD is currently not joined to a Zigbee network.</td></tr>
<tr><td>Zigbee PAN ID (high byte)</td><td>1 octet</td></tr>
<tr><td>Zigbee NWK address (low byte)</td><td>1 octet</td><td rowspan="2">The ZDD’s 16-bit NWK address within the Zigbee network. SHALL be set to 0xFFFF when the ZDD is currently not joined to a Zigbee network.</td></tr>
<tr><td>Zigbee NWK address (high byte)</td><td>1 octet</td></tr>
</table>

When the value of one of the items in this Zigbee Direct extension to the BLE advertisement changes (e.g. due to the ZDD joining or leaving a Zigbee network), then the ZDD SHALL reflect

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

this change in the BLE advertisement.

It is RECOMMENDED for the ZDD to include the Zigbee Direct Commissioning Service (see section 7.7.2) as a primary service in its BLE advertisement. See Appendix A as an example of a complete BLE advertisement.

7.6.4. ZDD additional information

In case a ZDD is going to provide additional information like HW version, SW version, etc., it is a common practice to use a standard BLE Device Information Service (0x180A) for this.

```
Some operating systems automatically read the information from a device's Device
Information Service (when included). For that reason (and because it's a Bluetooth
SIG defined service) these characteristics must not be protected/encrypted using
Zigbee Direct security.
```

Some widely used device information characteristics can be found in the Table 25.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 25. Some UUIDs of BLE Device Information Characteristics

Characteristic Name	Characteristic UUID	Supported BLE Procedures	BLE Format Type[6]	Authentication Level	Mandatory / Optional
Device Name String	0x2A00	Read request	String	No authentication required	O
Model Number String	0x2A24	Read request	String	No authentication required	O
Serial Number String	0x2A25	Read request	String	No authentication required	O
Firmware Revision String	0x2A26	Read request	String	No authentication required	O
Hardware Revision String	0x2A27	Read request	String	No authentication required	O
Software Revision String	0x2A28	Read request	String	No authentication required	O
Manufacturer Name String	0x2A29	Read request	String	No authentication required	O

7.7. BLE Services

7.7.1. Introduction

This paragraph describes the BLE Services that a ZDD supports.

7.7.2. Zigbee Direct Commissioning Service

7.7.2.1. Introduction

The Zigbee Direct Commissioning Service is a BLE GATT service that can be used to configure (add and remove) ZDD devices for a Zigbee network. The Zigbee Direct Commissioning Service is a mandatory service, which SHALL have exactly one instance on a ZDD. The service SHALL be instantiated as a primary GATT service with the assigned 16-bit UUID: 0xFFF7 [R6]. The Assigned

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Value is provided by Bluetooth SIG, Inc. and may only be used by its members in compliance with all terms and conditions of use issued by Bluetooth SIG, Inc. For more information visit https://www.bluetooth.com/specifications/assigned-numbers.

The service contains several characteristics to trigger the ZDD to form, join, leave, open/close a network, as well as query them about their status. Jointly with other BLE GATT services, they can be used in the commissioning tool and in the ZDD for different commissioning workflows.

The BLE GATT service allows for instructing the ZDD to utilize MAC association to join a Zigbee network. Another possible commissioning workflow with the “join network” characteristic is out of band Zigbee commissioning, allowing to provision a ZDD with network parameters without the need to utilize the standard Zigbee association/joining process.

The out of band Zigbee commissioning workflow of a factory new ZDD allows any ZVD to establish secure BLE provisioning session (using one of the shared secrets) and provision the ZDD with network parameters (including the network key) to form a new Zigbee network or allow the ZDD to join an existing network.

A ZDD that is already provisioned to an existing Zigbee network can be removed from the network or added to another network. This can be preformed by any ZVD that is able to established secure BLE authorization session with the ZDD using the admin authorization key, and by doing so get access to the commissioning service.

The out of band commissioning workflow using the secure BLE session provides secure and convenient way to transfer the Zigbee network parameters (most notably the network key) from the ZVD to ZDD. Further, the workflow streamlines the Zigbee provisioning process by providing all network parameters required to join intended network before the Zigbee device provisioning exchange is initiated. This eliminates the need for network scanning and key delivery but also prevents the ZDD from joining the wrong network.

Relevant errors or status codes coming up whilst using the Zigbee Direct Commissioning Service SHALL be reported from the GATT server (ZDD) to the GATT client (ZVD) in either one of the following two ways:

1. Any errors or status codes related to the native BLE ATT client-server protocol will be reported using a BLE ATT error response (see [R3] Vol 3, Part F, Section 3)
2. Errors and other status codes related to the interpretation of the ATT payload, ranging from parsing and processing of TLVs to executing the individual Service characteristics' Zigbee Direct defined functionality SHALL make use of the Status Code TLV item listed in section 7.7.2.4. Additional information on the status codes related to the interpretation of the ATT payload as well as unrelated events MAY be reported using the Extended Status Code TLV which is outlined in the same section. Both the Status Code TLV and the Extended Status Code TLV are sent as notifications via the Commissioning Status characteristic. In addition to reporting the status code the ZDD SHALL respond to an otherwise correct BLE ATT write message with a write response with status of success. Throughout this section detailed information on status code handling for each individual service characteristic is provided.

If the GATT server (ZDD) is already processing a Commissioning Service request or command, it

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

SHALL NOT process any further commands until the previous request is completed. A request or command can be deemed completed when, depending on the characteristic, either a notification or write response was sent to Client (ZVD) or when an error response was sent. In case a request or command is received when processing a previous command, the ZDD SHALL send a notification on the Commissioning Status characteristic containing the Status Code TLV (see section 7.7.2.4) with the error code and domain appropriate to the characteristic the newly received request or command was targeting. If for the duration of *CommissioningServiceTimeout* (3 Seconds) no response of any kind is received via the Commissioning Status Characteristic, the operation SHALL be considered a failure, unless the ZVD proves otherwise via the results of an active read of the Commissioning Status characteristic.

All parameter values in this service SHALL be provided in Little Endian byte order (unless explicitly specified differently, see also [R12]).

7.7.2.2. Security

The Zigbee Direct Commissioning Service requires Zigbee Direct authentication for usage of any of its BLE GATT characteristics except for a Read Request of the Commissioning Status characteristic [9]. A ZVD SHALL setup a secure session with the ZDD that it wants to commission (see section 6.3). Refer to Table 26 for the required authentication level per characteristic. For each characteristic in this service, Bluetooth Low Energy security SHALL be set to LE Security Mode 1, Level 1 (no authentication and no encryption) per [R3] Vol 3, Part C, Section 10.2.1.

In case the ZVD attempts to read or write a characteristic for which its current authentication level is insufficient, then this SHALL result in an insufficient authentication exception, which SHALL be handled by the ZDD as described in section 6.4.7.4.

The ZDD SHALL NOT send BLE notifications or indications to a ZVD on a characteristic for which the ZVD's current authentication level is insufficient.

7.7.2.3. Zigbee Direct Commissioning Service characteristics

Table 26. Zigbee Direct Commissioning Service characteristics

Characteristic Name	Characteristic UUID	Supported BLE Procedures	BLE Format Type[6]	Authorization Level	Mandatory / Optional
Form Network	7072377D-0001-421C-B163-491C27333A61	• Write Request	struct	• Provisioning session established with un-provisioned ZDD	M
Join Network	7072377D-0002-421C-B163-491C27333A61	• Write Request	struct	• Provisioning session established with un-provisioned ZDD	M

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Characteristic Name	Characteristic UUID	Supported BLE Procedures	BLE Format Type[6]	Authorization Level	Mandatory / Optional
Permit Joining	7072377D-0003-421C-B163-491C27333A61	• Write Request	uint8	• Provisioning session established with un-provisioned ZDD • Admin level authorization	M
Leave Network	7072377D-0004-421C-B163-491C27333A61	• Write Request	struct	• Provisioning session established with un-provisioned ZDD • Admin level authorization	M
Commissioning Status	7072377D-0005-421C-B163-491C27333A61	• Read Request • Notification	struct	• Provisioning session • Admin level authorization • Basic level authorization	M
Manage Joiners	7072377D-0006-421C-B163-491C27333A61	• Write Request	struct	• Admin level authorization	O
Identify	7072377D-0007-421C-B163-491C27333A61	• Write Request • Read Request	uint16	• Provisioning session established with un-provisioned ZDD • Basic level authorization • Admin level authorization	M
Finding&Binding	7072377D-0008-421C-B163-491C27333A61	• Write Request	struct	• Basic level authorization • Admin level authorization	O

For admin level Authorization, the ZVD is required to have established an Authorization session using the Admin Key as outlined in section 6.3.2. For this the ZVD might have generated the Admin Key and provided it to the ZDD when instructing the ZDD to originally form the network using the Form Network characteristic. Alternatively the Admin Key might be shared from the ZDD to the ZVD in an out of band method which is out of scope for this specification.

For basic level Authorization, the ZVD is required to have established an Authorization session

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

using the Basic Key as outlined in section 6.3.2. For this the Basic Key can be provided to the ZVD during the ZVD provisioning as described in section 6.6 or the key might be shared with the ZVD via out of band method which is out of scope for this specification.

7.7.2.4. TLV Items

The Zigbee Direct Commissioning Service makes use of TLV items in its Form Network, Join Network, Commissioning Status and Manage Joiners characteristics. The TLV items are formatted according to the Zigbee specification (see [R12], annex I "Zigbee TLV Definitions And Format").

Any errors encountered during general processing of the TLV items are reported using the Status Code TLV.

7.7.2.4.1. General TLV processing

The ZDD and the ZVD SHALL process the TLV items following the "General TLV processing" rules as defined in the Zigbee specification (see [R12] Annex I, Section "General TLV Processing").

7.7.2.4.2. Local TLV IDs for Commissioning

The TLV IDs that are used within Form Network, Join Network and Commissioning Status characteristics of the Zigbee Direct Commissioning Service are defined in Table 27.

Table 27. Zigbee Direct Commissioning Service TLV IDs

Type ID	Name	Zigbee Data Type	Characteristic Usage	Description
0x00	Extended PAN ID	EUI64	Form Network Join Network Commissioning Status	Specifies the Zigbee network's extended PAN ID. Range: 0x0000.0000.0000.0001 to 0xffff.ffff.ffff.fffe inclusive
0x01	Short PAN ID	uint16	Form Network Join Network Commissioning Status	Specifies the Zigbee network's short PAN ID. Range: 0x0001 to 0xfffe inclusive

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Type ID	Name	Zigbee Data Type	Characteristic Usage	Description
0x02	Network Channel	ChannelListStructure[R12]	Form Network Join Network Commissioning Status	To convey channel information for a device that MAY be operating on multiple MAC interfaces it is necessary to utilize a data structure that can convey this complete information. Channels are divided into groups known as pages. Each page indicates information about a particular band while the list of channels is indicated via bits within the page. A ChannelList structure SHALL contain only one instance of each channel page.
0x03	Network Key	key128	Form Network Join Network	Specifies the Zigbee Network Key.
0x04	Link Key	pair of <map8, key128>	Form Network Join Network	Specifies the Zigbee Link Key. It can be used to provide unique ZDD TC Link Key, see section 7.7.2.4.7, in case of Out-Of-Band (OOB) commissioning of ZDD, allowing the ZDD to join the Zigbee network, see section 7.7.2.7.
0x05	Device Type	uint8	Commissioning Status	Specifies the ZDD's Zigbee device type. See section 7.7.2.4.3.
0x06	NWK Address	uint16	Form Network Join Network Commissioning Status	Specifies the ZDD's 16-bit Zigbee network address. Range: 0x0000 to 0xfff7 inclusive
0x07	Joining Method	uint8	Join Network	Specifies the method the ZDD is instructed to use to join the network. See section 7.7.2.4.4.
0x08	IEEE Address	EUI64	Commissioning Status	The 64-bit IEEE address of the ZDD.
0x09	Trust Center Address	EUI64	Form Network Join Network Commissioning Status	The 64-bit IEEE address of the Trust Center.
0x0a	Network Status Map	map8	Commissioning Status	Contains a bitmap that indicates the ZDD's network status. See section 7.7.2.4.5.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Type ID	Name	Zigbee Data Type	Characteristic Usage	Description
0x0b	NWK Update ID	uint8	Form Network Join Network Commissioning Status	Current value of the *nwkUpdateId* attribute of the Zigbee NIB.
0x0c	NWK Active Key Seq Number	uint8	Form Network Join Network Commissioning Status	Current value of the *nwkActiveKeySeqNumber* attribute of the Zigbee NIB.
0x0d	Admin Key	key128	Form Network Join Network	Specifies the 128-bit Admin Key. See section 6.3.2.1.
0x0e	Status Code	pair of <uint8, uint8>	Form Network Join Network Permit Joining Leave Network	Commissioning status as described in section 7.7.2.4.6.
0x0f	Extended Status Code	variable	Any	Ext. Status code for custom use.
0x10-0xff	*Reserved*			

7.7.2.4.3. Device Type TLV

Table 28. Device Type TLV values

Key	Description
0x00	Zigbee coordinator
0x01	Zigbee router
0x02	Zigbee end device

7.7.2.4.4. Joining Method TLV

Table 29. Joining Method TLV values

Key	Description
0x00	Join through MAC association procedure.
0x01	Join using secure NWK rejoining procedure (with key).
0x02	Out-of-band Commissioning. Join network using provided network parameters. The parameters will be applied "as is", without additional connectivity checks; plausibility and range checks still apply.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.7.2.4.5. Network Status Map TLV

Table 30. Network Status Map TLV values

<table>
<tr><th>Bits</th><th>Field Name</th><th>Description</th></tr>
<tr><td>0..2</td><td>Joined Status</td><td>The possible join states.<table><tr><th>Key</th><th>Description</th></tr><tr><td>0</td><td>Not commissioned</td></tr><tr><td>1</td><td>Commissioning in progress</td></tr><tr><td>2</td><td>Commissioned</td></tr></table></td></tr>
<tr><td>3</td><td>Open / Closed Status</td><td>The open / closed status of the Zigbee network.<table><tr><th>Key</th><th>Description</th></tr><tr><td>0</td><td>Network closed</td></tr><tr><td>1</td><td>Network open</td></tr></table>This field is only valid in case Joined Status is "Joined network".</td></tr>
<tr><td>4</td><td>Network Type</td><td>The type of the Zigbee network.<table><tr><th>Key</th><th>Description</th></tr><tr><td>0</td><td>Distributed</td></tr><tr><td>1</td><td>Centralized</td></tr></table>This field is only valid in case Joined Status is "Joined network" or "Joined network, no parent".</td></tr>
<tr><td>5..7</td><td>Reserved</td><td></td></tr>
</table>

7.7.2.4.6. Status Code TLV

Table 31 indicates the format of the Status Code TLV.

Table 31. Format of the Zigbee Direct Status Code TLV

Octets: 1	Octets: 1
Domain/Operation Code	Status Code

The fields of the Zigbee Direct Status Code TLV are defined in Table 32.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 32. Fields of the Zigbee Direct Status Code TLV

Field	Description
Domain/Operation	This indicates the domain or a particular operation that the following status code pertains to.
Status code	The 8-bit status code within the context/scope of the domain/operation.

Table 33. Status Code TLV Domain/Operation values

Key	Description
0x00	General domain or unspecific operation
0x01	Form Network Operation
0x02	Join Network Operation
0x03	Permit Joining Operation
0x04	Leave Network Operation
0x05	Manage Joiners Domain
0x06	Identify Operation
0x07	Finding & Binding Domain
0x08 - 0xFF	Reserved.

Status Code Values

A status code is returned in response to a particular operation being requested, or might originate unsolicited from within a particular functional domain. The status code has meaning within the context or scope as specified by the leading operation/domain value.

As a general rule, successful completion of an operation SHALL be indicated by the value zero, SUCCESS (0). Conversely, all non-zero values SHALL be considered indicative of a failure to perform the requested operation, or an error arising within the specified domain, respectively. There is no concept of a partially successful operation - an operation either fails or succeeds.

Implementations SHALL provide at least a coarse indication of success or failure. Where possible implementations SHOULD provide more granular status codes for failure conditions. For example, a ZVD might want to understand if a requested operation failed due to a transitional error (e.g. temporary lack of resources, an ephemeral connectivity issue, etc.) and might be reattempted; or the requested operation would always fail (e.g. unsupported feature or frequency band).

Implementations SHALL only use status code values defined in the present Zigbee Direct specification, any Connectivity Standards Alliance specification referenced herein [R12], [R8], [R10], or any third-party specification referenced therein. Particular status codes might be added or removed in future versions of these specifications and implementations applying the general

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

rule above inherently withstand and tolerate such changes.

For example, as a result of a Join Network Operation, the ZDD MAY include the status code conveyed in the Status parameter of the NLME-NETWORK-FORMATION.confirm primitive comprising SUCCESS, a network layer status code (INVALID_REQUEST, STARTUP_FAILURE), or a MAC layer status code (NO_SHORT_ADDRESS, SUPERFRAME_OVERLAP, TRACKING_OFF, INVALID_PARAMETER, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, or CHANNEL_ACCESS_FAILURE). Notice that these symbolic status codes SHALL be represented using numeric values as defined in [R12] or specifications referenced directly or indirectly therein. Similarly, commissioning status notification generated in response to a Join Network operation MAY convey the value of the Status parameter of the corresponding NLME-JOIN.confirm.

7.7.2.4.7. Link Key TLV

Table 34 indicates the format of the Link-key TLV.

Table 34. Format of the Zigbee Direct Link-Key TLV

Octets: 1	**Octets: 16**
APS Key Flags	APS Link Key

The fields of the Zigbee Direct APS Key Flags are defined in Table 35.

Table 35. Zigbee Direct APS Link-Key TLV Flags

Bit(s)	**Field Name**	**Description**
0	Unique	Unique vs. global APS key. **Value** / **Description** 0 – Global Key 1 – Unique Key A global key SHALL be stored in an apsKeyPairSet entry with its LinkKeyType field set to Global, whereas a unique key SHALL be stored with the LinkKeyType set to Unique.
1	Provisional	The life-time of the key: provisional vs. permanent key. **Value** / **Description** 0 – Permanent key 1 – Provisional key A permanent key SHALL be stored in an apsKeyPairSet entry with its KeyAttribute set to Permanent, whereas a provisional key SHALL be stored with the KeyAttribute set to Provisional.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.7.2.5. Form Network characteristic

The Form Network characteristic can be used to form a Zigbee network and to setup the ZDD with a number of network parameters.

This characteristic lets the ZDD form a Zigbee network without opening the network for other devices to join. In case other devices need to join the Zigbee network through association, then the Permit Joining characteristic should be used.

The Form Network characteristic can only be written in case the ZDD is currently not a member of a Zigbee network (i.e. the Joined Status field of the Network Status Map TLV item holds the value "No network"), otherwise the ZDD SHALL notify the ZVD of any failure to form a network by sending a notification on the Commissioning Status characteristic containing the Status Code TLV with error code "Form Network error" and domain code 0x01 "Form Network Operation" (see section 7.7.2.4.6).

In case the Form Network characteristic is used to form a centralized security network on a ZDD that is not capable to operate as a ZC, then the ZDD SHALL notify the ZVD of a failure to form a network by sending a notification on the Commissioning Status characteristic containing the Status Code TLV with error code "Form Network error" and domain code 0x01 "Form Network Operation" (see section 7.7.2.4.6).

Once the network formation has completed, the Zigbee network SHALL be in a closed state, and a notification containing all TLVs listed in Table 43 will be sent on the Commissioning Status characteristic (where Joined Status field of the Network Status Map TLV is set appropriately to reflect the result of the network formation operation). It is not required for this notification to include the Status Code TLV to additionally indicate success of the operation.

Once the network has been formed, the ZDD, SHALL commence sending out periodic Link Status messages as per section 3.6.3.4 of [R12] (to signal the node's network presence to neighboring nodes).

7.7.2.5.1. Form Network Write

The Form Network characteristic can be used in multiple ways, with a mixture of parameters (if provided). If a particular TLV item is not provided or in case no TLV items are provided at all (i.e. zero bytes written to the characteristic), then the ZDD will either use the default value for a specific parameter, if available, or it will generate a random value.

