Link Manager Protocol (LMP)

***Bluetooth*® Test Suite**

- **Revision:** LMP.TS.p50
- **Revision Date:** 2026-05-05
- **Prepared By:** BTI
- **Published during TCRL:** TCRL.pkg103

This document, regardless of its title or content, is not a Bluetooth Specification as defined in the Bluetooth Patent/Copyright License Agreement ("PCLA") and Bluetooth Trademark License Agreement. Use of this document by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG Inc. ("Bluetooth SIG") and its members, including the PCLA and other agreements posted on Bluetooth SIG's website located at www.bluetooth.com.

THIS DOCUMENT IS PROVIDED "AS IS" AND BLUETOOTH SIG, ITS MEMBERS, AND THEIR AFFILIATES MAKE NO REPRESENTATIONS OR WARRANTIES AND DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, THAT THE CONTENT OF THIS DOCUMENT IS FREE OF ERRORS.

TO THE EXTENT NOT PROHIBITED BY LAW, BLUETOOTH SIG, ITS MEMBERS, AND THEIR AFFILIATES DISCLAIM ALL LIABILITY ARISING OUT OF OR RELATING TO USE OF THIS DOCUMENT AND ANY INFORMATION CONTAINED IN THIS DOCUMENT, INCLUDING LOST REVENUE, PROFITS, DATA OR PROGRAMS, OR BUSINESS INTERRUPTION, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, AND EVEN IF BLUETOOTH SIG, ITS MEMBERS, OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document is proprietary to Bluetooth SIG. This document may contain or cover subject matter that is intellectual property of Bluetooth SIG and its members. The furnishing of this document does not grant any license to any intellectual property of Bluetooth SIG or its members.

This document is subject to change without notice.

Copyright © 2004–2026 by Bluetooth SIG, Inc. The Bluetooth word mark and logos are owned by Bluetooth SIG, Inc. Other third-party brands and names are the property of their respective owners.