The TLV types that are used in the Form Network characteristic payload are as specified in Table 36.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 36. TLV usage in Form Network characteristic

TLV	Usage	Description
Admin Key	Optional	Used to provide the Admin key as defined and described in section 6.3.2.1. A ZVD provisioning a ZDD to form a distributed security network will not be able to access the ZDD using admin authorization if the key is not provided during the provisioning session.
Trust Center Address	Required (conditionally)	SHALL be included and set to 0xFFFFFFFFFFFFFFFF when forming a distributed security network. SHALL be excluded when forming a centralized security network, as the ZDD will always become the TC.
Extended PAN ID	Optional	When omitted, then the ZDD SHALL select an appropriate - as specified in [R12] (NLME-NETWORK-FORMATION.request) - extended PAN ID for network formation.
Short PAN ID	Optional	When omitted, then the ZDD SHALL select an appropriate - as specified in [R12] (NLME-NETWORK-FORMATION.request) - PAN ID for network formation.
Network Channel	Optional	When included and the provided ChannelList structure contains multiple channels, then the ZDD SHALL perform an energy scan and select the channel with the lowest energy level, on the provided channels only. When omitted, then the ZDD SHALL implicitly perform an energy scan. The results of the energy scan SHALL be used to select a network channel. The implementation (including the exact channels and duration) of this scan is product-specific and therefore not specified here. When this TLV item is included and the provided ChannelList structure contains exactly one channel, then the ZDD SHALL NOT perform an energy scan and SHALL directly use the provided channel for network formation.
Network Key	Optional	When omitted, then the ZDD SHALL generate a random Zigbee Network Key that is used to form the network.
Link Key	Optional	When omitted, then a persistently stored (proprietary) link key SHALL be used to form the Zigbee network. If no (proprietary) link key is available, then the default distributed security global link key SHALL be used when forming a distributed security network, or the ZigBeeAlliance09 key SHALL be used – as the global trust center link key – when forming a centralized security network.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

TLV	Usage	Description
NWK Address	Optional	When omitted, then the ZDD SHALL assign itself a random short address when forming a distributed security network. TLV item SHALL be ignored in case of forming a centralized security network, which SHALL always result in ZDD assigning itself short address 0x0000.
NWK Update ID	Optional	When omitted, then the ZDD SHALL assume that the NWK Update ID is 0.
NWK Active Key Seq Number	Invalid	The ZDD SHALL assume that the NWK Active Key Seq Number is 0.

7.7.2.6. Permit Joining characteristic

This characteristic lets the ZDD permit or prohibit the joining of the Zigbee network by new devices if the ZDD is a ZC or ZR itself, and in addition lets the ZDD broadcast a ZDO *Mgmt_Permit_Joining_req*. The value that is written to this characteristic holds the PermitDuration, which is the time in seconds during which the Zigbee coordinator or router will allow associations. This value SHALL be used in the ZDO *Mgmt_Permit_Joining_req*'s PermitDuration parameter.

Table 37. Permit Joining characteristic values

Value	Description
0x00	Closes the network
0x01..0xfe	Opens the network with the specified time limit (in seconds).
0xff	0xFF SHALL be interpreted as 0xFE, following the Zigbee Core Specification of the ZDO *Mgmt_Permit_Joining_req* (see section 2.4.3.3.7 of [12]).

The command is only accepted where the ZDD is currently a member of a Zigbee network (i.e. the Joined Status field of the Network Status Map TLV item holds the value "Joined network"), otherwise the ZDD SHALL notify the ZVD of any failure to permit or prohibit joining the network by sending a notification on the Commissioning Status characteristic containing the Status Code TLV with error code "Permit Joining error" and domain code 0x03 "Permit Joining Operation" (see section 7.7.2.4.6).

Opening the network while the network has already been opened will keep the network open for the specified time limit. Therefore, keeping the network open for a longer period of time than 254 (0xFE) seconds can be accomplished by doing an "open network" again just before the network is closed.
Once the join period or permissions have been updated a notification containing all TLVs listed in Table 43 will be sent on the Commissioning Status characteristic (where Open / Closed Status field of the Network Status Map TLV is set appropriately to reflect the result of the operation). It is not required for this notification to include the Status Code TLV to additionally indicate success of the operation.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Additionally, the open / closed status of the ZDD's Zigbee network MAY be obtained by reading the Commissioning Status characteristic (Open / Closed Status field of the Network Status Map TLV item), as described in section 7.7.2.9.

7.7.2.7. Join Network characteristic

The Join Network characteristic can be used to instruct the ZDD join or rejoin a Zigbee network. Once the network joining has completed successfully, a notification SHALL be sent on the Commissioning Status characteristic containing all TLVs listed in Table 43 (where Joined Status field of the Network Status Map TLV shows success of the operation). In case joining was unsuccessful for any reason, the ZDD SHALL notify the ZVD of the failure by sending a notification on the Commissioning Status characteristic containing the Status Code TLV with error code "Join Network error" and domain code 0x02 "Join Network Operation" (see section 7.7.2.4.6).
The Join Network characteristic can only be written in case the ZDD is currently not a member of a Zigbee network (i.e. the Joined Status field of the Network Status Map TLV item holds the value "No network"), otherwise the ZDD SHALL notify the ZVD of any failure to join a network by sending a notification on the Commissioning Status characteristic containing the Status Code TLV with error code "Join Network error" and domain code 0x02 "Join Network Operation" (see section 7.7.2.4.6).
The characteristic value for writing consists of one or more TLV items which are specified in section 7.7.2.4.

Once the network settings have been applied, the device, if configured to operate as a ZC or ZR, SHALL commence sending out periodic Link Status messages as per section 3.6.3.4 of [R12] (to signal the node's network presence to neighboring nodes).

The Join Network characteristic can be written to in multiple ways, with a mixture of parameters. It is not possible for the ZVD to set the device type (e.g. router, end device) of the joining ZDD, instead the joining ZDD will use its inherent device type. The following sections illustrate the parameter usage for the various joining methods.

7.7.2.7.1. Common TLV item usage in Join Network

This section describes the TLV item usage within the Join Network characteristic that is common for all joining methods.

The mandatory *Joining Method* TLV item specifies the joining method used (see section 7.7.2.4.4).

In case the Admin Key is not provided to the ZDD when joining a centralized security network, then the ZDD SHALL derive the Admin Key from the TCLK (as described in section 6.3.2.1). In case the Admin Key is not provided to the ZDD when joining a distributed security network, then a ZVD MAY NOT be able to setup an Admin level authorization session with this ZDD.

When the Link Key is provided by the ZVD for either OOB commissioning or MAC association of the ZDD, that key SHALL be added to the ZDD's set of pre-programmed link keys as the preferred link key that is used for joining. When the Link Key is not provided to the ZDD, a pre-programmed link key can be used when joining the network. If no pre-programmed link keys are

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

available, then the default keys SHALL be used: The distributed security link key will be used when joining a distributed network and ZigBeeAlliance09 key or the install code based key will be used when joining a centralized security network.

7.7.2.7.2. Joining Method: MAC association

This section describes the TLV item usage and ZDD behavior in case the Joining Method TLV item is set to join through MAC association. The behavior SHALL be as specified in Table 38 and [joining-method-active-scan-results-usage].

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 38. Joining Method: MAC association ZDD behavior

Network Channel	Extended PAN ID	Behavior
Provided	Provided	The ZDD SHALL do an IEEE 802.15.4 beacon request on the provided channel. From the received beacons it SHALL select the network that matches the provided extended PAN ID for joining.
Not provided	Provided or omitted	The ZDD SHALL perform an active scan to find a suitable network for joining. Active scan SHALL be done on the requested network channels, or (when omitted) on all channels. When Extended PAN ID is provided the ZDD SHALL look for a matching open network from the scan results and it SHALL try to join that network. When Extended PAN ID is ommitted the ZDD SHALL look for an open network and it SHALL try to join that network.
Provided	Omitted	The ZDD SHALL perform an active scan to find a suitable network for joining. Active scan SHALL be done on the requested network channels, or (when omitted) on all channels. The scan results are used as specified in [joining-method-active-scan-results-usage]. The ZDD SHALL look for an open network and it SHALL try to join that network.

[10]

Table 39 below is listing the use of TLV items for the joining method of MAC association.

Table 39. TLV usage for Joining Method: MAC association

Required TLV items	Optional TLV items	Invalid TLV items
Joining Method	Extended PAN ID Network Channel Link Key NWK Address Trust Center Address Admin Key	Short PAN ID Network Key Device Type IEEE Address Network Status Map NWK Update ID NWK Active Key Seq Number

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.7.2.7.3. Joining Method: Rejoin

This section describes the TLV item usage and ZDD behavior in case the Joining Method TLV item is set join using (secure) NWK rejoining procedure.

Table 40. TLV usage for Joining Method: Rejoin

Required TLV items	Optional TLV items	Invalid TLV items
Joining Method Extended PAN ID	Network Channel Trust Center Address Admin Key Network Key Link Key	Short PAN ID NWK Address Device Type IEEE Address Network Status Map NWK Update ID NWK Active Key Seq Number

Use-cases and examples

This section provides a few example combinations of TLVs and the expected outcome regarding the ZDD's behavior.

Joining Method TLV and Extended PAN ID TLV are mandatory for all rejoins.

To perform a network-layer (secure) rejoin, the ZVD additionally includes the Network Key TLV. In addition, it may include the admin key to claim ownership of the ZDD. It may further include the Trust Center Address and Link Key TLVs (either provisional or permanent), in which case the ZDD adds a corresponding entry to the apsDeviceKeyPairSet table (pre-configured trust center link-key). Note: A reciprocal entry could be made on the trust center using the Manage Joiners characteristic (if the trust center is also a ZDD) or a manufacturer-specific out-of-band method. If only a Link Key TLV is included, it is expected to be a provisional key to be used by the ZDD during future trust center rejoins until a permanent key is established between ZDD and TC by other means.

To perform a trust center rejoin, the ZVD omits the network key. It may include the admin key to claim ownership of the ZDD. It may optionally include the Trust Center Address and Link Key TLVs (either provisional or permanent), in which case the ZDD adds a corresponding entry to the apsDeviceKeyPairSet table (pre-configured trust center link-key) and will use this key during the trust center rejoin (and beyond, if the key is a permanent key). If a Link Key is provided, but no Trust Center Address, then the provisional link key is set as the ZDDs pre-configured trust center link key and will be used during the trust center rejoin. If no Link Key TLV is included, the ZDD will use its built-in pre-configured trust center link-key or pre-shared secret, which are typically derived from installation codes or short device setup codes (passcodes) for PAKE protocols.

The trust center address is neither expected nor required to be known up-front. When specified, it will only be used as a key for adding a link-key to the apsDeviceKeyPairSet table. Typically, it will not be used by the ZDD to match the real trust center address (as learned in course of the network rejoin procedure) and the one provided to the ZDD via a write to the Join Network characteristic.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.7.2.7.4. Joining Method: Out-of-band Commissioning

This section describes the TLV item usage and ZDD behavior in case the Joining Method TLV item is set to the Out-of-band Commissioning method that is used to set up a ZDD with all the network parameters required to join a specific network.

The ZDD SHALL apply the provided network settings and join the network using these settings. The Network Channel ChannelList Structure SHALL consist of exactly one channel, otherwise the ZDD SHALL report an error to the ZVD by sending a notification on the Commissioning Status Characteristic containing the Status Code TLV with error code "Join Network error" and domain code 0x02 "Join Network Operation" (see section 7.7.2.4.6).

In case the NWK Address TLV is omitted, then the device will assign a random 16-bit network address to itself.

In case the Trust Center TLV is omitted, then the device will set the trust center address to 0xFFFFFFFFFFFFFFFF (for distributed security).

In case the NWK Update ID TLV is omitted, then the device will assume that the NWK Update ID is 0.

In case the NWK Active Key Seq Number TLV is omitted, then the device will assume that the NWK Active Key Seq Number is 0.

If the ZDD was instructed to join a centralized security network via out of band commissioning and the TCLK provided was a provisional one, the ZDD SHALL attempt to update the TCLK (as defined in section 10.2.5 of [R10]). If the procedure is not successful, the ZDD SHALL NOT leave the network; it SHALL attempt to update the TCLK when a connection with the TC is available.

Once the network settings have been applied, the ZDD, if configured to operate as a ZC or ZR, SHALL commence sending out periodic Link Status messages as per section 3.6.3.4 of [R12] (to signal the node’s network presence to neighboring nodes).

The ZDD SHALL broadcast a Device Announcement using ZDO (ZDO Device_annce, [R12] 2.4.3.1.11) once it has successfully joined the Zigbee network using the out-of-band commissioning method.

Note: The out-of-band commissioning method is similar to the NWK rejoin with key method. One significant difference is that the secure rejoin option verifies connectivity to at least one potential parent router and as such involves an active scan and a rejoin request/response exchange, which also verifies the network key.

Note 2: A ZVD provisioning a ZDD to join a distributed security network will not be able to access the ZDD using admin authorization if the key is not provided during the provisioning session.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 41. TLV usage for Joining Method: Out-of-band Commissioning

Required TLV items	Optional TLV items	Invalid TLV items
Joining Method Extended PAN ID Short PAN ID Network Channel Network Key	Trust Center Address Link Key NWK Address NWK Update ID NWK Active Key Seq Number Admin Key	Device Type IEEE Address Network Status Map

7.7.2.8. Leave Network characteristic

The Leave Network characteristic can be used to request the ZDD to leave the Zigbee network of which it is currently a member. The payload that is written to this characteristic is defined in Table 42

Table 42. Leave Network characteristic payload

Field Name	Data Type	Octet #	Range	Description
Remove Children	boolean	0	false / true	Set this parameter to TRUE to ask the device to also remove its child devices, if any. Otherwise, set this parameter to FALSE.
Rejoin	boolean	1	false / true	Set this parameter to TRUE to ask the device to rejoin the network. Otherwise, set this parameter to FALSE.

The *Remove Children* and *Rejoin* parameters have the same behavior as in the ZDO *Mgmt_Leave_req* command (see section 2.4.3.3.5 of [R12]).

The action is only accepted in case the ZDD is currently a member of a Zigbee network, otherwise the ZDD SHALL notify the ZVD of any failure to leave the network by sending a notification on the Commissioning Status Characteristic containing the Status Code TLV with error code "Leave Network error" and domain code 0x04 "Leave Network Operation" (see section 7.7.2.4.6).
Once the device has left the network, the device SHALL send a notification on the Commissioning Status characteristic containing all TLVs listed in Table 43 where success of this operation can be derived from the Joined Status field of the Network Status Map TLV. It is not required for this notification to include the Status Code TLV to additionally indicate success of the operation.
The Leave Network operation SHALL ensure that network specific parameters, including the keys, (as provided by Form Network or Join Network characteristics) SHALL be erased and forgotten.

7.7.2.9. Commissioning Status characteristic

The Commissioning Status characteristic is used to retrieve the network status of the ZDD as well as to report status codes from the ZDD to the ZVD. This read-only / notification characteristic

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

consists of one or more TLV items which are specified in section 7.7.2.4.

A ZDD SHALL send a notification containing the TLVs listed in Table 43 below on the Commissioning Status characteristic in case one of the following changed on the device:

- Joined Status
- Open / Closed Status - including when the remaining time the network is opened got extended.
- Network address
- Network channel

The Commissioning Status characteristic payload (for both read response and notification) SHALL always include the following TLV types:

Table 43. TLV usage for Commissioning Status

TLV	Usage
Network Status Map	SHALL always be included in the Commissioning Status characteristic payload.
Network Channel	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded. The value of this TLV item SHALL be set to a channel mask with single bit set indicating the current operating channel as used by the Zigbee network.
Extended PAN ID	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded.
Short PAN ID	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded.
NWK Address	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded.
IEEE Address	SHALL always be included in the Commissioning Status characteristic payload.
Trust Center Address	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded. If the Zigbee network is a distributed security network, the Trust Center Address SHALL be set to 0xFFFFFFFFFFFFFFFF
Device Type	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded.
NWK Update ID	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded.
Active Key Seq Number	This TLV item SHALL be included in case the ZDD is currently joined to a Zigbee network, otherwise it SHALL be excluded.

The Commissioning Status characteristic payload (for both read response and notification)

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

SHALL NOT include the following TLV types:

- Network Key
- Link Key
- Admin Key
- Basic Key

The commissioning status characteristic SHALL be used to send a notification containing the Status Code TLV as and when required to do so by the definition of the Zigbee Direct Commissioning Service. When sending a Status Code TLV an Extended Status Code TLV containing additional (manufacturer specific) information MAY also be sent. The Status Code TLV as well as the Extended Status Code TLV SHALL only be included in a notification. When sending a notification with only a Status Code TLV, the Network Status Map and IEEE Address fields MAY be excluded.

7.7.2.10. Manage Joiners characteristic

The Manage Joiners characteristic is optional and can be used in case when a ZDD implements a TC role. In this case the characteristic is used to manage Zigbee Provisional Link Keys for potential joiners on the TC side.

A possible use case of this characteristic can be found in the following diagram. In this case both a Zigbee TC and a Joiner are ZDDs and the Commissioner is a ZVD.

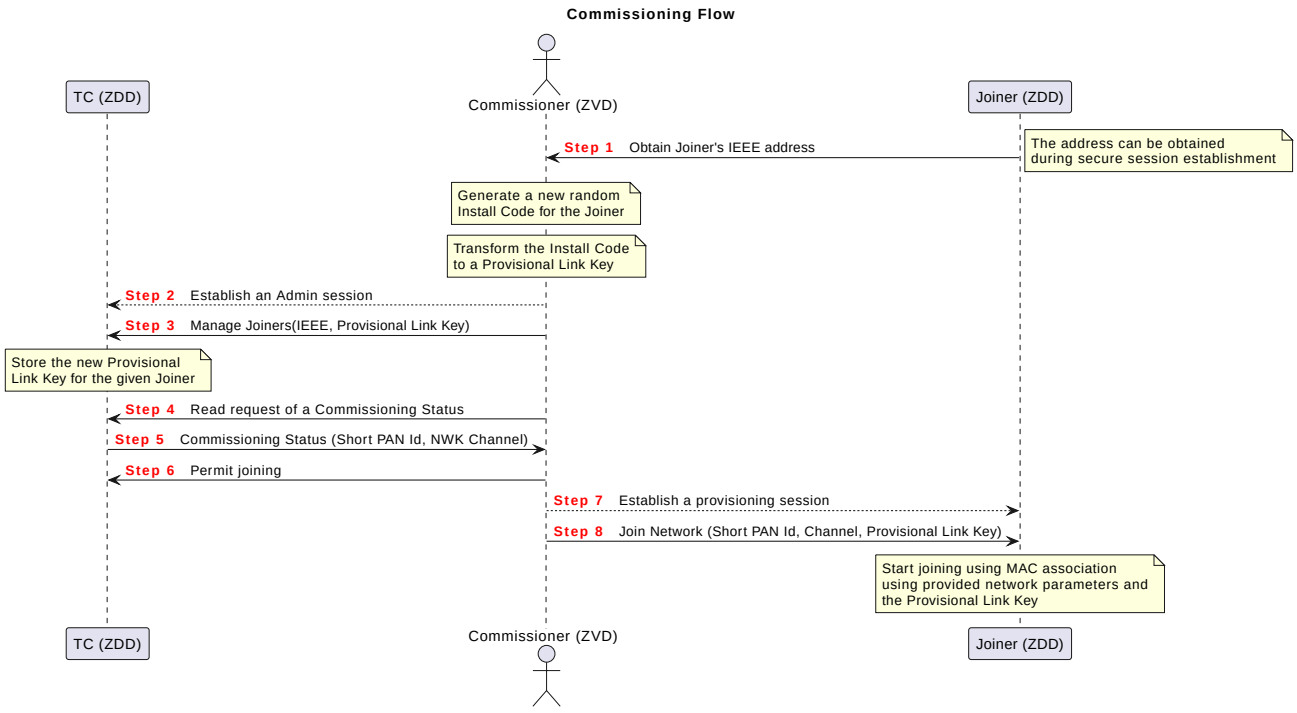


Figure 12. Commissioning use case TC ZDD

Once a Manage Joiners command has been completed, a ZDD SHALL send a notification on the Commissioning Status characteristic with a Status Code TLV indicating Success or Manage Joiners error code, and domain code 0x05 "Manage Joiners Domain" as outlined in section 7.7.2.4.6.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

If a non-TC ZDD receives a Manage Joiner Command, it SHALL send a Manage Joiners error notification on the Commissioning Status characteristic.

The Manage Joiners characteristic can be written to in multiple ways, with a mixture of parameters. The following table illustrates the parameter usage for the various commands.

7.7.2.10.1. Local TLV IDs for Manage Joiners characteristic

The TLV IDs that are used within the Manage Joiners characteristic of the Zigbee Direct Commissioning Service are defined in Table 44.

Table 44. Zigbee Direct Manage Joiners TLV IDs

Type ID	Name	Zigbee Data Type	Description
0x00	Provisional Link Key	key128	Specifies the Zigbee Link Key.
0x01	IEEE Address	EUI64	The 64-bit IEEE address of the ZDD.
0x02	Manage Joiners Command	uint8	Specifies a command should be executed. See section 7.7.2.10.2.
0x03-0xff	*Reserved*		

7.7.2.10.2. Manage Joiners Command TLV

Table 45. Manage Joiners Command TLV values

Key	Description
0x00	Drop all joiners' Provisional Link Keys.
0x01	Add a joiner’s Provisional Link Key.
0x02	Remove a joiner’s Provisional Link Key.

Table 46. TLV usage for Manage Joiners commands

Command	Required TLV items
Drop all joiners' Provisional Link Keys	Manage Joiners Command
Add a joiner’s Provisional Link Key	Manage Joiners Command IEEE Address Link Key
Remove a joiner’s Provisional Link Key	Manage Joiners Command IEEE Address

7.7.2.10.3. Drop All Joiners

Once this command received, a ZDD-TC SHALL drop all stored Provisional Link Keys for all

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

devices.

7.7.2.10.4. Add Joiner's Provisional Link Key

Once this command received, a ZDD-TC SHALL add the provided Provisional Link Key for a device with a given IEEE address. In case the TC has another Provisional Link Key for the device, the Key must be rewritten with the new one.

7.7.2.10.5. Remove Joiner's Provisional Link Key

Once this command received, a ZDD-TC SHALL remove a stored Provisional Link Key for the device with a given IEEE address.

If after the execution of the Remove Joiner's Provisional Link Key command no Provisional Link Key for the device with the referenced IEEE Address is on record, the ZDD SHALL send a notification on the Commissioning Status characteristic with a Status Code TLV indicating Success and domain code 0x05 "Manage Joiners Domain".

7.7.2.11. Identify characteristic

The Identify characteristic is mandatory and MAY be used to request the ZDD to identify itself.

The identify command might be useful to make sure which device is being commissioned by Installer and to avoid commissioning of wrong devices like a neigbour's bulb or a maliciuos device, created by an attacker.

A possible use case of this characteristic can be found in the following diagram.

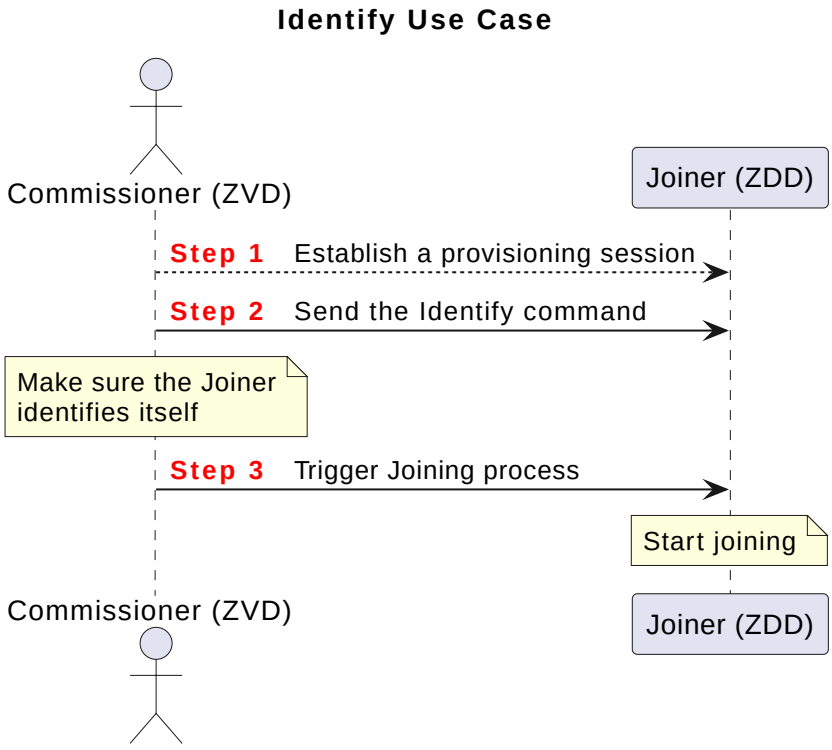


Figure 13. Identify use case

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

In case a ZDD is unprovisioned, the Identify command SHALL always be accepted and executed by the ZDD. In case the ZDD is provisioned, Basic or Admin access is required for the command execution. A provisioned ZDD SHALL send an error notification on the Commissioning Status characteristic if the command was received during a provisioning session.

In case a malformed packet was received by the Identify characteristic, the ZDD SHALL notify the ZVD of an error by sending a notification on the Commissioning Status Characteristic with the Status Code TLV indicating Success or Identify error code, and domain code 0x06 "Identify Operation" as outlined in section 7.7.2.4.6.

The Identify SHALL be processed in the same way as it is described in the Zigbee Cluster Library Specification - 075123 [R8], section 3.5.2.

If a ZDD implements more than one endpoint, it will be up to the manufacturer to decide on which endpoint to perform identifying.

The value that is written to this characteristic holds a duration of identification in seconds. Zero timeout value, written to the characteristic, SHALL stop the identification.

Table 47. Identify characteristic write values

Octets	Data type	Field name
2	uint16	Timeout

The value that is read from the characteristic contains a length of time in seconds, that the device will continue to identify itself.

Table 48. Identify characteristic read values

Octets	Data type	Field name
2	uint16	Timeout

7.7.2.12. Finding&Binding characteristic

The Finding&Binding characteristic is OPTIONAL and MAY be used to request the ZDD to start Zigbee Finding&Binding procedure. Request payload is defined in the following table.

Table 49. Finding&Binding characteristic payload

Field Name	Field size	Octet #	Description
Endpoint	1 octet	0	Number of an endpoint which is used for Finding & Binding.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

<table>
<tr><th>Field Name</th><th>Field size</th><th>Octet #</th><th>Description</th></tr>
<tr><td>Flags</td><td>1 octet</td><td>1</td><td>Flags indicating operation details.
<table>
<tr><th>Bit</th><th>Flag name</th><th>Description</th></tr>
<tr><td>0</td><td>Initiator</td><td>1 in case the device is a Finding&Binding initiator and 0 if the device is a target device.</td></tr>
<tr><td>1-7</td><td>Reserved</td><td>Reserved, set to 0</td></tr>
</table>
</td></tr>
</table>

The action is only accepted in case the ZDD is currently a member of a Zigbee network, otherwise the ZDD SHALL notify the ZVD of an error by sending a notification on the Commissioning Status Characteristic containing the Status Code TLV which is listing the Finding&Binding error code, and domain code 0x07 "Finding & Binding Domain" as outlined in section 7.7.2.4.6.

Upon receiving the request, a ZDD, being a Zigbee network member, SHALL start a Finding&Binding procedure on the selected endpoint.

Once the command has been completed, a ZDD SHALL send a notification on the Commissioning Status characteristic containing a Status Code TLV which is listing the Success error code as outlined in section 7.7.2.4.6.

7.7.3. Zigbee Direct Tunnel Service

7.7.3.1. Introduction

The Zigbee Direct Tunnel Service is a BLE GATT service that is used to transport Zigbee Network Packet Data Units (NPDUs) over the BLE interface. From a Zigbee Specification perspective, the Zigbee Direct Tunnel Service is a Trusted Link service provider. The Zigbee Direct Tunnel Service is an optional service, which may have at most one instance on a ZDD. When supported, then the service shall be instantiated as a primary GATT service with the 128-bit UUID: 8BD178FD-0000-45F4-8120-B2378BD5313F.

A ZDD that operates as Zigbee End Device SHALL NOT instantiate the Zigbee Direct Tunnel Service and SHALL have the ZDTS flag of the Zigbee Direct BLE Advertisement Extension set to false (see section 7.6.3).

7.7.3.2. Security

The Zigbee Direct Tunnel Service requires Zigbee Direct authentication for usage of any of its BLE GATT characteristics. A ZVD SHALL setup a secure session with the ZDD before it can access the ZDD's Zigbee Direct Tunnel Service (see section 6.3). Refer to Table 50 for the required authentication level per characteristic. For each characteristic in this service, Bluetooth Low

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Energy security SHALL be set to LE Security Mode 1, Level 1 (no authentication and no encryption) per [R3] Vol 3, Part C, Section 10.2.1.

In case the ZVD attempts to read or write a characteristic for which its current authentication level is insufficient, then this SHALL result in an insufficient authentication exception, which SHALL be handled by the ZDD as described in section 6.4.7.4.

The ZDD SHALL NOT send BLE notifications or indications to a ZVD on a characteristic for which the ZVD's current authentication level is insufficient.

7.7.3.3. Zigbee Direct Tunnel Service characteristics

Table 50. Zigbee Direct Tunnel Service characteristics

Characteristic Name	Characteristic UUID	Supported BLE Procedures	BLE Format Type[6]	Authentication Level
ZDTS NPDU	8BD178FD-0001-45F4-8120-B2378BD5313F	• Write Request • Indication	struct	• Provisioning session (with restrictions as described in section 7.7.3.6.2) • Basic level authorization

7.7.3.4. Zigbee Direct Tunnel Service NPDU Characteristic

The Zigbee Direct Tunnel Service NPDU characteristic is used to transport Zigbee Network Packet Data Units (NPDU's) over BLE. The ZVD (BLE Central) sends NPDU Messages via BLE Write Requests (write with response). The ZDD (BLE Peripheral) sends NPDU Messages BLE Indications (transfer with acknowledgement). Towards the Zigbee stack, the Zigbee Direct Tunnel Service appears as a MAC interface representing a Trusted Link.

The following sequence diagram shows the non-normative flow and communication between a ZVD, a ZDD and another Zigbee node as an example of using the Zigbee Direct Tunnel Service (APS acknowledgements are not shown in this diagram for clarity):

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

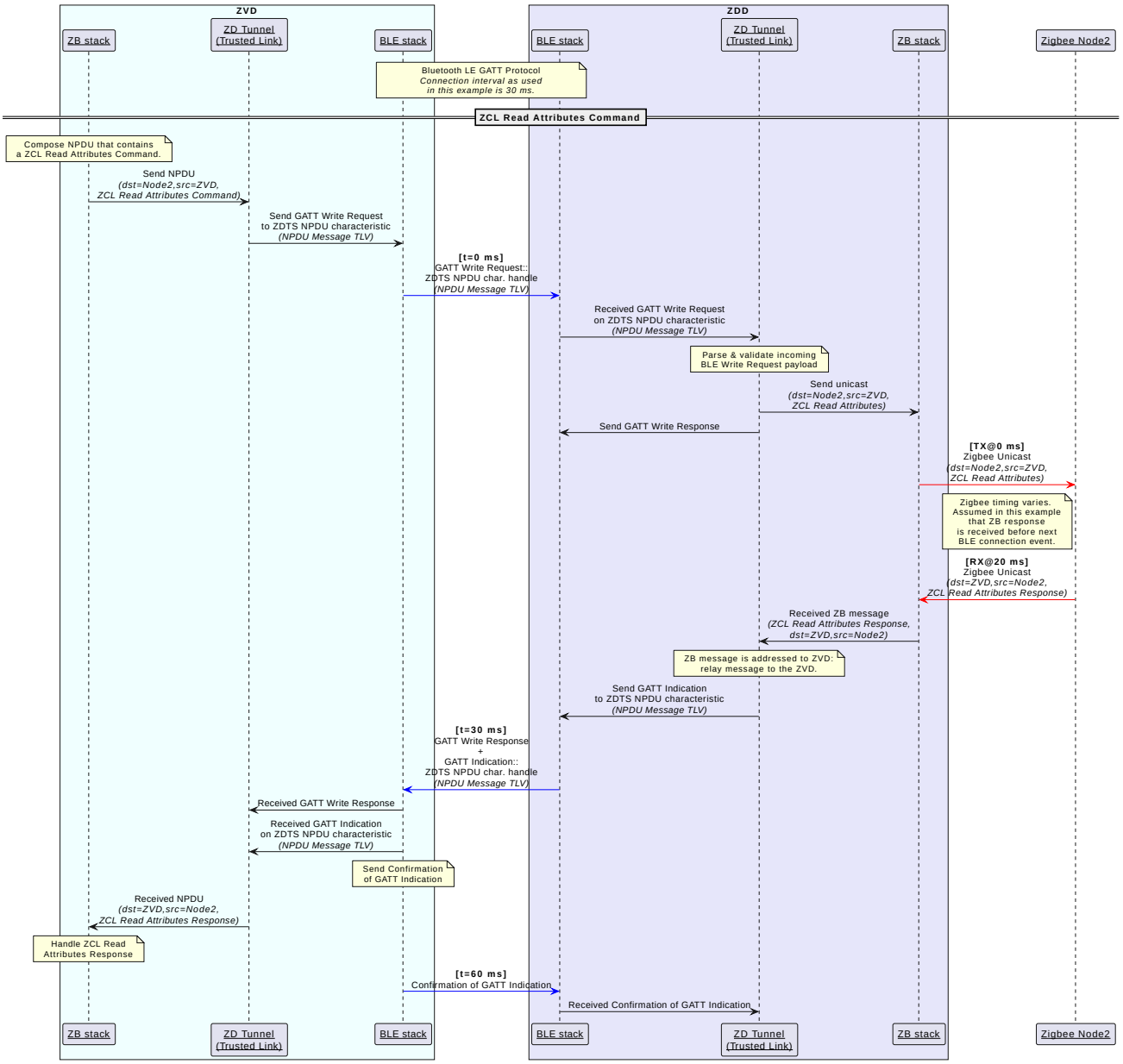


Figure 14. ZVD ZDD message flow sequence

7.7.3.5. Payload format

The payload of the Zigbee Direct Tunnel Service NPDU characteristic SHALL be formatted as follows:

Table 51. Zigbee Direct Tunnel Service NPDU characteristic format

Octets: Variable	**Variable**	**...**	**Variable**
TLV 1	TLV 2	...	TLV *n*

The payload of the Zigbee Direct Tunnel Service NPDU characteristic SHALL include one or more Type-Length-Value data elements (TLVs), as long as the Zigbee Direct Tunnel Service characteristic payload fits the BLE connection's negotiated MTU size. The TLV items are formatted according to the Zigbee TLV format (see [R12], Annex I "Zigbee TLV Definitions And Format"), with the addition that multiple TLVs of the Tunnel Service NPDU characteristic SHALL

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

be allowed.

In case multiple TLVs are included, then each TLV shall be handled sequentially as if it was received via an individual request (with a best-effort approach).

In the scope of the Tunnel characteristic, the following Type IDs have been defined:

Table 52. ZDTS NPDU characteristic Type IDs

Type ID	**Name**
0x00	NPDU Message

7.7.3.5.1. TLV Length Field

The length field contains the size (in octets) of the data that is included in the TLV value field **minus one**.

Table 53. Length field

Range:	0x00 to 0xF1[11]

7.7.3.5.2. TLV Value Field

The value field contains the data of the TLV item.

7.7.3.6. NPDU Message TLV

This TLV item is used to transfer an NPDU from ZVD to ZDD, and vice versa. This is accomplished by a GATT Write Request, or GATT Indication respectively.

7.7.3.6.1. Value Field Format

The value field of the NPDU Message TLV item SHALL be formatted as follows:

Table 54. Value field

Octets: 1	**1**	**Variable**
Flags	NPDU Length	NPDU

The additional length field as part of the value field makes this primitive future proof: Additional fields can be added in the future following the NPDU field, thus allowing a legacy ZDD to ignore any additional fields that might have been appended to the NPDU Message TLV.

The Table 55 specifies the sub-fields (parameters) of the NPDU Message TLV:

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Table 55. Sub fields of NPDU Message TLV

Name	Type	Valid Range	Description
Flags	map8	*See description*	The flags for transfer of the NPDU, formatted as follows: **Bit: 0** | **1-7** SecurityEnable | *Reserved* The SecurityEnable flag is used to indicate whether the NPDU is to be assumed as being NWK encrypted.
NPDU Length	uint8	0x00 - 0x62[12]	The number of octets comprising the NPDU to be transferred.
NPDU	Set of Octets	-	The set of octets comprising the NPDU to be transferred. Holds a Zigbee NWK frame according to the format that has been specified in section 3.3.1 of the Zigbee specification [R12]. The NPDU SHALL NOT include the auxiliary frame header and the security sub-field of the NPDU's frame control field SHALL be set to 0.

7.7.3.6.2. Security

The content of the ZDTS NPDU characteristic SHALL be encrypted as described in section 6.4.1. This means that a successful secure session establishment must have taken place.

When a secure session establishment has been taken place for 'ZVD Provisioning', NPDU messages with SecurityEnable flags equal to True SHALL be dropped by the Zigbee Direct Tunnel Service. In all other cases ('ZDD Provisioning', and 'Authorized'), the Zigbee Direct Tunnel Service SHALL accept NPDU messages independent of the SecurityEnable flag.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.7.4. Functional Description

7.7.4.1. Effect on Receipt of an NLME-TRUSTEDLINK-POSTPROCESSING.request

On receipt of an NLME-TRUSTEDLINK-POSTPROCESSING.request, a device implementing this specification SHALL construct an NPDU Message TLV according to section 7.7.3.6, with

1. the SecurityEnable sub-field of the Flags field set to the value of the assumeSecurity parameter of the request;
2. the NPDU Length field set to the number of octets contained in the npdu parameter of the request; and
3. the NPDU field set to the value of the npdu parameter of the request.

The device SHALL then issue an NLME-TRUSTEDLINK-POSTPROCESSING.confirm with the linkNPDU parameter of this confirmation containing the NPDU Message TLV.

If for any reason constructing the NPDU Message TLV fails, the device SHALL issue a confirmation with the linkNPDU parameter set as empty. Such reasons might include an excessive NPDU length beyond the MTU size supported by the device or transitional errors like temporary unavailability of resources required to complete the request.

7.7.4.2. Effect on Receipt of an NLME-TRUSTEDLINK-PREPROCESSING.request

On receipt of an NLME-TRUSTEDLINK-PREPROCESSING.request, a device implementing this specification SHALL regard the value of the linkNPDU parameter as an NPDU Message TLV according to section 7.7.3.6.

If the TLV is malformed the device SHALL issue an NLME-TRUSTEDLINK-PREPROCESSING.confirm with the assumeSecurity parameter set to FALSE and the npdu parameter empty. It SHALL stop preprocessing in this case.

Otherwise, if the TLV is valid, the device SHALL issue an NLME-TRUSTEDLINK-PREPROCESSING.confirm, with

1. the assumeSecurity parameter set to the value of the SecurityEnable sub-field of the Flags field of the NPDU Message TLV; and
2. the npdu parameter set to the sequence of octets contained in the NPDU field of the NPDU Message TLV and further delimited by the NPDU Length field of that TLV.

7.7.4.3. Joining centralized security network

Before any new ZVD can be provisioned onto a centralized security Zigbee network the network's coordinator or one or more of its routers SHALL be caused to issue an *Mgmt_Permit_Joining_req* command, as defined by Zigbee specification [R12] or later, such that the network's constituent nodes respond with Beacons with the Association Permit bit set when requested, see Figure 15.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

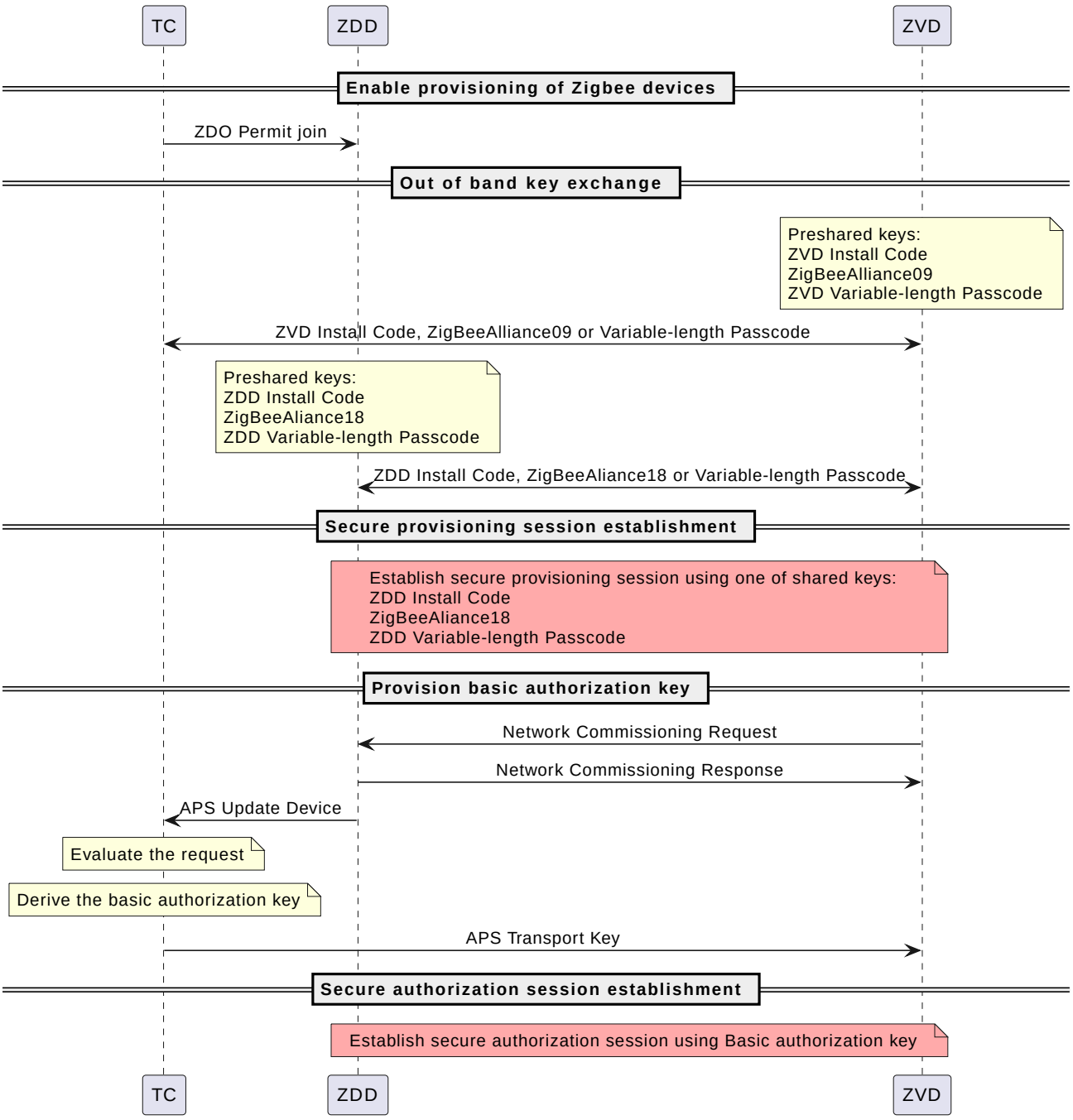


Figure 15. ZVD Joining Centralized Security Network

To request the access to a centralized security network a new ZVD SHALL establish secure provisioning session as described in section 6.3.1.