Contents

1 Scope 1-1

2 References, definitions, and abbreviations 2-1

2.1 References 2-1

2.2 Definitions 2-1

2.3 Acronyms and abbreviations 2-1

3 Test Suite Structure (TSS) 3-1

3.1 References 3-1

3.2 Overview 3-1

3.3 Test Suite Structure (TSS) 3-4

3.3.1 Test groups 3-4

3.3.2 Protocol groups 3-4

3.3.3 Behavior testing groups 3-5

3.3.4 HCI Command and Event Version 3-5

4 Test cases (TC) 4.1-1

4.1 Introduction 4.1-1

4.1.1 References 4.1-1

4.1.2 Test case identification conventions 4.1-2

4.1.3 Conformance 4.1-2

4.1.4 Baseband assumptions 4.1-3

4.1.5 Role Switch 4.1-3

4.1.6 Applicable Parameter Values 4.1-3

4.1.7 Advertisement of Features for test cases 4.1-3

4.2 Default settings 4.2-1

4.2.1 References 4.2-1

4.2.2 Authentication 4.2-2

4.2.3 Encryption 4.2-3

4.2.4 Information Requests 4.2-4

4.2.5 Link Handling 4.2-5

4.2.6 Secure Simple Pairing 4.2-6

4.2.7 AES-CCM Encryption 4.2-7

4.2.8 Secure Simple Pairing P-256 ..... 4.2-8

4.3 Preambles 4.3-1

4.3.1 References 4.3-1

4.3.2 Connection Establishment IUT Central 4.3-2

4.3.3 Connection Establishment Lower Tester 4.3-3

4.3.4 Default settings 4.3-4

4.3.5 External Frame Configuration 4.3-5

4.3.6 Pass/Inconclusive/Fail verdict conventions 4.3-5

4.4 Common Packet Contents 4.4-1

4.4.1 References 4.4-1

4.4.2 Fields and Bits Reserved for Future Use 4.4-1

4.5 Authentication procedures 4.5-1

4.5.1 References 4.5-1

4.5.2 Authentication – Both Central and Peripheral 4.5-1

LMP/AUT/BI-01-C [Error Return When a Unit Key is Requested] 4.5-1

LMP/AUT/BV-01-C [Authentication Reject, No Link Key] 4.5-3

LMP/AUT/BV-36-C [Legacy Authentication of Previously Authenticated or Stored Link Key] 4.5-4
LMP/AUT/BV-40-C [Legacy Authentication of Previously Authenticated or Stored Link Key, v6.0] 4.5-5
4.5.3 Pairing – Both Central and Peripheral 4.5-7
LMP/AUT/BV-03-C [Create Link Key] 4.5-7
LMP/AUT/BV-04-C [Pairing, IUT Initiator] 4.5-9
LMP/AUT/BV-52-C [Pairing, IUT Initiator – HCI Command Required to Pair] 4.5-9
LMP/AUT/BV-05-C [IUT Responder, Fixed PIN] 4.5-12
LMP/AUT/BV-06-C [IUT Initiator; Responder has Fixed PIN] 4.5-14
LMP/AUT/BV-53-C [IUT Initiator; Responder has Fixed PIN – HCI Command Required to Pair] 4.5-14
LMP/AUT/BV-24-C [Create Link Key – Rejects Bad Authentication Response] 4.5-17
LMP/AUT/BV-25-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Optional] 4.5-19
LMP/AUT/BV-54-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Optional, HCI Command Required to Pair] 4.5-19
LMP/AUT/BV-45-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Mandatory] 4.5-19
LMP/AUT/BV-55-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Mandatory, HCI Command Required to Pair] 4.5-19
LMP/AUT/BV-26-C [IUT Responder, Fixed PIN – Rejects Bad Authentication Response] 4.5-22
LMP/AUT/BV-27-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Optional] 4.5-24
LMP/AUT/BV-56-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Optional, HCI Command Required to Pair] 4.5-24
LMP/AUT/BV-46-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Mandatory] 4.5-25
LMP/AUT/BV-57-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Mandatory, HCI Command Required to Pair] 4.5-25
LMP/AUT/BV-34-C [Pairing, IUT rejects Pairing Procedure – Host is in Non-Pairable Mode] 4.5-28
LMP/AUT/BI-04-C [Reject Role Switch, Pairing, Responder] 4.5-29
LMP/AUT/BI-08-C [Pairing, IUT Initiator, Invalid Combination Key] 4.5-31
4.5.4 Change Link Key – Both Central and Peripheral 4.5-33
LMP/AUT/BV-12-C [Change Link Key, IUT Responder] 4.5-33
LMP/AUT/BV-13-C [Change Link Key, IUT Initiator] 4.5-34
LMP/AUT/BV-28-C [Change Link Key, IUT Responder – Rejects Bad Authentication Response] 4.5-36
LMP/AUT/BV-29-C [Change Link Key, IUT Initiator – Rejects Bad Authentication Response, Detach Optional] 4.5-37
LMP/AUT/BV-47-C [Change Link Key, IUT Initiator – Rejects Bad Authentication Response, Detach Mandatory] 4.5-37
4.5.5 Secure Authentication procedures 4.5-39
LMP/AUT/BV-14-C [Secure Authentication, Responder (IUT) has link key, Initiator (Lower Tester) is Central] 4.5-39
LMP/AUT/BV-15-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Peripheral] 4.5-40
LMP/AUT/BV-16-C [Secure Authentication, Responder (Lower Tester) has link Key, Initiator (IUT) is Central] 4.5-41
LMP/AUT/BV-17-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral] 4.5-43
LMP/AUT/BV-18-C [Role switch during Secure Authentication before Authentication Response, Responder (IUT) has link key, Initiator (Lower Tester) is Central] 4.5-44
LMP/AUT/BV-19-C [Role switch during Secure Authentication before Authentication Response, Responder (IUT) has link key, Initiator (Lower Tester) is Peripheral] 4.5-46
LMP/AUT/BV-20-C [Role switch during Secure Authentication before Authentication Response, Responder (Lower Tester) has link key, Initiator (IUT) is Central] 4.5-48
LMP/AUT/BV-21-C [Role switch during Secure Authentication before Authentication Response, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] 4.5-50
LMP/AUT/BV-22-C [Role switch during Secure Authentication before Random Number, Responder (Lower Tester) has link key, Initiator (IUT) is Central] 4.5-52