Once the session is established the ZVD SHALL construct an NPDU message containing a Network Commissioning Request command, and with the SecurityEnable flag set to False. In line with [R12], the ZVD SHALL use a randomly generated valid value for *nwkAddress*, and SHALL set the Network Commissioning Type to 0x00 - Initial Join. The following TLVs SHALL be appended to the command:

- The Joiner Encapsulation Global TLV containing:
  - the Supported Key Negotiation Methods Global TLV,
  - the Fragmentation Parameters Global TLV,

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

- the Device Capability Extension Global TLV (see [device_capability_extension_global_tlv]).

The Trust Center SHALL evaluate the APS Update Device message against the policy defined by the administrator of the network and provide the ZVD with the basic authorization key and the corresponding key sequence number of the active Zigbee NWK key used to derive the authorization key. The authorization key is transported using APS Transport Key message. Transport of the authorization key is secured by one of the Pre-shared keys (ZVD Install Code or ZA09 key) shared between the ZVD and the Trust Center.

The ZVD that received the authorization SHALL establish a new secure authorization session over the same BLE link using the basic authorization key as described in section 6.3.2. The joining process SHALL be finalized over the newly created authorization session.

7.7.4.4. ZVD operating as trust center on a centralized network

A ZVD MAY form a centralized network on which it operates in the capacity of the trust center. In order to do so, it MAY first out-of-band commission a first router on the network. Besides all required TLV items listed in Table 41 it MAY also include the Trust Center Address TLV item with the value set to its (the ZVD's) own EUI-64; it MAY also include the Link-Key TLV item to establish a unique link-key between itself (ZVD) and the first router (ZDD).

The ZVD TC MAY select channel and PAN-ID information at random. Alternatively, the ZVD MAY also first instruct the ZDD to form a centralized or distributed security network, which will result in the ZDD performing active and energy scans, and extract the resulting channel and PAN-ID for its own reuse in out-of-band commissioning the first router as outlined above.

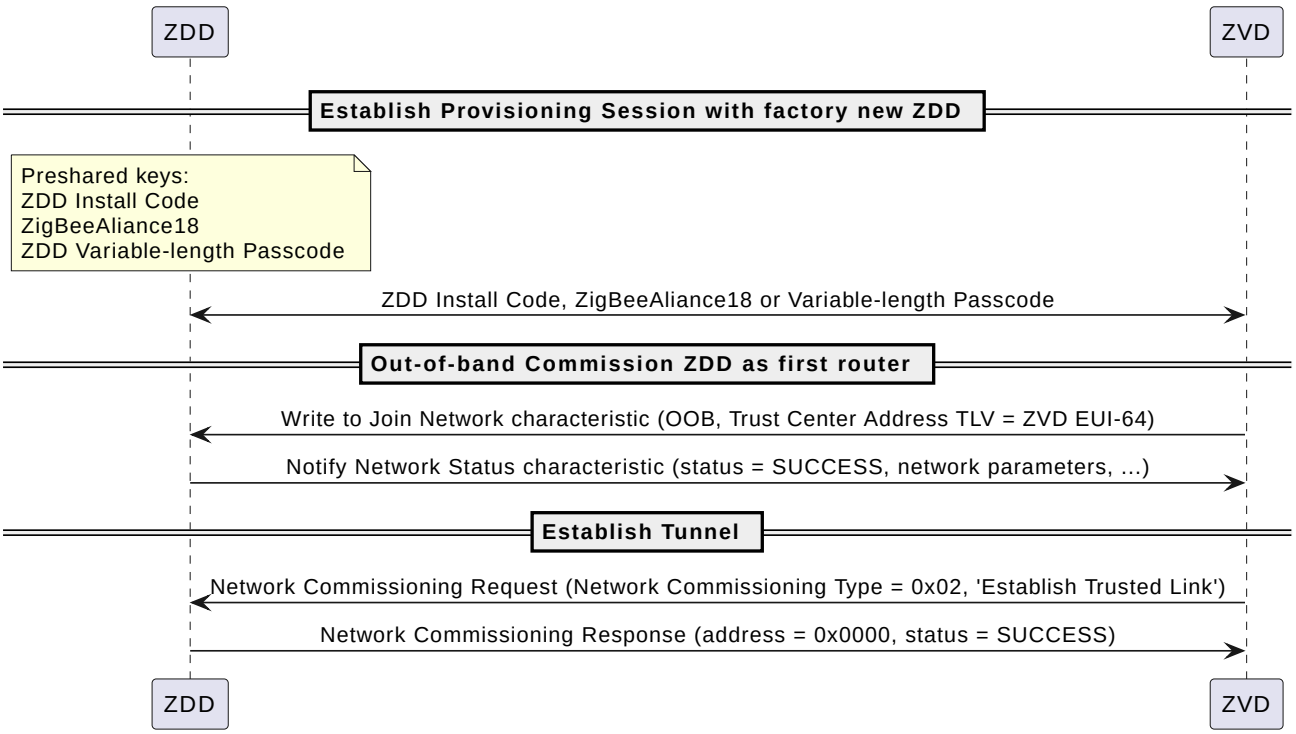


Figure 16. ZVD Operating as Trust Center on a Centralized Security Network

To commission the factory-new ZDD as first router, a factory-new ZVD SHALL establish a secure provisioning session as described in section 6.3.1. Once the session is established the ZVD MAY

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

write to the Join Network characteristic with the Join Method TLV indicating out-of-band commissioning as sketched above.

After out-of-band commissioning has succeeded the ZVD SHALL construct an NPDU message containing a Network Commissioning Request command, and with the SecurityEnable flag set to true. In line with [R12], the ZVD SHALL use 0x0000 for *nwkAddress*, and SHALL set the Network Commissioning Type to 0x02 - Establish Trusted Link. The following TLVs SHALL be appended to the command:

- The Joiner Encapsulation Global TLV containing:
  - the Supported Key Negotiation Methods Global TLV,
  - the Fragmentation Parameters Global TLV,
  - the Device Capability Extension Global TLV (see [device_capability_extension_global_tlv]).

7.7.4.5. Joining distributed security network

Before any new ZVD can be provisioned onto a distributed security Zigbee network one or more of the network's routers SHALL be caused to issue an Mgmt_Permit_Joining_req command as defined by Zigbee specification [R12] or later such that the network's constituent nodes respond with Beacons with the Association Permit bit set. The distributed security Zigbee network SHALL be open to accept new devices using ZDO Permit Join message as defined by Zigbee specification [R12] before any new ZVD can be provisioned, see Figure 17.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

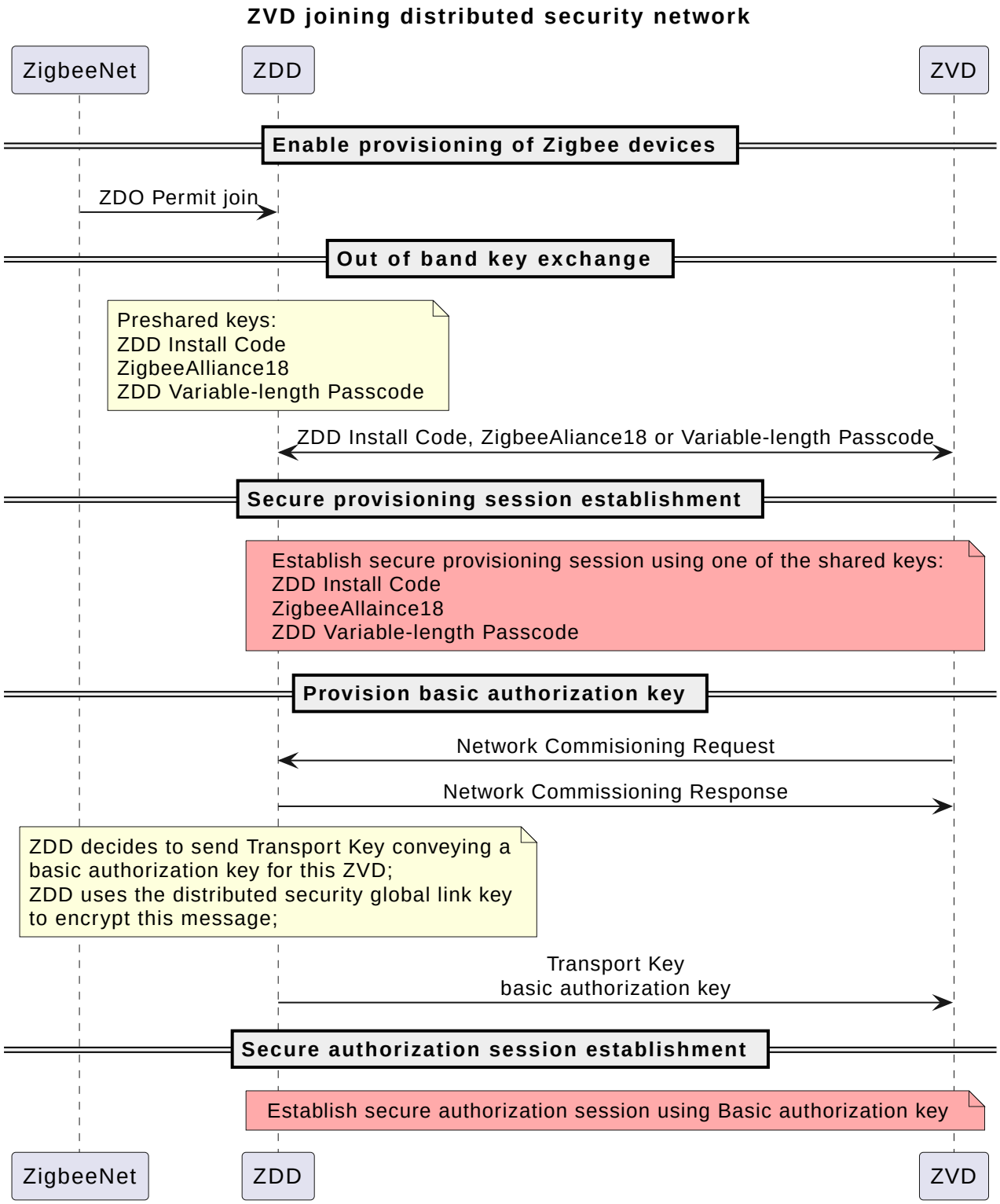


Figure 17. ZVD joining distributed security network

To request the access to a distributed security network a new ZVD SHALL establish secure provisioning session using one of the Pre-shared keys as described in section 6.3.1. Once the session is established the ZVD SHALL construct an NPDU message containing a Network Commissioning Request command, and with the SecurityEnable flag set to False. In line with [R12], the ZVD SHALL use a randomly generated valid value for *nwkAddress*, and SHALL set the Network Commissioning Type to 0x00 - Initial Join. The following TLVs SHALL be appended to the command:

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

- The Joiner Encapsulation Global TLV containing:
  - the Supported Key Negotiation Methods Global TLV,
  - the Fragmentation Parameters Global TLV,
  - the Device Capability Extension Global TLV (see section 11.12).

If open to accept new Zigbee devices, the ZDD SHALL derive and provide the ZVD with the basic authorization key. The authorization key is transported using APS Transport Key message and is secured by distributed security global link key. The ZVD is also provided with the key sequence number of the Zigbee NWK key used to derive the basic authorization key.

The ZVD that received the authorization SHALL establish a new authorization session over the same BLE link using the basic authorization key as described in section 6.3.2. The joining process SHALL be finalized over the newly created authorization session.

7.7.4.6. ZVD Rejoining

For rejoining, the ZVD SHALL use the Basic Key to establish the authorization session with the ZDD. It SHALL verify that the NWK key sequence number received in message 2, see section 6.5.1.2.3, corresponds with the sequence number provisioned together with the Basic Key. If it does not, the ZVD has detected a network key rotation that occurred while it was off network and it needs to obtain a new trust center link-key. The session that has just been established is restricted in respect to the traffic that can be exchanged through the tunnel service. The authorization granted in this session is sufficient to renew the Basic Authorization key, but does not allow other operations. Refer to Limited Authorization Session section 9.1 for details.

In case of TC Rejoin ([R12] 4.6.3.3.2) of the ZVD (ZVD provisioning - the ZVD is not factory-new, but likely has missed a roll of the network key), the ZVD SHALL construct an NPDU message containing a Network Commissioning Request command, and with the SecurityEnable flag set to False. In line with [R12], the ZVD SHALL set the Network Commissioning Type to 0x01 - Rejoin. The following TLVs SHALL be appended to the command:

- a Joiner Encapsulation Global TLV containing:
  - the Supported Key Negotiation Methods Global TLV,
  - Fragmentation Parameters Global TLV,
  - the Device Capability Extension Global TLV (see [device_capability_extension_global_tlv]).

In case of secure rejoin of the ZVD (authorization session, ZVD has all the correct network settings) the ZVD SHALL construct an NPDU message containing a Network Commissioning Request command, and with the SecurityEnable flag set to True. In line with [R12], the ZVD SHALL set the Network Commissioning Type to 0x01 - Rejoin. The following TLVs SHALL be appended to the command:

- a Joiner Encapsulation Global TLV containing:
  - the Supported Key Negotiation Methods Global TLV,
  - Fragmentation Parameters Global TLV,

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

- the Device Capability Extension Global TLV (see [device_capability_extension_global_tlv]).

The ZVD SHALL then await and process the Network Commissioning Response in-line with the requirements set forth in [R12].

7.7.4.7. ZVD Forming a network

Depending on the supported device type (ZC, ZR, ZED), the ZVD MAY be capable of forming a distributed or centralized security network. A ZVD wishing to form a network SHALL come up with all the parameters required for the network of the requisite type, including the Zigbee channel. In addition, in case of centralized security network, the ZVD SHALL come up with all parameters a Zigbee Direct aware Coordinator needs to establish a network.

To request an unprovisioned ZDD to join its network, a new ZVD SHALL establish a secure provisioning session using "ZigbeeAlliance18" key as described in section 6.3.1.

Once the session is established the ZVD SHALL use the Zigbee Direct Commissioning Service as outlined in section 7.7.2 to instruct the ZDD to join the newly formed network out of band (without check for nearby beacons).

On success of this operation the ZVD SHALL close the provisioning session. If the ZVD wants to continue interacting with ZDD the ZVD SHALL establish a new authorization session using the admin or basic authorization key as described in section 6.3.2.

7.7.4.8. Additional Requirements on a ZDD

If a ZVD for whatever reason does not indicate that it is a Zigbee Direct Virtual Device through the Device Capability Extension Global TLV, the end result would likely be that the TrustCenter would send the Network Key instead of the Basic Key to the ZVD using an APS Transport Key command. This is undesired. For this reason, a ZDD SHALL drop NPDU messages that include a Network Commissioning Request command where the Device Capability Extension Global TLV is missing, or when this TLV has its Zigbee Virtual Device bit set to '0'.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

8. ZVD Specification

8.1. Introduction

A Zigbee Virtual Device (ZVD) is a Bluetooth Low Energy device performing a certain role on a Zigbee network or with a Zigbee Direct Device (ZDD). Fundamentally a ZVD can play two distinct roles: First a Commissioning ZVD that is intended for commissioning of ZDDs only, and secondly a ZVD-TS that additionally is able to participate in the Zigbee Network the ZDD is part of via the Tunnel Service.

Examples of a typical ZVD are a smart phone or smart speaker.

Unlike a typical Zigbee device, a ZVD is - as its name suggests - a virtual device which is not tied to a specific hardware, but instead its main dependency on its operating environment is the presence of a BLE interface as described in section 7.4. As per the definition in [R11], a ZVD is to be considered either as a User Interface Component, or as a Component with simple or complex next higher layer interface.

In case a ZVD is not supporting the Tunnel Service (Commissioning ZVD) it SHALL be implemented as a type 1 User Interface Component, a type 2 User Interface Component, or a Component. In the latter two cases it MAY make use of a certified Zigbee Direct Commissioning Component as described in section C.1 with either a simple or a complex interface.

In case the ZVD is supporting the Tunnel Service (ZVD-TS) it SHALL be implemented either as a type 2 User Interface Component, or a Component. As before it MAY make use of a certified Zigbee Direct Commissioning Component as described in section C.1 with either simple or complex interface and in addition it SHALL have a depencency on a a certified Zigbee Stack Component with a complex interface as specified in [R12].

8.2. BLE Network Parameters

A ZVD has control over some of the parameters for Bluetooth Low Energy that are relevant for the Zigbee Direct services. In general - for both ZVD and ZDD - these are described in section 7.4[13].

The ZVD SHALL perform ATT_MTU negotiation with a ZDD with a recommended MTU size of *RecommendedAttMtuSize* octets or more, but no less than *MinAttMtuSize*. Note that some execution environments for ZVDs/ZDDs might negotiate the ATT_MTU size automatically, without any means for implementers of ZVDs/ZDDs to influence the protocol flow or negotiation details. ZVDs/ZDDs running in such environments are compliant to this specification, as long as the MTU size exceeds *MinAttMtuSize*.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

8.3. Functional description

The ZVD MAY perform one of the following roles :

- Commissioning client
- Any of the Zigbee Roles [R3] through the Tunnel Client:
  - Zigbee End Device
  - Zigbee Router
  - Zigbee Coordinator / Trust Center

To perform any of these Zigbee roles, to function as a Tunnel Client the ZVD SHALL support a Zigbee stack. A typical ZVD architecture with Zigbee Stack is shown below in Figure 18 (informative):

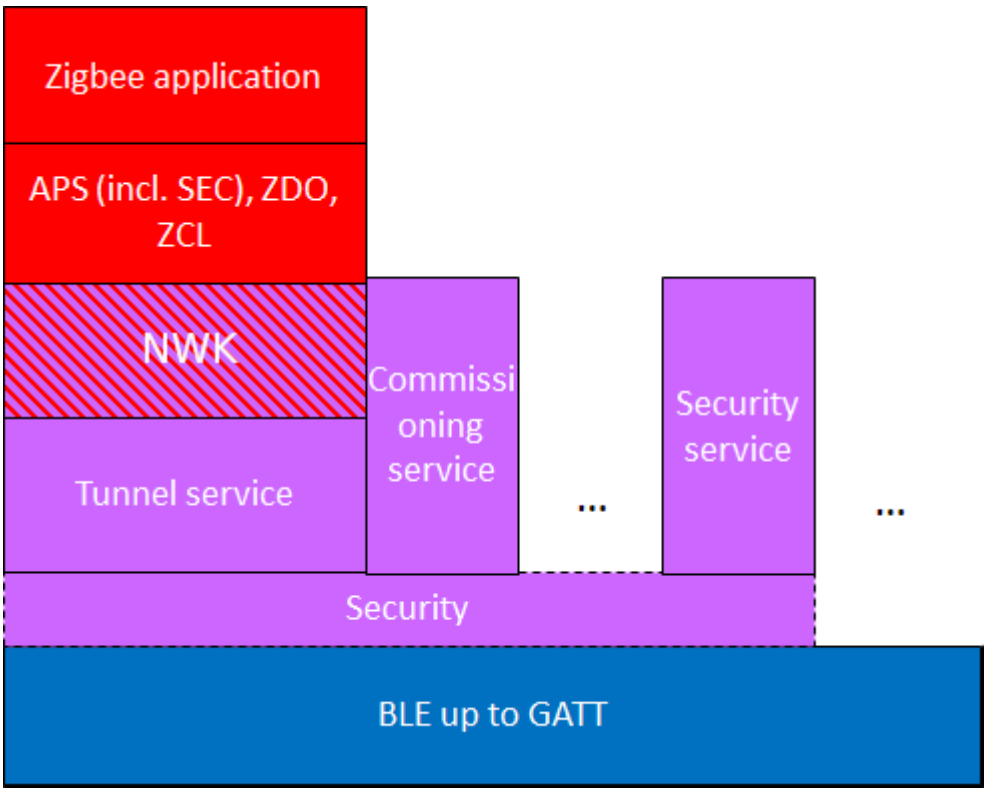


Figure 18. Zigbee Direct Virtual Device (ZVD) architecture

The following steps are executed by the ZVD, all using Bluetooth Low Energy :

1. Discover the ZDD

   To identify nearby ZDDs, the ZVD SHALL perform discovery of ZDDs as described in section 8.3.1.

2. Connect to the ZDD

   The ZVD SHALL setup a Bluetooth Low Energy connection with a selected ZDD.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

3. Establish secure session

   Once connected to a ZDD, the ZVD SHALL setup a Zigbee Direct secure session, as described in section 6. Specific security behavior for the ZVD is described in section 8.3.2.

4. Access required characteristics

   The ZVD accesses the required characteristics for the services listed in section 7.7.2 and/or section 7.7.3 - refer to the individual paragraphs below. The ZVD MAY access additional, manufacturer defined, services and characteristics.

5. Keep connection active when using the Zigbee Direct Tunnel Service

   After a connection to a ZDD is established, and the ZVD performs a role on the Zigbee network, the ZVD SHALL do a best effort to maintain connection to Zigbee Direct, refer to section 8.3.3.

   The above steps MAY be repeated for connections to multiple ZDDs in parallel, where any new Zigbee Direct connection SHALL NOT cause existing Zigbee Direct connections to drop. A ZVD MAY be capable of accessing multiple Zigbee networks through their respective Zigbee Direct services, one at a time, for this purpose the ZVD would store the relevant information on the Zigbee network, such as PAN ID and security material.

8.3.1. Discovery of ZDD

The ZVD SHALL be able to discover nearby ZDD devices based on the advertisement sent out by the ZDD by scanning for these advertisements using the primitives defined in [R3] Vol 2 Part E.

The ZVD SHALL be able to filter advertisements based on the primary service UUID defined in section 7.6.2. The ZVD MAY reconnect to a ZDD using a previously obtained BLE address.

This procedure MAY be performed upon start of the ZVD and whenever the ZVD needs to reconnect to the ZDD.

8.3.2. Security

The ZVD SHALL communicate to a ZDD over a secure channel established using the service described in section 6. The ZVD does not rely on Bluetooth Low Energy specific security for the Zigbee Direct defined services, and as such does not make use of any of the pairing methods defined in [R3] Vol 3, Part H for Zigbee Direct.

When establishing the secure channel, the ZVD SHALL use the Zigbee Direct Security Service to exchange messages with the ZDD in a specific order, described in section 6.5.1.1.

The ZVD SHALL support a mechanism to receive security material for authorization for the Zigbee Direct services. There are multiple means by which security material can be provisioned to the ZVD:

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

- The provisioning MAY be done via a secure out-of-band mechanism, such as via a cloud connection.
- If the ZVD creates the network, ie. it is commissioning the Zigbee network via the Commissioning Service, it will create and subsequently store the required security material.
- The ZVD SHALL also support in-band provisioning, as described in section 6 that allows the ZVD to receive the Zigbee Direct security material directly from the ZDD for centralized security networks.

The ZVD MAY use manufacturer defined security mechanisms, overriding the Zigbee Direct defined mechanisms at runtime. Definition of these additional security mechanisms is out-of-scope for this document. It is strongly recommended that any manufacturer defined security mechanism be at least as secure as the Zigbee Direct.

8.3.3. Connection re-establishment mechanism

This method is required for a ZVD to remain connected to a ZDD providing access to a Zigbee network. An example of when such a mechanism is used is when a ZVD on a phone is brought out of range of a ZDD and back into range of the same ZDD or a different ZDD providing access to the same Zigbee network.

This mechanism applies to a ZVD that uses a connected ZDD's Zigbee Direct Tunnel Service for performing a role on its Zigbee network. If for some reason the ZVD loses connection to the ZDD, or the secure session to the ZDD has been lost, the ZVD SHALL do a best effort to restore the connection to the selected Zigbee network by performing these steps:

1. Try to (re-)connect to the original ZDD or another ZDD on the same network.

   This can for example be accomplished by letting the ZVD scan for advertisements with the correct 16-bit PAN identifier as described in section 8.3.1 or by reconnecting to the ZDD using the BLE address(es) of previously known ZDD(s).

   The amount of times this process is repeated is application controlled, but the ZVD SHALL retry the discovery process at least once.

2. Once connected, re-establish a secure session with the ZDD as described in section 8.3.2, based on the existing key material for the Zigbee network. If session establishment fails, the ZVD SHALL discover an alternative ZDD starting from step 1.
3. A ZVD that supports the Zigbee Direct Tunnel Service SHALL rejoin the network by tunneling a Network Commissioning Request, and then awaiting and processing the corresponding Network Commissioning Response.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

8.4. Zigbee Direct BLE Service clients

8.4.1. Introduction

This paragraph describes the BLE clients for Zigbee Direct services that a ZVD supports. The ZVD SHALL implement the Zigbee Direct Security client as well as the Zigbee Direct Commissioning client, for which the Services are described in section 6 and section 7.7.2 respectively. The ZVD MAY implement the Zigbee Direct Tunnel client as described in section 7.7.3.

8.4.2. Commissioning Client

8.4.2.1. Introduction

The commissioning client allows a ZVD to commission devices on a Zigbee network by using Zigbee Direct Commissioning Service (section 7.7.2).

8.4.2.2. Security

The Zigbee Direct Commissioning Client SHALL use security as described in section 6.

8.4.2.3. Client Functional Description

The ZVD MAY implement a Zigbee Direct Commissioning Client. After secure connection to a ZDD that provides the Zigbee Direct Commissioning Service, the ZVD SHOULD follow these steps:

1. Subscribe to notifications on the Network Status characteristic (see section 7.7.2.9).
2. Perform a commissioning action on the connected ZDD by writing to one of its Zigbee Direct Commissioning Service characteristics.
3. Process any incoming notification as received on the Network Status characteristic
4. Repeat the process from step 2 for additional characteristics
5. (Optionally) Disconnect once the ZDD has been commissioned. The connection MAY be kept if, for example, ZVD intends to become part of the commissioned network using the tunnel client as described in section 8.4.3.

This functionality SHALL be implemented either as a singular block of software implementing the ZVD's functionality (type 1 component), or by including a Comissioning Component as outlined in section C.1 (type 2 component, see [R11]).

8.4.3. Tunnel Client

8.4.3.1. Introduction

The tunnel client allows a ZVD to tunnel data to and from a Zigbee network using the Zigbee Direct Tunnel Service on that ZDD, refer to section 7.7.3.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

The ZVD MAY implement a Zigbee Direct Tunnel Client, thus implementing a potentially purpose built Zigbee stack including BDB and ZCL functionality, e.g. by the use of software components.

8.4.3.2. Security

The Zigbee Direct Tunnel Client SHALL use security as described in section 6. Zigbee NWK layer security SHALL NOT be used. Zigbee APS layer security MAY be used.

8.4.3.3. Client Functional Description

A ZVD with support for the tunnel client (ZVD-TS) SHALL implement:

- interfaces to lower layer (ZVD OS and hardware) and higher layer (Zigbee Direct client interface)
- Zigbee Stack (APS, NWK, ZDO)
- BDB (where applicable)
- ZCL

…either as a singular block of software implementing the ZVD's functionality (type 1 component), or by referencing software components implementing the required functionality (type 2 component) (see [R11]).

After establishing secure connection to a ZDD that provides the Zigbee Direct Tunnel Service, the ZVD SHALL follow these steps:

1. Send a Network Commissioning Request, acquire the ZDD IEEE address. A ZVD SHALL acquire a network short address using the procedure specified in [R12] 3.6.1.8
2. Await and process the Network Commissioning Response.
3. Based on the Zigbee Role, execute actions following the component description.
4. Maintain the Zigbee Direct connection as described in section 8.3.3

Upon losing connection to the ZDD, follow the steps to reacquire connection, then perform the steps starting at step 1.

8.5. Extended Unique Identifier Allocation

A ZVD that supports the Tunnel Client requires a 64-bit IEEE Extended Unique Identifier (EUI-64) to operate on a Zigbee network.

ZVDs that have access to a globally unique EUI-64 provided by the underlying hardware platform and know that the platform does not support other ZVD instances MAY use this EUI-64.

Conversely, ZVDs that run on platforms that support multiple instances of ZVDs require dynamically generated EUI-64s. These MAY be randomly generated; randomly generated EUI-64 SHALL have the U/L bit set to denote a locally administered EUI-64 [R7].

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

9. Network Key Rotation

In centralized security networks, the trust center of the network may decide to rotate the network key on a regular basis, e.g. once per year or once per month; trust centers compliant to the Zigbee Specification 2015 (R21) or later will also rotate network keys when a device in their vicinity exceeds 50% of its outgoing security frame counter range, i.e. 2^31 (0x80000000). In such cases the Trust Center will send a Transport Key message, either without APS encryption, when sent as a broadcast, or encrypted with device-individual Trust Center link keys. In order to prevent leakage of the new network key to the ZVD, a ZDD SHALL perform filtering on these messages as follows:

- The NWK layer of ZDD that intends to forward a network layer data message to a ZVD based on standard routing decisions SHALL hold off forwarding and first ask its APSME for permission to forward this frame.
- The APSME SHALL make this decision by inspecting the application layer contents of this frame in order to determine whether it is a Transport Key message conveying an active or prospective network key. It SHALL perform this determination as follows:
  1. If the APS command frame is not using APS layer security, the APSME of the ZDD SHALL examine the APSME command identifier, and in case of a Transport Key message, the StandardKeyType field of that message. If the StandardKeyType equals 'Standard network key' (0x01) the message is considered a Transport Key message conveying an active or prospective network key and the APSME SHALL decline forwarding of this frame to the ZVD.
  2. If the APS command frame is using APS layer encryption, the APSME of the ZDD SHALL examine the auxiliary security header, and in particular the Key Identifier sub-field of the Security Control field. If the Key Identifier equals 'key-transport key' (0x02) the message SHALL be considered a Transport Key message conveying an active or prospective network key and the APSME SHALL decline forwarding of this frame to the ZVD.
- If the APSME declined forwarding of this frame to the ZVD the NWK layer SHALL NOT forward this frame over a Bluetooth Low Energy connection; otherwise the NWK SHALL proceed with forwarding the frame as usual.
- If the APSME declined forwarding of this frame to the ZVD due to it having detected a Transport Key message conveying an active or prospective network key as specified above, the ZDD SHALL close the Bluetooth connection to the ZVD in order to enforce a session re-establishment.

In addition to the filtering above, the APSME of a ZDD operating on a legacy network SHALL perform additional inspection and filtering as specified in section 10.1.

9.1. Limited Authorization Session

Whenever a ZVD establishes a secure session with a ZDD, it SHALL perform a network key sequence number check in order to detect a potential network key rotation that occured while

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

the ZVD was off the network. It SHALL compare the network key sequence TLV (The ZVD using Basic Key to establish the authorization session SHALL verify that the NWK key sequence number (section 6.5.2.4) received in Session Establishment Message 2 (section 6.5.1.2.3), if one is included, to the nwkActiveSeqNumber of the Network Information Base (NIB) maintained by the Network Layer Management Entity (NLME). If the values differ, the ZVD has recognized a network key rotation. This is also detailed in section 7.7.4.6.

On any type of centralized security network (ZD-aware or legacy), when a ZVD recognizes that the network key has changed since it was last connected to the network, and, as a consequence, its Basic Authorization key has become void, it SHALL initiate a trust center rejoin. In order to do so, the ZVD SHALL establish a Limited Authorization session. A Limited Authorization session is established by the ZVD selecting a Preshared Key (PSK) in Session Establishment Message 1 for Basic Authorization and including a Key Sequence Number TLV with the network key sequence number of the key which the ZVD currently has on record. For the ZDD, this will be a Basic Authorization key that was created based on a past network key.

In order for a ZDD to be able to recognize past basic authorization keys, a ZDD operating on any type of centralized security network (ZD-aware or legacy) SHALL store the current network key prior to switching to a new network key along with the network key sequence number, whenever a network key update occurs. A ZDD SHALL store at least one such key in non-volatile memory and MAY store more networking keys if it has sufficient persistent storage space.

Such past basic authorization key SHALL only be used by the ZDD to allow for TC rejoin. The ZDD SHALL NOT allow any other action on behalf of any ZVD authenticated with such past basic authorization key. The same general authorization rules apply, which apply to a simple Provisioning Session, with the exception that Tunnel Data commands carrying an active or prospective network key will be translated to Transport Key commands conveying basic authorization keys as detailed in section 10.

A ZDD SHALL NOT generate a basic authorization key from a transport key message conveying an active or prospective network key outside a Limited Authorization session.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

10. Support for Legacy Networks

Networks that have been deployed prior to the release and adoption of Zigbee Direct, or are formed by devices, in particular Trust Center devices, which do not actively support Zigbee Direct, may nevertheless take advantage of the Zigbee Direct feature. Such support may be introduced by adding at least one ZDD to an existing network or by upgrading at least one device to a newer firmware version that includes ZDD functionality. This kind of support for legacy networks is available for three classes of networks:

1. Class I: Distributed security networks,
2. Class II: Centralized security networks, which do not utilize install codes for joining devices, and
3. Class III: Centralized security networks requiring install codes.

In order to support these classes of pre-existing networks, the ZDD SHALL implement alternate APSME Tunnel command handling as follows:

1. The APSME of a ZDD that receives a Tunnel command SHALL determine if such command is destined to a standard IEEE 802.15.4 Zigbee device or a ZVD by matching the Destination address field of the Tunnel command against the entries in its neighbor table.
    1. If the frame is destined to a ZVD, a ZDD operating on such a legacy/pre-existing network (as determined by the procedure described in section 6.2.3) SHALL NOT perform standard Tunnel command handling, i.e. under any and all circumstances it SHALL NOT forward the contained ASDU unmodified to any ZVD.
    2. Otherwise the ZDD SHALL perform standard Tunnel command handling as specified in the Zigbee Specification.
2. When the ZVD has established a simple provisioning session, i.e. is attempting an initial join, the ZDD is not able to authenticate the ZVD and hence SHALL require proof of authorization by sending the ZVD a transport key message with either the global or unique ephemeral authorization key prior to fully allowing the ZVD onto the network. The decision whether a global or unique ephemeral authorization key is sent is based on the following procedure:
    1. The ZDD SHALL attempt to decrypt the contents of the Tunnel command frame using the well-known global key for centralized security networks, "ZigBeeAlliance09".
        1. If the attempt to decrypt the contents of the Tunnel command frame succeeds, the ZDD SHALL construct an ASDU with a transport key message with the StandardKeyType denoting a global ephemeral authorization key for this ZVD, and the Key Descriptor field being absent. This transport key message SHALL be encrypted with the default trust center link key ("ZigBeeAlliance09") under the 'key-load key' (0x03) derivative. The ZDD SHALL then send this message to the ZVD.
        2. If the attempt to decrypt the contents of the Tunnel command frame fails, the ZDD SHALL construct an ASDU with a transport key message with the StandardKeyType denoting a unique ephemeral authorization key for this ZVD, and the Key Descriptor field being absent. This transport key message SHALL be encrypted with the default

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

trust center link key ("ZigBeeAlliance09") under the 'key-load key' (0x03) derivative. The ZDD SHALL then send this message to the ZVD.

3. When the ZVD has established a Limited Authorization Session according to section 9.1, i.e. it is attempting a Trust Center rejoin, the ZDD SHALL construct an ASDU with a transport key message with the StandardKeyType denoting a basic authorization key for this ZVD, and the Key Descriptor field set to the basic authorization key derived from the active network key and the ZVD's EUI-64 as detailed in section 6.3.2.1. The ZDD SHALL then send this message to the ZVD.

10.1. Ephemeral Authorization

Once a ZDD has sent an ephemeral authorization key to the ZVD in a provisioning session, this session is elevated to an ephemeral authorization session. In an ephemeral authorization session, the ZDD SHALL discard all messages to and from the ZVD, with the following exceptions:

1. NPDU from the ZVD with a destination address in the NHR equal to 0x0000 (Trust Center) and which are APS messages
    1. utilizing APS security, or
    2. not-utilizing APS security, and received after end-to-end authentication is deemed successful (see below)
2. NPDU to the ZVD with a source address in the NHR equal to 0x0000 (Trust Center) and which are APS messages
    1. utilizing APS security, or
    2. not-utilizing APS security, and received after end-to-end authentication is deemed successful (see below)
3. Relay Message Commands:[14]
    1. Relay Message Downstream Command - see [relay-message-downstream]
    2. Relay Message Upstream Command - see [relay-message-upstream]

This will allow the ZVD to exchange messages exclusively with the Trust Center of the network, and such messages being limited to those that utilize APS encryption or are required to establish unique trust center link keys, e.g. the Verify Key and Confirm Key commands, acknowledgment frames to those commands, etc.

The ZDD SHALL deem end-to-end authentication between the Trust Center and the ZVD to have succeeded, when it observes an APS-encrypted message from the Trust Center to the ZVD with the Key Identifier sub-field of the Security Control field of the auxiliary security header denoting a 'key-load key' (0x03). If after a period determined by apsSecurityTimeoutPeriod from establishment of the Ephemeral Authorization session, no proof of end-to-end authentication has been observed, the ZDD SHALL disconnect from the ZVD and consequently remove the provisional entry for the ZVD from its neighbor table. Conversely, as soon as end-to-end authentication is deemed successful, the ZDD SHALL allow the exchange of any NPDU between the ZVD and, initially, the Trust Center. It SHALL then await another APS-encrypted message

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

from the Trust Center to the ZVD, this time with the Key Identifier sub-field of the Security Control field of the auxiliary security header denoting a 'data key' (0x00). Only then the authentication sequence has fully completed and the ZDD SHALL send a transport key message with the StandardKeyType denoting a basic authorization key and the key field containing the basic authorization key derived for the ZVD's EUI-64. This transport key message SHALL be encrypted with the default trust center link key ("ZigBeeAlliance09") under the 'key-load key' (0x03) derivative. ~~The source address of this Transport Key Message SHALL be that of the Trust Center (0x0000)~~[15][16]. The ZDD SHALL then update the relationship field of the corresponding neighbor table entry from 'unauthenticated child' to either 'child' or 'sibling' depending on ZVD capabilities (router vs. end-device); it SHALL also elevate the authorization level of the session from ephemeral to basic authorization thereby allowing the traversal of any NPDU between the ZVD and the Zigbee network.

A ZVD that receives a transport key message with an ephemeral authorization key (global or unique), SHALL update its link key with the Trust Center before operating on the network. This is a precondition for receiving the basic authorization key. The ZVD MAY perform this update either by instigating the Zigbee 3.0 Trust Center Link Key Exchange (Request Key method), or by application-defined means, for example by instigating a Certificate Based Key Exchange (CBKE). In order to so, it SHALL initially use the global trust center link key if the global ephemeral authorization key has been delivered, or it shall use its unique Trust Center link-key if the unique ephemeral authorization key has been delivered. Only after receiving the basic authorization key from the ZDD, the ZVD SHALL start normal network operation, typically broadcasting a ZDO Device Announcement as one of its first messages. During ephemeral authorization, encrypted APS ACKs SHALL be exchanged between ZVD and the Trust Center[17].

10.1.1. Routing on behalf of the ZVD

While a ZVD is operating in an ephemeral authorization session and, hence, not fully authenticated on the network, it cannot participate in routing. Any traffic with the 'assume network security' flag set to '1', with the few exceptions mentioned above, will be blocked by the ZDD, which acts as security enforcement point. Specifically, neither route requests, nor route replies, nor route record frames would be allowed to pass the trusted link.

While the joining ZVD can safely assume that its parent router ZDD has a route to the trust center of the network and leverage this information to send the Request Key message, the reverse path has not been established yet. The trust center needs a route towards the joining ZVD in order to send APS acknowledgment frames and transport a new unique link-key[18].

The ZDD SHALL participate in routing on behalf of a ZVD that is attached in the ephemeral authorization state in order to ensure that the trust center link-key exchange works across multiple hops. Solely for the purposes of routing, a ZDD SHALL treat each ZVD that is attached in an ephemeral authorization session as if it was an end-device child and follow the procedures defined in Zigbee Specification [R12] sections

- *"3.6.4.5.1.2 Routing and Route Discovery Table Maintenance, Route Request Forwarding"*, and
- *"3.6.4.5.5 Initiation and Processing of a Route Record Command Frame"*

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

or the respective sections in [R12].

Once a routing-capable ZVD is fully authenticated on the network all routing related behavior is excercised by the ZVD alone - as is normal behavior for any native IEEE 802.15.4 Zigbee router. It will receive or initiate route requests and respond with route replies as necessary. The ZDD SHALL NOT perform routing on behalf of the ZVD as specified above in normal operating state any more.