LMP/AUT/BV-23-C [Role switch during Secure Authentication before Random Number, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] 4.5-54
LMP/AUT/BI-02-C [Mistimed role switch during Secure Authentication, Responder (IUT) has link key, Initiator (Lower Tester) is Central] 4.5-56
LMP/AUT/BI-03-C [Mistimed role switch during Secure Authentication, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] 4.5-58
LMP/AUT/BV-30-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Central – Rejects Bad Authentication Response, Detach Optional] 4.5-60
LMP/AUT/BV-48-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Central – Rejects Bad Authentication Response, Detach Mandatory] 4.5-60
LMP/AUT/BV-31-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Peripheral – Rejects Bad Authentication Response, Detach Optional] 4.5-62
LMP/AUT/BV-49-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Peripheral – Rejects Bad Authentication Response, Detach Mandatory] 4.5-62
LMP/AUT/BV-32-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Central – Rejects Bad Authentication Response, Detach Optional] 4.5-64
LMP/AUT/BV-50-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Central – Rejects Bad Authentication Response, Detach Mandatory] 4.5-64
LMP/AUT/BV-33-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral – Rejects Bad Authentication Response, Detach Optional] 4.5-66
LMP/AUT/BV-51-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral – Rejects Bad Authentication Response, Detach Mandatory] 4.5-66
LMP/AUT/BV-35-C [Secure Authentication, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] 4.5-68
LMP/AUT/BI-06-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Central, Reject Role Switch] 4.5-70
LMP/AUT/BI-07-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Peripheral, Reject Role Switch] 4.5-72
LMP/AUT/BV-37-C [Secure Authentication of Previously Authenticated or Stored Link Key] 4.5-73
4.5.6 Legacy Authentication procedures 4.5-75
LMP/AUT/BV-38-C [Mutual Legacy Authentication, Initiator, Central] 4.5-75
LMP/AUT/BV-39-C [Mutual Legacy Authentication, Initiator, Peripheral] 4.5-75
LMP/AUT/BV-41-C [Delete Stored Link Key, In Connection] 4.5-77
LMP/AUT/BV-42-C [Repeated Authentication Failure] 4.5-81
LMP/AUT/BV-43-C [Repeated Authentication Failure, Secure Authentication, Central] 4.5-83
LMP/AUT/BV-44-C [Repeated Authentication Failure, Secure Authentication, Peripheral] 4.5-85
4.6 Encryption 4.6-1
4.6.1 References 4.6-1
4.6.2 Encryption – Peripheral 4.6-1
LMP/ENC/BV-01-C [Accept Encryption] 4.6-2
LMP/ENC/BV-26-C [Accept AES-CCM Encryption Request] 4.6-2
LMP/ENC/BV-33-C [Accept AES-CCM Encryption Request – Legacy Host] 4.6-2
LMP/ENC/BV-02-C [Accept Broadcast Encryption] 4.6-4
LMP/ENC/BV-66-C [Accept Encryption] 4.6-6
LMP/ENC/BV-67-C [Accept Broadcast Encryption] 4.6-8
LMP/ENC/BV-71-C [Accept AES-CCM Encryption Request] 4.6-11
LMP/ENC/BV-72-C [Accept AES-CCM Encryption Request – Legacy Host] 4.6-13
LMP/ENC/BV-04-C [Stop Encryption, Central Command] 4.6-16
LMP/ENC/BV-09-C [Stop Encryption, Host Command] 4.6-16
LMP/ENC/BV-11-C [Semi-permanent Link Key] 4.6-17
LMP/ENC/BV-68-C [Semi-permanent Link Key] 4.6-19
LMP/ENC/BV-12-C [Reject Broadcast Encryption] 4.6-23
LMP/ENC/BV-15-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Peripheral Initiated] 4.6-24
LMP/ENC/BV-38-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Peripheral Initiated as a result of change connection link Key] 4.6-24