10.2. Joining flows for various network classes

10.2.1. Class I: Distributed Security Networks

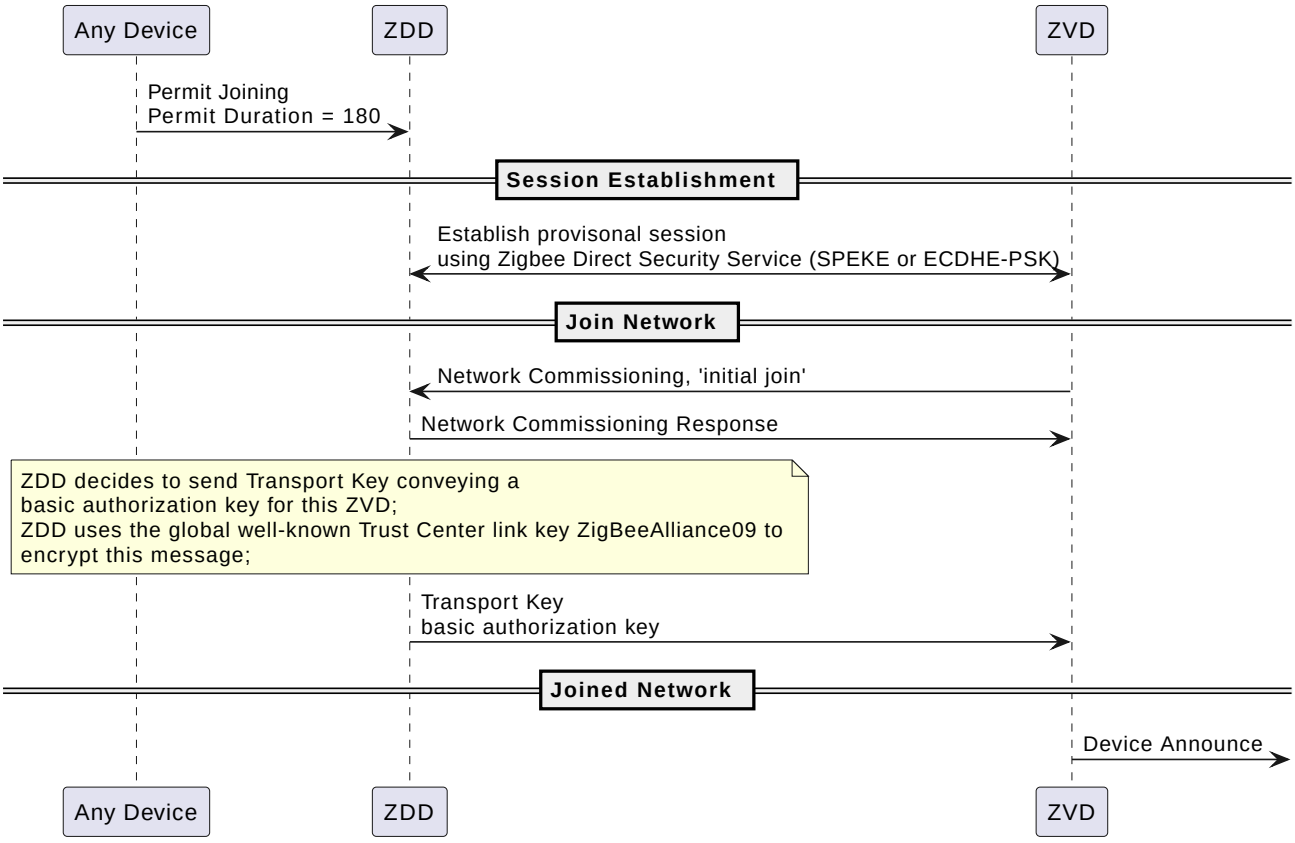


Figure 19. Legacy distributed security networks

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

10.2.2. Class II: Networks not utilizing install codes

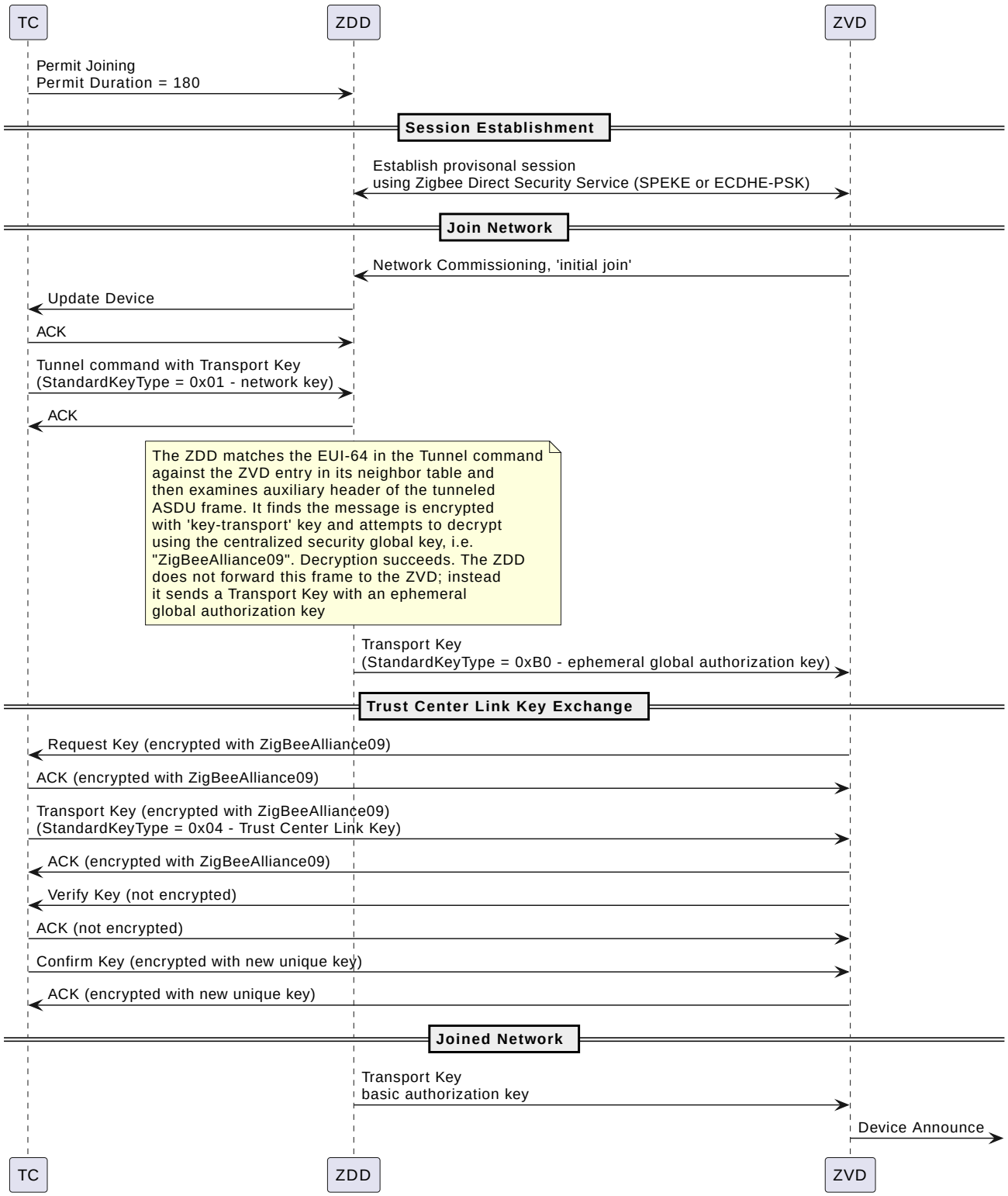


Figure 20. Legacy networks not utilizing install codes

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

10.2.3. Class III: Networks utilizing install codes

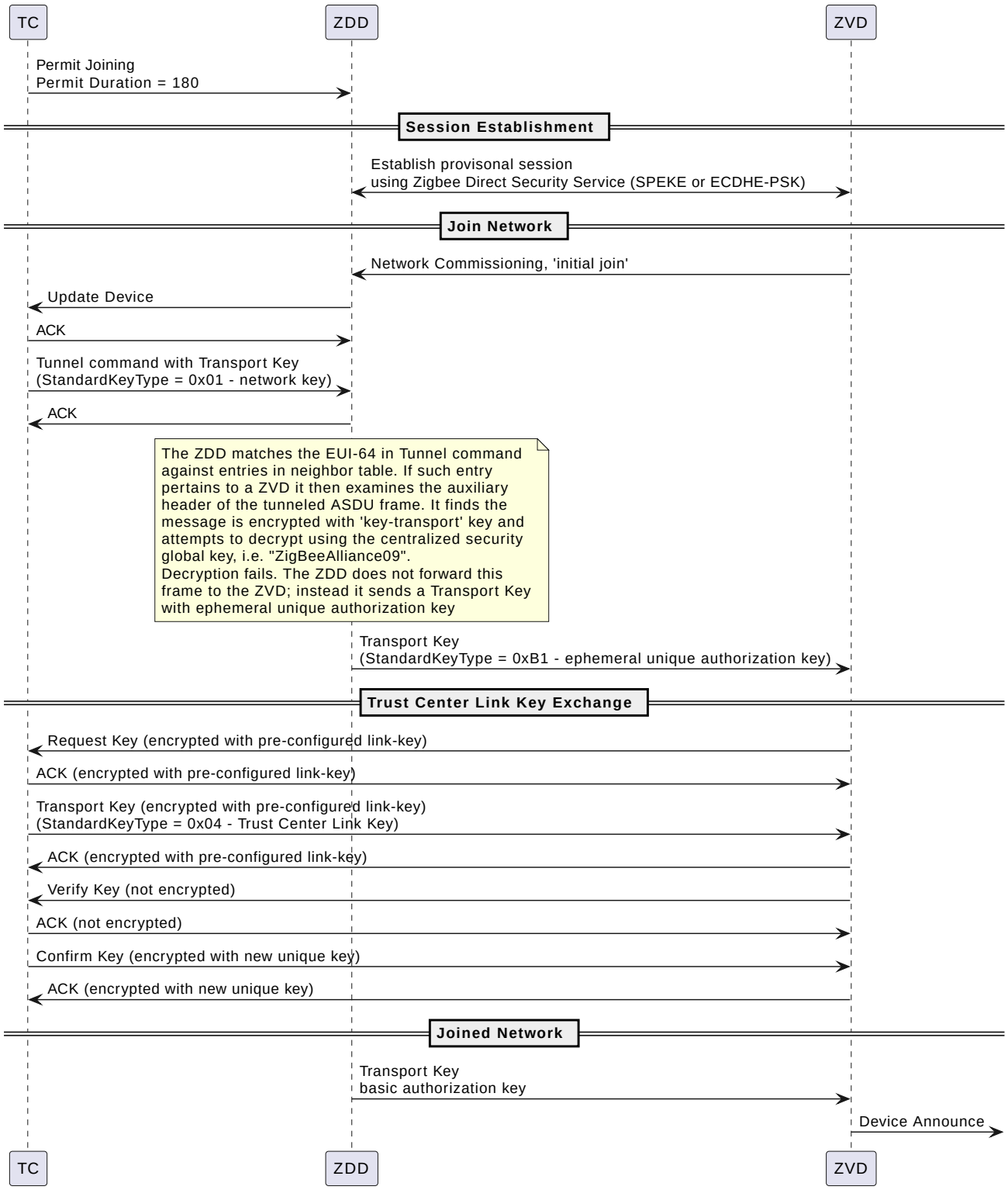


Figure 21. Legacy networks utilizing install codes

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

11. Additions to Zigbee Cluster Library Specification

11.1. Introduction

This annex describes proposed additions to the Zigbee Cluster Library Specification - 075123 [R8].

11.2. Additions to Chapter 3 - General

11.3. Zigbee Direct Configuration Cluster

11.3.1. Overview

This cluster provides an interface for configuring Zigbee Direct. The commands received by this cluster SHALL only be unicast.

11.3.2. Revision History

The global *ClusterRevision* attribute value SHALL be the highest revision number in the table below.

Rev	Description
1	initial revision

11.3.3. Classification

Hierarchy	Role	PICS Code	Primary Transaction
1	initial revision	<ZBDRCT>	Type 1 (client to server)

11.3.4. Cluster Identifiers

Identifier	Name
0x003d	Zigbee Direct Configuration

11.3.5. Server

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

11.3.5.1. Security

In a centralized security network, all interactions with the Zigbee Direct Configuration cluster, including read and discovery commands, SHALL require APS encryption with the Trust Center link-key. In a Centralized Security Network, the authorized source SHALL be the Trust Center. In a distributed security network, any device on the network is authorized to interact with this cluster without APS security.

11.3.5.2. Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 56.

Table 56. Zigbee Direct Attribute Sets

Attribute Set Identifier	Description
0x000	Zigbee Direct Information

11.3.5.3. Zigbee Direct Information Attribute set

Table 57. Zigbee Direct Information Attribute set

Identifier	Name	Type	Range	Access	Default	M/O
0x0000	*InterfaceState*	map8	0x00 to 0xFF	R	-	M
0x0001	*AnonymousJoinTimeout*	uint24	0x000000 to 0x100000	R	-	M

11.3.5.3.1. InterfaceState

The *InterfaceState* attribute is a bitmap.

Bit 0 specifies the current state of the Zigbee Direct interface as follows: 1 = enabled, 0 = disabled.

All other bits are reserved.

11.3.5.3.2. AnonymousJoinTimeout

The *AnonymousJoinTimeout* attribute SHALL represent the timeout in seconds after a power-cycle or local manufacturer specific stimulus that a ZVD can establish a secure provisioning session with the provisioned ZDD using the Anonymous Well-Known Secret "ZigbeeAlliance18" when the Zigbee Network is open to accept new devices, i.e. when the network's constituent nodes respond with Beacons with the Association Permit bit set when requested.

When the *AnonymousJoinTimeout* is set to 0, the Anonymous Well-Known Secret SHALL not be

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

accepted by the provisioned ZDD when establishing a new provisioning session with a ZVD even if the Zigbee Network is open to accept new devices.

When the AnonymousJoinTimeout is set to the INVALID_VALUE (0xFFFFFF), the Anonymous Well-Known Secret SHALL be accepted by the provisioned ZDD when establishing a new provisioning session with a ZVD while the Zigbee Network is open to accept new devices. [19]

The recommended default value for the *AnonymousJoinTimeout* is 3600 seconds.

11.3.5.4. Commands

11.3.5.4.1. Commands Received

Table 58. Zigbee Direct Configuration Cluster Received Command IDs

Command Identifier	Description	M/O
0x00	Configure Zigbee Direct Interface	M
0x01	Configure Zigbee Direct Anonymous Join Timeout	M

11.3.5.4.2. Generic Usage Notes

On receipt of the *Configure Zigbee Direct Interface* command or *Configure Zigbee Direct Anonymous Join Timeout* command via the groupcast or broadcast transmission service, the client SHALL ignore the command and no processing SHALL be done.

11.3.5.4.3. Configure Zigbee Direct Interface Command

This command enables or disables the Zigbee Direct Interface on the device. These configuration changes SHALL be persistent, i.e. the changes must survive a reboot.

When the Zigbee Direct Interface is disabled using this command, the following SHALL happen:

- any *new* requests to the services defined for Zigbee Direct ([Zigbee Direct Specification section 7.7]) SHALL be ignored
- *existing* requests to the services defined for Zigbee Direct ([Zigbee Direct Specification section 7.7]) SHALL be dropped
- the advertisement of Zigbee Direct SHALL be disabled. If there are manufacturer specific extensions advertised, advertisements MAY still be sent, but the Zigbee Direct specific advertisement data ([Zigbee Direct Specification section 7.6.3]) SHALL be removed from the advertisement and the advertisement frequency SHOULD be set to the lowest possible value for the device.

When the Zigbee Direct Interface is enabled using this command, the following SHALL happen:

- new requests to the services defined for Zigbee Direct ([Zigbee Direct Specification section

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

7.7]) SHALL be processed

- the advertisement of Zigbee Direct SHALL be enabled. If there are manufacturer specific extensions advertised, Zigbee Direct specific advertisement data (according to [Zigbee Direct Specification section 7.6.3]) SHALL be included in the advertisement and the advertisement frequency SHALL be set to the default value for the device.

Payload Format

Table 59. Format of the Configure Zigbee Direct Interface Command

Octets	1
Data Type	map8
Field Name	Interface State

InterfaceState Field

The Interface State field is 8-bits in length and specifies the state of the Zigbee Direct interface on the device to be configured. This field SHALL contain one of the non-reserved values listed in Table 60.

Table 60. Values of the InterfaceState Field of the Configure Zigbee Direct Interface Command

Interface State Field Value	Effect	Notes
bit 0 = 0	Disabled	Disables the Zigbee Direct interface on the device
bit 0 = 1	Enabled	Enables the Zigbee Direct interface on the device
bits 1-7	Reserved	Reserved for future use

Effect on Receipt

On receipt of this command, the device SHALL perform the following procedure:

1. The device SHALL verify that the source is authorized according to section 11.3.5.1. If not, the device SHALL send back a Default Response with a status of NOT_AUTHORIZED.
2. The device SHALL ignore values in the Interface State that are marked reserved in Table 60.
3. The device compares the current interface state with that of the requested Interface State by the client. If the interface state is the same, the device SHALL set the status to SUCCESS and continues from step 5.
4. The device changes the state of the interface to the requested state. If the interface state change is unsuccessful, the device SHALL set the status to FAILURE. If the interface state change is successful, the device SHALL set the status to SUCCESS.
5. The device SHALL generate a Configure Zigbee Direct Interface Response Command section 11.3.5.5.1 with the Status Field set to the evaluated status and the Interface State to the

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

current state of the Zigbee Direct Interface on the device.

11.3.5.4.4. Configure Zigbee Direct Anonymous Join Timeout Command

This command configures the timeout in seconds after a power-cycle or local manufacturer specific stimulus that a ZVD can establish a secure provisioning session with a provisioned ZDD using the Anonymous Well-Known Secret "ZigbeeAlliance18" when the Zigbee Network is open to accept new devices, i.e. when the network's constituent nodes respond with Beacons with the Association Permit bit set when requested, per [Zigbee Direct Specification section 7.7.4.3, section 7.7.4.5]. These configuration changes SHALL be persistent, i.e. the changes must survive a reboot.

When the Anonymous Join Timeout is configured to 0, the Anonymous Well-Known Secret SHALL not be accepted by the provisioned ZDD for establishing a new provisioning sessions with a ZVD when the Zigbee Network is open to accept new devices.

When the Zigbee Direct Anonymous Join Timeout is set to a non-zero value, the following SHALL happen:

- after a power-cycle or local manufacturer specific stimulus of the provisioned ZDD, or when the Anonymous Join Timeout value is set using this command, an Anonymous Join Countdown Timer SHALL start with the value set to Anonymous Join Timeout seconds
- when a ZVD establishes a secure provisioning session using the Anonymous Well-Known Secret "ZigbeeAlliance18", the provisioned ZDD SHALL verify whether the network is open to accept new devices, i.e. when the network's constituent nodes respond with Beacons with the Association Permit bit set when requested, per [Zigbee Direct Specification section 7.7.4.3, section 7.7.4.5]
- when the network is open to accept new devices, the provisioned ZDD SHALL verify whether the Anonymous Join Countdown Timer is set to 0, when this is the case the connection to the ZVD SHALL be closed and further processing SHALL be stopped.
- when the Anonymous Join Countdown Timer is set to a value higher than 0, the provisioned ZDD SHALL continue establishing a provisioning session with the ZVD per [Zigbee Direct Specification section 7.7.4.3, section 7.7.4.5]

Payload Format

Table 61. Format of the Configure Zigbee Direct Anonymous Join Timeout Command

Octets	**3**
Data Type	uint24
Field Name	Anonymous Join Timeout

Anonymous Join Timeout Field

The Anonymous Join Timeout field is 24-bits in length and specifies the timeout in seconds after a power-cycle that a ZVD can establish a secure provisioning session with the provisioned ZDD using the Anonymous Well-Known Secret "ZigbeeAlliance18" when the Zigbee Network is open to

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

accept new devices, i.e. when the network's constituent nodes respond with Beacons with the Association Permit bit set when requested, per [Zigbee Direct Specification section 7.7.4.3, section 7.7.4.5].

Effect on Receipt

On receipt of this command, the device SHALL perform the following procedure:

1. The device SHALL verify that the source is authorized according to section 11.3.5.1. If not, the device SHALL send back a Default Response with a status of NOT_AUTHORIZED.
2. The device verifies that the Anonymous Join Timeout field contains a valid 24-bit value within the defined range. If the value is not valid or not in the defined range, the device SHALL generate a Default Response with a status of INVALID_VALUE and stops further processing.
3. The device compares the current Anonymous Join Timeout with that of the requested Anonymous Join Timeout by the client. If the Anonymous Join Timeout is the same, the device SHALL set the status to SUCCESS and continues from step 5.
4. The device changes the Anonymous Join Timeout to the requested value and (re-)start the Anonymous Join Countdown Timer based on the new value. If the Anonymous Join Timeout change is unsuccessful, the device SHALL set the status to FAILURE. If the Anonymous Join Timeout change is successful, the device SHALL set the status to SUCCESS.
5. The device SHALL generate a Default Response with the current status.

11.3.5.5. Commands Generated

Table 62. Zigbee Direct Configuration Cluster Generated Command IDs

Command Identifier	Description	M/O
0x00	Configure Zigbee Direct Interface Response	M

11.3.5.5.1. Configure Zigbee Direct Interface Response Command

This command is sent by a device in response to a Configure Zigbee Direct Interface Command.

Payload Format

Table 63. Payload Format

Octets	1	1
Data Type	enum8	map8
Field Name	Status	Interface State

When Generated

This command is generated in response to a Configure Zigbee Direct Interface Command. The Status field is according to the Effect on Receipt section of the Configure Zigbee Direct Interface

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Command. The Interface State is set the current state of the Bluetooth Low Energy interface on the device.

11.3.6. Client

The client cluster has no specific attributes. The client generates the cluster specific commands detailed in section 11.3.5.4. The client receives the cluster specific response command detailed in section 11.3.5.5.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

This page is intentionally blank

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Annex A: BLE Connectable Advertisement example

This annex provides an example (non-normative) of a BLE Connectable Advertisement that includes the Zigbee Direct extension to the BLE Advertisement as specified in section 7.6.3. A ZDD's BLE connectable advertisement MAY look as follows:

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

AD field	Value / Field Name	Octet #	Description
AD Length	0x02	0	Length of this Advertising Data (AD) field
AD Type	0x01	1	Advertising Data Type set to the Flags data type <<Flags>> as defined in [R5]
AD Data	0x06	2	Flags set: • General Discoverable Mode • BR/EDR Not Supported
AD Length	0x03	3	Length of this Advertising Data (AD) field
AD Type	0x02	4	Advertising Data Type set to the Service UUID data type <<Incomplete List of 16-bit Service UUIDs>> as defined in [R5]
AD Data	0xF7	5	16-bit UUID of the Zigbee Direct Commissioning Service = 0xFFF7 [R6]
	0xFF	6	
AD Length	0x08	7	Length of this Advertising Data (AD) field
AD Type	0x16	8	Advertising Data Type set to the Service Data data type <<Service Data - 16 bit UUID>> as defined in [R5]
AD Data	0xF7	9	16-bit UUID of the Zigbee Direct Commissioning Service = 0xFFF7
	0xFF	10	
	0x11	11	Bits 0..3: Version of this Zigbee Direct extension to the BLE Advertisement = 0x1 Bits 4..7: Device's Zigbee Direct flags = 0x1 (Bit 4 "ZDTS" set to 1: Zigbee Direct Tunnel Service supported, Bit 5 "PermitJoin" set to 0: permit joining status is false)
	0x34	12	The 16-bit PAN identifier of the Zigbee network = 0x1234
	0x12	13	
	0xCD	14	Zigbee NWK address = 0xABCD
	0xAB	15	

NOTE Some of the field values as shown in this table are example values.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Annex B: Security Test Vectors

B.1. Key Agreement using ECDHE-PSK/P-256/SHA256/HMAC-SHA256-128

In the following example, a ZVD with IEEE EUI-64 `00:1f:ee:00:00:00:00:01` establishes a secure session with a ZDD with EUI-64 `00:1f:ee:00:00:00:00:02` using anonymous ECDHE-PSK with curve P-256, using hash algorithm SHA256 and message authentication algorithm HMAC-SHA256.

B.1.1. Round 1: Message 1 from ZVD to ZDD

In round 1, ZVD and ZDD derive a 128-bit session key.

The process starts with the ZVD generating an ephemeral private/public keypair by selecting a random number di:

```
di = 85 5b 95 f0 d4 1f e3 5d 57 64 3e 5c 59 7f 17 5a ac 33 2d 75 95 71 49 ce 89 26
5a 71 90 c3 67 7c
```

And computing the resulting public point Qi:

```
Qi = (37 7f 24 60 3f 7f 49 34 e3 bc 11 3f 58 ad 45 81 50 78 b3 39 a7 87 49 12 dd d9
bf 3d 94 35 8a 1d,
2e 03 06 08 42 81 ed bb 98 44 69 a2 83 64 ed 0c ba 6f 47 83 4d 18 cb 40 80 35 28 d9
75 4b 1c 42)
```

Notice: These are in-memory representations in big-endian format, which is historically the format that is being used to exchange P-256 material. For Curve25519 the opposite is true, i.e. Curve25519 material is exchanged in little-endian format, which is the native format for the majority of CPU architectures used nowadays.

The ZVD sends this information to the ZDD as Session Establishment Message 1:

Session Establishment Message 1

```
01 ①
00 01 03 ff ②
01 47 01 00 00 00 00 ee 1f 00 37 7f 24 60 3f 7f 49 34 e3 bc 11 3f 58 ad 45 81 50 78
b3 39 a7 87 49 12 dd d9 bf 3d 94 35 8a 1d 2e 03 06 08 42 81 ed bb 98 44 69 a2 83 64
ed 0c ba 6f 47 83 4d 18 cb 40 80 35 28 d9 75 4b 1c 42 ③
```

① Message header: Opcode

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

② Key Negotiation Method TLV

③ P-256 Public Point TLV

B.1.2. Round 1: Message 2 from the ZDD to the ZVD

The ZDD generates an ephemeral private/public keypair by selecting a random number dr:

```
dr = fc fc 52 02 c7 20 52 8b ab f4 bf 0d 1f f5 17 bb af e6 a2 4a c5 ad 21 f0 39 d5
88 bd 15 b8 10 a6
```

And computing the resulting public point Qr:

```
Qr = (b1 87 5c 74 87 c1 e2 ca ae e4 14 23 71 b2 cd 42 fd e9 d9 42 3b f3 a1 aa 08 40
7d 2c 01 8a 9e b8,
84 0c db 17 08 4d fd f9 b8 02 5a 9b ba 38 bd 3c f1 48 0f 3b 91 f7 16 89 2c ad a2 b6
77 a8 68 9b)
```

It has all information that is required to compute the shared Diffie-Hellman secret point xk:

```
xk = 1d 35 ce dc 0d 24 d3 25 d8 75 5a b6 0f 01 27 55 eb cd 4b 79 eb bb b9 24 76 78
d1 13 ff 30 31 07
```

And further, it computes the hash value H(xk || I || PSK):

```
s = SHA256(1d 35 ce dc 0d 24 d3 25 d8 75 5a b6 0f 01 27 55 eb cd 4b 79 eb bb b9 24
76 78 d1 13 ff 30 31 07 || 01 00 00 00 00 ee 1f 00 37 7f 24 60 3f 7f 49 34 e3 bc 11
3f 58 ad 45 81 50 78 b3 39 a7 87 49 12 dd d9 bf 3d 94 35 8a 1d 2e 03 06 08 42 81 ed
bb 98 44 69 a2 83 64 ed 0c ba 6f 47 83 4d 18 cb 40 80 35 28 d9 75 4b 1c 42 02 00 00
00 00 ee 1f 00 b1 87 5c 74 87 c1 e2 ca ae e4 14 23 71 b2 cd 42 fd e9 d9 42 3b f3 a1
aa 08 40 7d 2c 01 8a 9e b8 84 0c db 17 08 4d fd f9 b8 02 5a 9b ba 38 bd 3c f1 48 0f
3b 91 f7 16 89 2c ad a2 b6 77 a8 68 9b || 5a 69 67 42 65 65 41 6c 6c 69 61 6e 63 65
31 38)
  = 0d 00 11 6a 64 d3 7c 3e b9 d2 55 c0 0d 2b 1b 90 79 62 6a 6c af c6 bb 2a 9c 49 20
0f 0d 21 6d 36
```

And finally derives the session key using HMAC-SHA256-128(s, { 01 }) as follows:

```
data = 01

key = 0d 00 11 6a 64 d3 7c 3e b9 d2 55 c0 0d 2b 1b 90 79 62 6a 6c af c6 bb 2a 9c 49
20 0f 0d 21 6d 36

HMAC-SHA256 = 87 96 cc 62 87 30 37 d2 52 cf 28 ed df 45 c0 82 26 5f 8f cf 59 b8 73
```


Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

```
23 ec 85 4d 60 93 88 32 b1
```

The resulting session key are the first 128 bits of above result: `87 96 cc 62 87 30 37 d2 52 cf 28 ed df 45 c0 82`.

Then it sends its own public point back to the ZVD:

Session Establishment Message 2

```
02 ①
01 47 02 00 00 00 00 ee 1f 00 b1 87 5c 74 87 c1 e2 ca ae e4 14 23 71 b2 cd 42 fd e9
d9 42 3b f3 a1 aa 08 40 7d 2c 01 8a 9e b8 84 0c db 17 08 4d fd f9 b8 02 5a 9b ba 38
bd 3c f1 48 0f 3b 91 f7 16 89 2c ad a2 b6 77 a8 68 9b ②
```

① Message header: Opcode

② P-256 Public Point TLV

Notice that this specific example is for a factory new ZDD. If the ZDD had already been commissioned to a network the Network Key Sequence Number TLV would have to be included.

Having received the ZDD's public key, the ZVD is now also in a position to compute the shared Diffie-Hellman secret point and arrives at the same session key as the ZDD.

B.1.3. Round 2: Message 3 from ZVD to ZDD

In round 2, the ZVD starts with sending proof to the ZDD that it has arrived at the same session key. It computes the MacTag as follows:

```
data = 4b 43 5f 32 5f 55 02 00 00 00 00 ee 1f 00 b1 87 5c 74 87 c1 e2 ca ae e4 14 23
71 b2 cd 42 fd e9 d9 42 3b f3 a1 aa 08 40 7d 2c 01 8a 9e b8 84 0c db 17 08 4d fd f9
b8 02 5a 9b ba 38 bd 3c f1 48 0f 3b 91 f7 16 89 2c ad a2 b6 77 a8 68 9b 01 00 00 00
00 ee 1f 00 37 7f 24 60 3f 7f 49 34 e3 bc 11 3f 58 ad 45 81 50 78 b3 39 a7 87 49 12
dd d9 bf 3d 94 35 8a 1d 2e 03 06 08 42 81 ed bb 98 44 69 a2 83 64 ed 0c ba 6f 47 83
4d 18 cb 40 80 35 28 d9 75 4b 1c 42

key = 87 96 cc 62 87 30 37 d2 52 cf 28 ed df 45 c0 82

HMAC-SHA256 = cf 34 23 df 94 ca 5f 79 1d ca 21 b6 14 9b 71 84 5b 51 78 be 12 23 2f
21 2e 17 24 7c 25 e2 88 08
```

Session Establishment Message 3

```
03 ①
04 1f cf 34 23 df 94 ca 5f 79 1d ca 21 b6 14 9b 71 84 5b 51 78 be 12 23 2f 21 2e 17
24 7c 25 e2 88 08 ②
```

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

① Message header: Opcode

② MacTag TLV

B.1.4. Round 2: Message 4 from the ZDD to the ZVD

The ZDD, when it receives Message 3, verifies the MacTag received from the ZVD against the expected value. Then, it computes a MacTag to be sent back to the ZVD in Message 4:

```
data = 4b 43 5f 32 5f 56 02 00 00 00 00 ee 1f 00 b1 87 5c 74 87 c1 e2 ca ae e4 14 23
71 b2 cd 42 fd e9 d9 42 3b f3 a1 aa 08 40 7d 2c 01 8a 9e b8 84 0c db 17 08 4d fd f9
b8 02 5a 9b ba 38 bd 3c f1 48 0f 3b 91 f7 16 89 2c ad a2 b6 77 a8 68 9b 01 00 00 00
00 ee 1f 00 37 7f 24 60 3f 7f 49 34 e3 bc 11 3f 58 ad 45 81 50 78 b3 39 a7 87 49 12
dd d9 bf 3d 94 35 8a 1d 2e 03 06 08 42 81 ed bb 98 44 69 a2 83 64 ed 0c ba 6f 47 83
4d 18 cb 40 80 35 28 d9 75 4b 1c 42

key = 87 96 cc 62 87 30 37 d2 52 cf 28 ed df 45 c0 82

HMAC-SHA256 = f1 94 28 82 f2 46 dc 62 eb da 61 85 79 23 04 54 6f 08 f1 ec 98 9b 0c
35 f2 16 99 76 8c cc 97 4a
```

Session Establishment Message 4

```
04 ①
04 1f f1 94 28 82 f2 46 dc 62 eb da 61 85 79 23 04 54 6f 08 f1 ec 98 9b 0c 35 f2 16
99 76 8c cc 97 4a ②
```

① Message header: Opcode

② MacTag TLV

From the ZDD's perspective the session is established at the point where Message 4 is sent.

When the ZVD receives Message 4, it verifies the MacTag received from the ZDD against the expected value. At this point the session is established at both ends and the resulting key can be used for subsequent AES-CCM operations.

B.2. Key Agreement using SPEKE/Curve25519/AES-MMO-128/HMAC-AES-MMO-128

In the following example, a ZVD with IEEE EUI-64 `00:1f:ee:00:00:00:00:01` establishes a secure session with a ZDD with EUI-64 `00:1f:ee:00:00:00:00:02` using anonymous SPEKE with Curve25519, using hash algorithm AES-MMO-128 and message authentication algorithm HMAC-AES-MMO-128.

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

B.2.1. Round 1: Message 1 from ZVD to ZDD

In round 1, ZVD and ZDD derive a 128-bit session key.

The process starts with the ZVD computing the generator base-point G:

```
G = H*(PSK) = 90 2b 44 85 c8 4e c4 a0 59 44 ab 34 42 92 68 78 || 90 2b 44 85 c8 4e
c4 a0 59 44 ab 34 42 92 68 78
```

The ZVD then generates an ephemeral private/public keypair by selecting a random number di:

```
di = a0 f1 d9 2a 82 c8 d8 fe 43 4d 98 55 8c e2 b3 47 17 11 98 54 2f 11 2d 05 58 f5
6b d6 88 07 99 52
```

And computing the resulting public point Qi = diG, where G is the generator base-point as noted above:

```
Qi = 5d 28 29 8c 91 3d 58 8a 35 e0 1f 79 c9 11 66 10 b7 9b 38 e9 c6 cf 15 5c 55 b0
52 9b bc cc 85 56
```

Notice: These are in-memory representations in little-endian format, which is the standard format for exchanging Curve25519 material and the native format for the majority of CPU architectures used nowadays. For X25519 only the x coordinate is relevant and exchanged.

It sends this information to the ZDD as Message 1:

Session Establishment Message 1

```
01 ①
00 01 01 ff ②
02 27 01 00 00 00 00 ee 1f 00 5d 28 29 8c 91 3d 58 8a 35 e0 1f 79 c9 11 66 10 b7 9b
38 e9 c6 cf 15 5c 55 b0 52 9b bc cc 85 56 ③
```

① Message header: Opcode

② Key Negotiation Method TLV

③ Curve25519 Public Point TLV

B.2.2. Round 1: Message 2 from the ZDD to the ZVD

The ZDD also computes the generator base-point G:

```
G = H*(PSK) = 90 2b 44 85 c8 4e c4 a0 59 44 ab 34 42 92 68 78 || 90 2b 44 85 c8 4e
c4 a0 59 44 ab 34 42 92 68 78
```

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

And generates an ephemeral private/public keypair by selecting a random number dr:

```
dr = 48 33 62 41 f3 0d 23 e5 5f 30 d1 c8 ed 61 0c 4b 02 35 39 81 84 b8 14 a2 9c b4
5a 67 2a ca e5 48
```

And computing the resulting public point Qr = drG, where G is the generator base-point as noted above:

```
Qr = 82 77 2f 75 3d e9 4e b4 92 0f b4 17 d4 67 17 80 81 74 0c 1a 87 58 fd 3f 2f 63
02 b5 0b 98 35 6f
```

It has all information that is required to compute the shared Diffie-Hellman secret point xk:

```
xk = 2c 2e 83 1b 07 f6 09 20 05 70 27 e7 89 52 f1 6a 54 41 d6 6d 69 4a dd 3b a0 8a
2b f5 1a e2 34 28
```

And further, it computes the hash value H(xk || I || G):

```
s = AES-MMO-128(2c 2e 83 1b 07 f6 09 20 05 70 27 e7 89 52 f1 6a 54 41 d6 6d 69 4a dd
3b a0 8a 2b f5 1a e2 34 28 || 01 00 00 00 00 ee 1f 00 5d 28 29 8c 91 3d 58 8a 35 e0
1f 79 c9 11 66 10 b7 9b 38 e9 c6 cf 15 5c 55 b0 52 9b bc cc 85 56 02 00 00 00 00 ee
1f 00 82 77 2f 75 3d e9 4e b4 92 0f b4 17 d4 67 17 80 81 74 0c 1a 87 58 fd 3f 2f 63
02 b5 0b 98 35 6f || 90 2b 44 85 c8 4e c4 a0 59 44 ab 34 42 92 68 78 90 2b 44 85 c8
4e c4 a0 59 44 ab 34 42 92 68 78)
  = ac b2 8c 1c 8f f4 2b 0e bc f8 47 bd 05 e8 cd c0
```

And finally derives the session key using HMAC-AES-MMO-128(s, { 01 }):

```
data = 01

key = ac b2 8c 1c 8f f4 2b 0e bc f8 47 bd 05 e8 cd c0

session key = af f1 93 4b e6 9b 40 11 17 9d 81 d9 f9 a5 ff eb
```

Then it sends its own public point back to the ZVD:

Session Establishment Message 2

```
02 ①
02 27 02 00 00 00 00 ee 1f 00 82 77 2f 75 3d e9 4e b4 92 0f b4 17 d4 67 17 80 81 74
0c 1a 87 58 fd 3f 2f 63 02 b5 0b 98 35 6f ②
```

① Message header: Opcode

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

② Curve25519 Public Point TLV

Notice that this specific example is for a factory new ZDD. If the ZDD had already been commissioned to a network the Network Key Sequence Number TLV would have to be included.

Having received the ZDD's public key, the ZVD is now also in a position to compute the shared Diffie-Hellman secret point and arrives at the same session key as the ZDD.

B.2.3. Round 2: Message 3 from ZVD to ZDD

In round 2, the ZVD starts with sending proof to the ZDD that it has arrived at the same session key. It computes the MacTag as follows:

```
data = 4b 43 5f 32 5f 55 02 00 00 00 00 ee 1f 00 82 77 2f 75 3d e9 4e b4 92 0f b4 17
d4 67 17 80 81 74 0c 1a 87 58 fd 3f 2f 63 02 b5 0b 98 35 6f 01 00 00 00 00 ee 1f 00
5d 28 29 8c 91 3d 58 8a 35 e0 1f 79 c9 11 66 10 b7 9b 38 e9 c6 cf 15 5c 55 b0 52 9b
bc cc 85 56

key = af f1 93 4b e6 9b 40 11 17 9d 81 d9 f9 a5 ff eb

HMAC-AES-MMO-128 = da 21 b1 04 d0 73 96 6f 5a 38 80 27 2f 8a f2 f0
```

Session Establishment Message 3

```
03 ①
04 0f da 21 b1 04 d0 73 96 6f 5a 38 80 27 2f 8a f2 f0 ②
```

① Message header: Opcode

② MacTag TLV

B.2.4. Round 2: Message 4 from the ZDD to the ZVD

The ZDD, when it receives Message 3, verifies the MacTag received from the ZVD against the expected value. Then, it computes a MacTag to be sent back to the ZVD in Message 4:

```
data = 4b 43 5f 32 5f 56 02 00 00 00 00 ee 1f 00 82 77 2f 75 3d e9 4e b4 92 0f b4 17
d4 67 17 80 81 74 0c 1a 87 58 fd 3f 2f 63 02 b5 0b 98 35 6f 01 00 00 00 00 ee 1f 00
5d 28 29 8c 91 3d 58 8a 35 e0 1f 79 c9 11 66 10 b7 9b 38 e9 c6 cf 15 5c 55 b0 52 9b
bc cc 85 56

key = af f1 93 4b e6 9b 40 11 17 9d 81 d9 f9 a5 ff eb

HMAC-AES-MMO-128 = c2 3b 54 c1 3d 8a 8a 14 18 a4 ff a6 9d 2b bb 0d
```

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Session Establishment Message 4

```
04 ①
04 0f c2 3b 54 c1 3d 8a 8a 14 18 a4 ff a6 9d 2b bb 0d ②
```

① Message header: Opcode

② MacTag TLV

From the ZDD's perspective the session is established at the point where Message 4 is sent.

When the ZVD receives Message 4, it verifies the MacTag received from the ZDD against the expected value. At this point the session is established at both ends and the resulting key can be used for subsequent AES-CCM operations.

B.3. Deriving authorization keys on a ZDD or TC

Assume the ZDD or the Trust Center of the network needs to derive either a basic or an administrative authorization key for ZVD `00:1f:ee:00:00:00:00:01`. Further assume the network key being `c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf` and the trust center link key for the ZDD being `d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df`.

B.3.1. Basic Authorization Key

```
s = H(ZVD EUI-64 || network key)
  = AES-MMO-128(01 00 00 00 00 ee 1f 00 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce
cf)
  = 80 c3 73 e6 6e 99 15 fc c3 bf 66 f9 44 e4 15 09

HMAC-AES-MMO-128(s, { 03 }) = 5e e6 7a dc 0b a6 eb d8 af 0c b0 43 f2 78 3c e9
```

The resulting basic authorization key is `5e e6 7a dc 0b a6 eb d8 af 0c b0 43 f2 78 3c e9`.