LMP/ENC/BV-17-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Peripheral Initiated with Role Switch] 4.6-27
LMP/ENC/BV-42-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Peripheral Initiated with role switch] 4.6-27
LMP/ENC/BV-22-C [Initiate Encryption] 4.6-30
LMP/ENC/BV-25-C [Initiate AES-CCM Encryption] 4.6-30
LMP/ENC/BV-69-C [Initiate Encryption] 4.6-31
LMP/ENC/BV-70-C [Initiate AES-CCM Encryption] 4.6-34
LMP/ENC/BV-23-C [Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated as a result of change connection link Key] 4.6-37
LMP/ENC/BV-40-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Central Initiated as a result of change connection link Key] 4.6-37
LMP/ENC/BV-24-C [Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated with role switch] 4.6-40
LMP/ENC/BV-44-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Central Initiated with role switch] 4.6-40
LMP/ENC/BV-51-C [Key_Size Negotiation as Peripheral - E0, Acceptor] 4.6-43
LMP/ENC/BV-52-C [Key_Size Negotiation as Peripheral - AES, Acceptor] 4.6-43
LMP/ENC/BV-55-C [Key_Size Negotiation as Peripheral - E0, Initiator] 4.6-43
LMP/ENC/BV-56-C [Key_Size Negotiation as Peripheral - AES, Initiator] 4.6-43
LMP/ENC/BV-59-C [Key_Size Negotiation as Peripheral - E0, Acceptor] 4.6-43
LMP/ENC/BV-60-C [Key_Size Negotiation as Peripheral - AES, Acceptor] 4.6-43
LMP/ENC/BV-61-C [Key_Size Negotiation as Peripheral - E0, Initiator] 4.6-43
LMP/ENC/BV-62-C [Key_Size Negotiation as Peripheral - AES, Initiator] 4.6-43
LMP/ENC/BI-01-C [Encryption, Peripheral, Reject Role Switch] 4.6-46
LMP/ENC/BI-03-C [Reject Start Encryption Request, Encrypted Connection, Peripheral] 4.6-48
LMP/ENC/BV-19-C [Stopping and Restarting Encryption with Legacy Device - Peripheral Initiated] 4.6-49
LMP/ENC/BV-73-C [Key_Size Negotiation as Peripheral - E0, Acceptor] 4.6-52
LMP/ENC/BV-74-C [Key_Size Negotiation as Peripheral - AES, Acceptor] 4.6-52
LMP/ENC/BV-75-C [Key_Size Negotiation as Peripheral - E0, Initiator] 4.6-52
LMP/ENC/BV-76-C [Key_Size Negotiation as Peripheral - AES, Initiator] 4.6-52
LMP/ENC/BV-77-C [Key_Size Negotiation as Peripheral - E0, Acceptor] 4.6-52
LMP/ENC/BV-78-C [Key_Size Negotiation as Peripheral - AES, Acceptor] 4.6-52
LMP/ENC/BV-79-C [Key_Size Negotiation as Peripheral - E0, Initiator] 4.6-52
LMP/ENC/BV-80-C [Key_Size Negotiation as Peripheral - AES, Initiator] 4.6-53
LMP/ENC/BV-81-C [Verify Key Size Mask] 4.6-56
4.6.3 Encryption – Central 4.6-57
LMP/ENC/BV-05-C [Initiate Encryption] 4.6-57
LMP/ENC/BV-34-C [Initiate AES-CCM Encryption] 4.6-57
LMP/ENC/BV-50-C [Initiate AES-CCM Encryption – Legacy Host] 4.6-57
LMP/ENC/BV-82-C [Initiate Encryption] 4.6-59
LMP/ENC/BV-85-C [Initiate AES-CCM Encryption] 4.6-61
LMP/ENC/BV-86-C [Initiate AES-CCM Encryption – Legacy Host] 4.6-63
LMP/ENC/BV-06-C [Peripheral Declines Encryption] 4.6-65
LMP/ENC/BV-07-C [Initiate Encryption Stop] 4.6-67
LMP/ENC/BV-08-C [Stop Encryption, Peripheral Request] 4.6-67
LMP/ENC/BV-10-C [Initiate Broadcast Encryption] 4.6-69
LMP/ENC/BV-83-C [Initiate Broadcast Encryption] 4.6-72
LMP/ENC/BV-13-C [Initiate Semi-permanent Link Key Change] 4.6-75
LMP/ENC/BV-84-C [Initiate Semi-permanent Link Key Change] 4.6-77
LMP/ENC/BV-14-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Central Initiated] 4.6-80
LMP/ENC/BV-37-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Central Initiated as a result of change connection link Key] 4.6-80
LMP/ENC/BV-16-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Central Initiated with Role Switch] 4.6-83