B.3.2. Administrative Authorization Key

```
s = H(ZVD EUI-64 || link key)
  = AES-MMO-128(01 00 00 00 00 ee 1f 00 d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de
df)
  = fd 60 b2 b6 84 92 89 fd 03 79 39 af a5 6c be 2b

HMAC-AES-MMO-128(s, { 04 }) = a8 3d 8a 58 89 c3 03 87 75 a0 c3 48 05 59 5c 0a
```

The resulting administrative authorization key is `a8 3d 8a 58 89 c3 03 87 75 a0 c3 48 05 59 5c 0a`.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

B.4. Accessing secure characteristics via AES-CCM-128

Assume the ZVD 00:1f:ee:00:00:00:00:01 has derived the session key a2 e2 b6 1b f1 a8 05 81 d0 7f 1f 06 72 d1 d6 24.

It then wishes to write the payload

```
07 00 03 00 07 d9 1f 00 00 00 ae 1f 00 01 01 26 90 02 03 00 00 01 00 03 0f 5c d2 f8
be d4 ea 50 56 ac 02 a8 ee 84 1a a0 86 06 01 cd ab
```

to the Join Network characteristic (7072377D-0002-421C-B163-491C27333A61) of the Commissioning Service (16-bit service UUID FFF7). Further assuming it is the second authenticated message being sent from the device, i.e. outgoing frame counter = 2.

To do so, it first assembles authentication data for the message as follows:

Authentication Data

```
00 00 ff f7 00 00 10 00 80 00 00 80 5f 9b 34 fb ①
00 ②
70 72 37 7d 00 02 42 1c b1 63 49 1c 27 33 3a 61 ③
00 ④
```

① 128-bit service UUID, in this case expanded from 16-bit service UUID according to standard Bluetooth UUID processing rules

② Service instance ID

③ 128-bit characteristic UUID

④ Characteristic instance ID

And a nonce as follows:

Nonce

```
01 00 00 00 00 ee 1f 00 ①
02 00 00 00 ②
05 ③
```

① The IEEE EUI-64 of the sending device

② Security frame counter, in little-endian format. Here security frame counter = 2

③ Security control field. Here 5: Encryption with 32-bit MIC

Finally, it runs AES-CCM with above session key, nonce, authentication data, and message payload. This yields the following encrypted and authenticated data:

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

```
23 b7 11 a6 91 6a ba 32 82 b3 7a 7c b4 4f 30 18 7c 43 c2 45 d3 bd 6f f1 af e5 9d 0b
3a 9b 0f da c5 26 d3 63 3d ae df 73 5d 78 35 fe 9f || 3f 12 bd f8
```

Finally it adds the frame counter as security header in front of the AES-CCM output to obtain the full attribute value as it is being written to the characteristic:

Resulting Attribute Value

```
02 00 00 00 ①
23 b7 11 a6 91 6a ba 32 82 b3 7a 7c b4 4f 30 18 7c 43 c2 45 d3 bd 6f f1 af e5 9d 0b
3a 9b 0f da c5 26 d3 63 3d ae df 73 5d 78 35 fe 9f ②
3f 12 bd f8 ③
```

① Frame Counter

② Encrypted Message

③ 32-bit Message Integrity Code

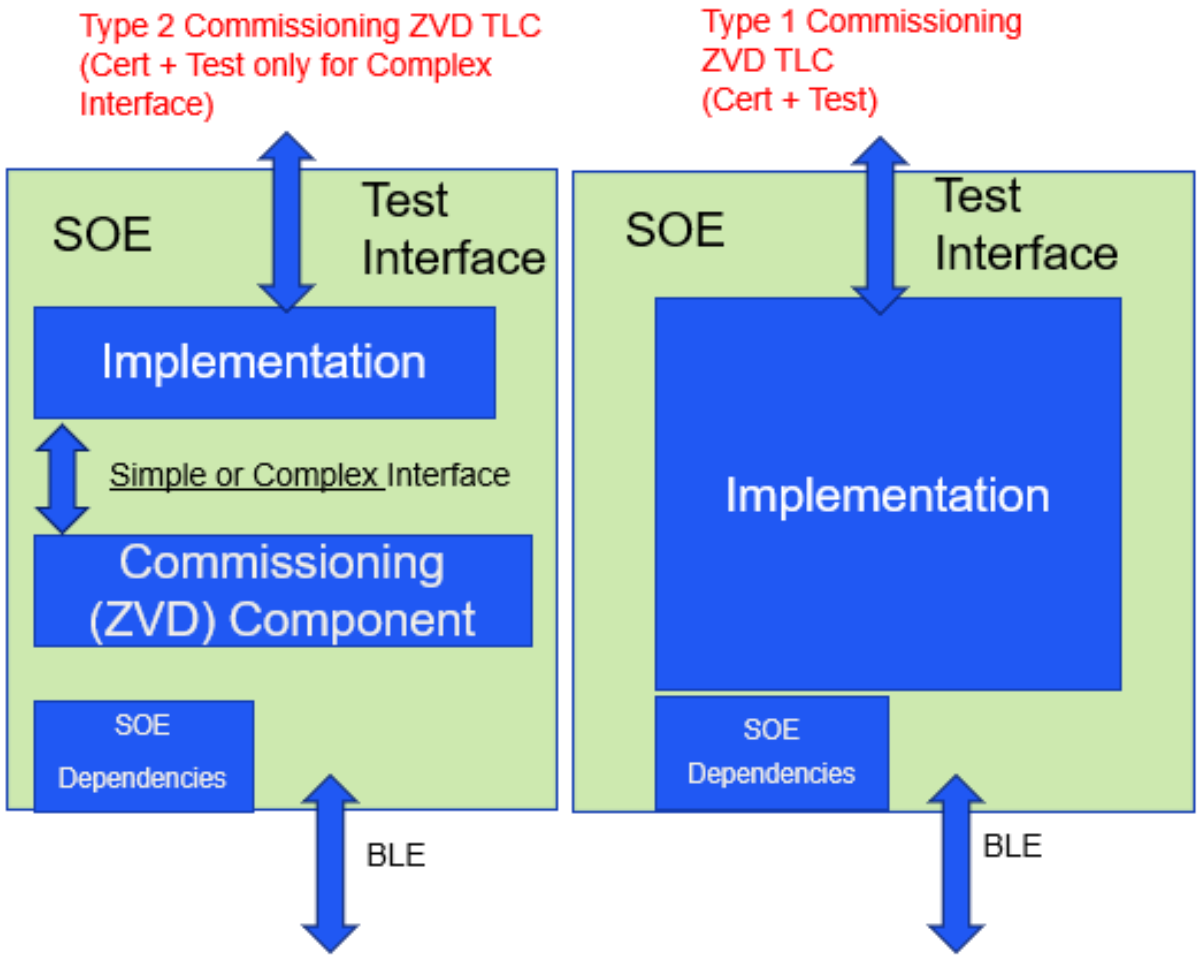
Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Annex C: Zigbee Direct Component Specifications

This section specifies a number of software components which can be certified as per the rules layed out in [R11]. These components have been defined specifically with Zigbee Direct usecases and requirements in mind, however there are no restrictions on using these components outside the scope of Zigbee Direct.

The figure Block diagrams of a Commissioning ZVD shows block diagrams outlining the options for implementing a Commissioning ZVD, i.e. a ZVD without the optional Tunnel Service. As per [R11] a Type 1 implementation is implementing all required functionality in one piece of software whereas a Type 2 implementation is taking advantage of separately (pre-)certified components.

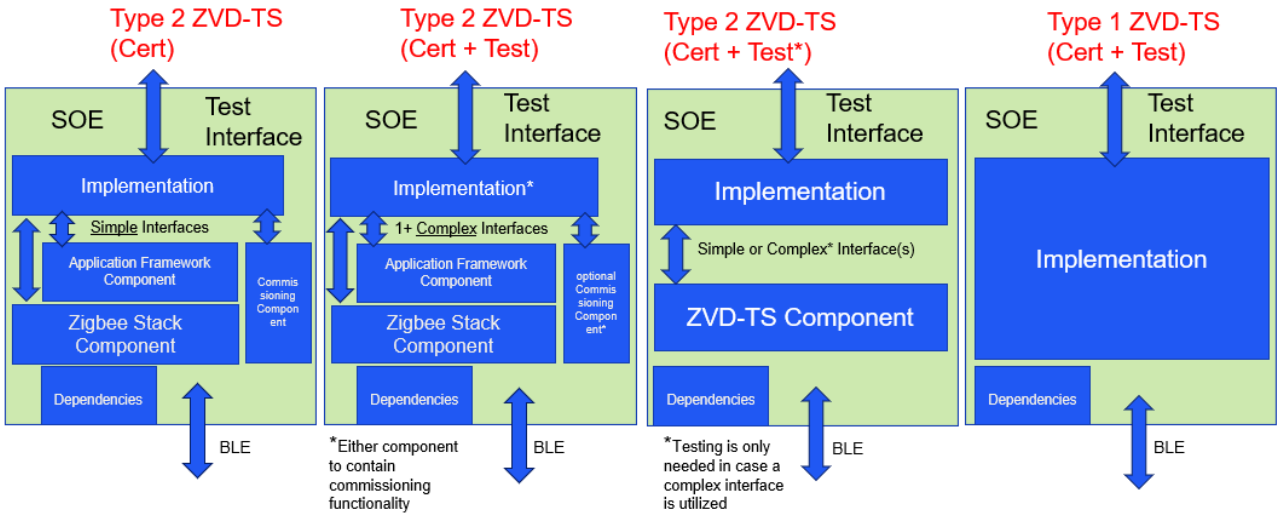
Block diagrams of a Commissioning ZVD



The figure Block diagrams of a ZVD-TS shows block diagrams outlining the options for implementing a ZVD-TS, i.e. a ZVD which includes the optional Tunnel Service.

Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Block diagrams of a ZVD-TS



C.1. Zigbee Direct Commissioning ZVD Component Specification [2]

A Zigbee Direct Commissioning ZVD Component covers all functionality described in section 8, that is required by a ZVD which is not making use of the optional Tunnel Service (i.e. it is exclusively supporting the functionality of both Commissioning and Security Services). This component can be either a User Interface Component (UIC), or a regular software component which provides a simple interface to the next higher layer component.

C.1.1. Testing and Certification

Certification of a Zigbee Direct Commissioning ZVD Component involves undergoing all required testcases detailed in [R2] for the device role of ZDZVD-C. In case a UIC with this functionality is build on a previously certified Zigbee Direct Commissioning ZVD Software Component only a subset of testing is required as outlined in [R2] for the device role of ZDZVD-C.

C.1.2. Applicable Supported Operating Environments

Any Supported Operating Environments which fulfills the general requirements on a SOE as layed out in [R11] and provides a BLE interface fulfilling the requirements layed out in section 7.4.

C.1.3. Next Higher Layer Component Interface Specification

Unless this component is a UIC, the Next Higher Layer Component Interface of the Zigbee Direct Commissioning ZVD Component SHALL be a Simple Interface, it SHALL NOT have a complex interface. To qualify as a simple interface, the next higher layer SHALL be able to operate the Commissioning ZVD Component without detailed knowledge of the BLE Services and

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

Characteristics on one hand and the formation of payloads and TLVs (including the security related exchanges) on the other hand. The detailed interface definition is manufacturer specific and out of scope of this document.

C.1.4. Lower Component Interface

The Zigbee Direct Commissioning ZVD Component SHALL enable the over the air BLE behaviour as described in this document and the accompanying test specification [R2] via one or more dependent lower layer components or SOE dependencies like for example a specific BLE stack. The Zigbee Direct Commissioning ZVD Component vendor SHALL declare all required dependencies or range of dependencies.

C.2. Application Framework Component Specification[3]

The Application Framework Software Component covers the functionality required to send and receive messages as described in [R8] (ZCL) and in addition it covers the functionality described in [R10] (BDB). In order to certify an Application Framework Software Component it is required to undergo testing for all implemented and certifiable server as well as client clusters as outlined in the respective test specifications.

C.2.1. Testing and Certification

Testing and certification of the Application Framework Software Component covers both the BDB as well as ZCL requirements as outlined below:

For ZCL related testing, the regular test setup for Zigbee Device certification may be employed, but with the important modification that the testcases are to be exercised with the Application Framework Software Component running on a device with Zigbee tunneling capability (e.g. a ZVD-TS Component as described in section C.4), which is connected to a bridging device (e.g. reference ZDD as specified in section 7) which is bridging the ZCL messages into the IEEE802.15.4 domain where the standard test tools can be employed. Support for not (yet) certifyable as well as manufacturer specific clusters is permitted and does not require additional testing. In case only manufacturer specific clusters or not (yet) certifyable clusters are implemented, certification is not possible.
In case a new version of the Application Framework Software Component is released, which exclusively adds support for new, certifyable clusters, or clusters which originally weren't certifiable, but have now become certifiable in the meantime due to extensions of the relevant test specification, isolated testing for those clusters is required to uphold the validity of the certification.

To Prove compliance to the BDB specification, again the regular test setup for Zigbee Device certification may be employed, but with the important modification that the testcases are to be exercised with the Application Framework Software Component running on a device with Zigbee tunneling capability (e.g. a ZVD-TS Component as described in section C.4), which is connected to

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

a bridging device (e.g. reference ZDD as specified in section 7) which is bridging the Zigbee messages into the IEEE802.15.4 domain where the standard test tools can be employed.+

C.2.2. Applicable Supported Operating Environments

Any Supported Operating Environments which fulfill the general requirements on a SOE as layed out in [R11]. When used in the scope of a Zigbee Direct ZVD, the SoE is required to provide a BLE interface fulfilling the requirements layed out in section 7.4.

C.2.3. Next Higher Layer Component Interface Specification

Both the ZCL functionality as well as the BDB functionality as implemented by the Application Framework Software Component can be exposed to the Next Higher Layer Component in the form of a Simple Interface, a Complex Interface, or a combination of both.

An interface which does not require the next higher layer to generate the actual ZCL payload, but instead uses a simplified abstraction SHALL be considered simple.

Examples of a simple interface include commands to
- read or write each individual attribute of a cluster
- send individual cluster commands
- receive a callback for recived cluster commands
- receive a callback on changes to attributes
- set up reporting of individual attributes

Any interface requiering the next higher layer component to have knowledge of the ZCL message format, command IDs, attribute IDs etc. is considered complex.

In case the BDB component - when invoked - will under any case guarantee behaviour compliant to the requirements of [R10] is considered as having a simple interface. In case non-compliant behaviour can be triggered by any means, the interface is to be considered complex.

The detailed interface definition is manufacturer specific and out of scope of this document.

C.2.4. Lower Component Interface

Application Framework Software Component shall interface to the Zigbee Stack Component as defined in section C.3 ; it may be able to interface with a Zigbee Stack Component of more than one vendor. The Application Framework Software Component vendor SHALL declare all required dependencies or range of dependencies.

C.3. Zigbee Stack Component Specification

A Zigbee Stack Component consists of a Zigbee PRO stack conforming to the Zigbee PRO stack specifications [R4] or [R12][20].

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

C.3.1. Testing and Certification

In case the Zigbee PRO stack contained within this component has previously been part of a successful ZCP certification, only a subset of the testcases required for this component are required under the following premises:

1. The original specification [R4] or [R12] against which ZCP testing was performed has not been sunset
2. On top of any neccessary adaptation to the SOE no further changes have been applied to the Zigbee PRO Stack
3. Using an Exemplary Test Platform, the Zigbee Stack Component is tested to pass the required testcases for a Zigbee Stack Component (ZDSC-I) as outlined in [R2].

In case the Zigbee PRO stack contained within this component is not applicable to the above the full set of testcases for a Zigbee Stack Component (ZDSC-N) apply.

C.3.2. Applicable Supported Operating Environments

Any Supported Operating Environments which fulfill the general requirements on a SOE as layed out in [R11] and provide a BLE interface fulfilling the requirements layed out in section 7.4 or another interface capable of exchanging Zigbee messages in a way covered by a specification of the CSA Alliance.

C.3.3. Next Higher Layer Component Interface Specification

The Next Higher Layer Component Interface of the Zigbee Stack Component provides a predominantly complex interface as defined in [R11] to the next higher layer component. The interface required for formation, joining and maintenance of a Zigbee network MAY be simple, but the interface required for sending and receiving messages and paramertrizing the Zigbee PRO stack SHALL be considered complex.

C.3.4. Lower Component Interface

The Zigbee Stack Component enables the over the air Zigbee behaviour as described in this document and the accompanying test specification [R2] via one or more dependent lower layer components or SOE dependencies like for example a specific Bluetooth stack. The Zigbee Stack Component vendor SHALL state all required dependencies or range of dependencies for the Zigbee Stack Component.

C.4. ZVD-TS Component Specification

A ZVD-TS Component is implementing the functionality of a ZVD as outlined in section 8 including the optional Tunnel Service.

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

C.4.1. Testing and Certification

A ZVD-TS Component SHALL be architected in one of the following two ways:

1. Type 2 UIC: A ZVD-TS Component build on dependent Application Framework and Zigbee Stack Software Components. In case any part of a complex interface of the dependent Application Framework component is employed, certification of a ZVD-TS Component additionally includes any testing outlined for ZVDAF in [R2]. Furthermore a ZVD-TS Component MAY be build upon a Zigbee Direct Commissioning ZVD Component. Alternatively the ZVD-TS Component MAY chose to implement the required functionality without the use of either of these components, in which case certification includes passing any testcases outlined for ZDZVD-C in [R2].
2. Type 1 UIC: A ZVD-TS Component which implementation does not rely on dependent Application Framework and Zigbee Stack Software Components SHALL pass testcases as outlined in [R2] for ZVDSC-N, ZDZVD-C and ZVDAF as a prerequisite for certification.

C.4.2. Applicable Supported Operating Environments

Any Supported Operating Environments which fulfill the general requirements on a SOE as layed out in [R11] and provide a BLE interface fulfilling the requirements layed out in section 7.4.

C.4.3. Next Higher Layer Component Interface Specification

In most cases the ZVD-TS Component is expected to be a User Interface Component, however the Next Higher Layer Component Interface of the ZVD-TS Component may also be a Simple Interface, a Complex Interface, or a combination of both. A simple interface to the ZVD-TS component SHALL fullfil the requirements imposed on a simple interface by the Application Framework and Commissioning Software Component Specifications.
The detailed interface definition is manufacturer specific and out of scope of this document.

C.4.4. Lower Component Interface

The ZVD-TS Component shall enable the over the air BLE behaviour as described in this document and the accompanying test specification [R2] via multiple dependent lower layer components or SOE dependencies like for example a specific BLE stack. The ZVD-TS Component vendor SHALL state all required dependencies or range of dependencies.

[1] The version and date information in these references was correct at the time this document was released.

[2] CCB 3982: Megerd with Zigbee Direct Commissioning Component as both can be TLC and component

[3] CCB 3962: Merged ZCL and BDB Software Components into Application Framework Software Component

[4] CCB 3640

[5] A 16-bit UUID assigned by the Bluetooth SIG can be converted to a 128-bit UUID by usage of the Bluetooth Base UUID in the following way: '0000xxxx-0000-1000-8000-00805F9B34FB', where 'xxxx' is the 16-bit UUID.

[6] See <<R3>> Vol 3, Part G, Section 3.3.3.5.2 for an overview of Bluetooth GATT Characteristic Format Types

[7] CCB 3604

[8] Refer to typical values such as provided in https://developer.apple.com/accessories/Accessory-Design-Guidelines.pdf and https://interrupt.memfault.com/blog/ble-throughput-primer

 Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.

[9] CCB 3640

[10] CCB 3606

[11] The maximum size of the length field value (0xF1=241 octets) is determined by taking the maximum usable MTU size in a BLE Write Request (=244 octets), subtract the TLV header overhead (1 octet each for Type and Length fields) and subtract 1 octet for the minus one in the definition of the length field.

[12] The maximum value of the NPDU Length field (0x62=98 octets) is determined by taking the maximum 802.15.4 MPDU size (=127 octets), subtract the minimum MAC frame header size (9 octets) and MAC frame trailer size (2 octets), and subtract the size of the NWK auxiliary header (14 octets) and MIC (2 octets).

[13] A potential issue with BLE indications exists if two subsequent upcoming indications are overlapped. This might happen on the ZVD side if after the first indication has been received and confirmed by a BLE stack and the application is signaled, another indication occurs and the application reads new data instead of previous one. Please make sure your particular BLE stack explicitly provides the received data frame inside an Indication received event, or indication confirmation can be triggered by an application after the data has been read.

[14] CCB 3691

[15] CCB 3692

[16] CCB 4010: Repeal CCB 3961

[17] CCB 3663

[18] CCB 3693

[19] CCB 3656

[20] CCB 3841

csa connectivity standards alliance Copyright © 2024, Connectivity Standards Alliance, Inc. All rights reserved.