LMP/ENC/BV-41-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Central Initiated with role switch] 4.6-83
LMP/ENC/BV-18-C [Starting and Stopping Encryption with Legacy Device - Central Initiated] 4.6-85
LMP/ENC/BV-20-C [Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated as a Result of Change Connection Link Key] 4.6-88
LMP/ENC/BV-39-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Peripheral Initiated as a result of change connection link Key] 4.6-88
LMP/ENC/BV-21-C [Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated with Role Switch] 4.6-91
LMP/ENC/BV-43-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Peripheral Initiated with role switch] 4.6-91
LMP/ENC/BV-45-C [Broadcast Encryption is not used with AES-CCM encryption] 4.6-93
LMP/ENC/BV-53-C [Key_Size Negotiation as Central - E0] 4.6-95
LMP/ENC/BV-54-C [Key_Size Negotiation as Central - AES] 4.6-95
LMP/ENC/BV-63-C [Key_Size Negotiation as Central - E0] 4.6-95
LMP/ENC/BV-64-C [Key_Size Negotiation as Central - AES] 4.6-95
LMP/ENC/BV-87-C [Key_Size Negotiation as Central - E0] 4.6-99
LMP/ENC/BV-88-C [Key_Size Negotiation as Central - AES] 4.6-99
LMP/ENC/BV-89-C [Key_Size Negotiation as Central - E0] 4.6-99
LMP/ENC/BV-90-C [Key_Size Negotiation as Central - AES] 4.6-99
LMP/ENC/BV-57-C [Broadcast Encryption, Link Key Selection Request] 4.6-102
LMP/ENC/BV-58-C [Point-To-Point Encryption, Link Key Selection Request] 4.6-103
LMP/ENC/BI-02-C [Encryption, Central, Reject Role Switch] 4.6-104
4.6.4 Encryption – Both Connected Roles 4.6-107
LMP/ENC/BI-04-C [Reject Encryption Commands, Unencrypted Connection, Peripheral, Encryption Mode Request] 4.6-107
LMP/ENC/BI-05-C [Reject Encryption Commands, Unencrypted Connection, Peripheral, Pause Encryption Request] 4.6-107
LMP/ENC/BI-06-C [Reject Encryption Commands, Unencrypted Connection, Peripheral, Pause Encryption AES Request] 4.6-107
LMP/ENC/BI-07-C [Reject Encryption Commands, Unencrypted Connection, Central, Encryption Mode Request] 4.6-107
LMP/ENC/BI-08-C [Reject Encryption Commands, Unencrypted Connection, Central, Pause Encryption Request] 4.6-107
LMP/ENC/BI-09-C [Reject Encryption Commands, Unencrypted Connection, Central, Pause Encryption AES Request] 4.6-107
LMP/ENC/BV-28-C [Stop AES-CCM Encryption from Host] 4.6-108
LMP/ENC/BV-35-C [Initiate AES-CCM Encryption Stop] 4.6-108
LMP/ENC/BV-29-C [Combating forged acknowledgements when AES-CCM Encryption is enabled] 4.6-110
LMP/ENC/BV-46-C [Combating forged acknowledgements when AES-CCM Encryption is enabled] 4.6-110
LMP/ENC/BV-27-C [Stop AES-CCM Encryption from Central] 4.6-111
LMP/ENC/BV-36-C [Stop AES-CCM Encryption, Peripheral request] 4.6-111
LMP/ENC/BV-30-C [Responding to LMP_PING_REQ when AES-CCM Encryption is enabled] 4.6-112
LMP/ENC/BV-47-C [Responding to LMP_PING_REQ when AES-CCM Encryption is enabled] 4.6-112
LMP/ENC/BV-31-C [No response to LMP_PING_REQ] 4.6-113
LMP/ENC/BV-48-C [No response to LMP_PING_REQ] 4.6-113
LMP/ENC/BV-32-C [Modified Authentication Payload Timeout] 4.6-114
LMP/ENC/BV-49-C [Modified Authentication Payload Timeout] 4.6-114
4.7 Information Requests 4.7-1
4.7.1 References 4.7-1
4.7.2 Clock_Offset Request – Peripheral 4.7-1
LMP/INF/BV-01-C [Clock_Offset Response] 4.7-1
4.7.3 Clock_Offset Request – Central 4.7-2
LMP/INF/BV-02-C [Clock_Offset Request] 4.7-2
4.7.4 Timing Accuracy Information Request – Both Central and Peripheral 4.7-3
LMP/INF/BV-05-C [Timing Accuracy Response] 4.7-3

4.7.5 LMP Version – Both Central and Peripheral 4.7-4
LMP/INF/BV-08-C [Version/Company ID Response] 4.7-4
LMP/INF/BV-09-C [Request LMP Version] 4.7-5
4.7.6 Supported Features – Both Central and Peripheral 4.7-6
LMP/INF/BV-10-C [Supported Features Response] 4.7-6
LMP/INF/BV-11-C [Supported Features Request] 4.7-7
LMP/INF/BV-16-C [Extended_Features Request] 4.7-8
LMP/INF/BV-17-C [Extended_Features Response] 4.7-10
4.7.7 Name Request – Both Central and Peripheral 4.7-10
LMP/INF/BV-12-C [Name Response] 4.7-10
LMP/INF/BV-13-C [Name Request] 4.7-11
LMP/INF/BV-18-C [Remote Name Request - IUT Initiator] 4.7-13
LMP/INF/BV-19-C [Remote Name Request - Legacy remote device without Extended_Features - IUT Initiator] 4.7-13
LMP/INF/BV-22-C [LE Features in LMP Feature Set] 4.7-16
LMP/INF/BV-23-C [LE Features in LMP Feature Set, LE Supported] 4.7-16
LMP/INF/BV-24-C [LE Features in LMP Feature Set, LE Supported and Simultaneous LE and BR/EDR] 4.7-16
4.7.8 Invalid Packet Handling 4.7-18
LMP/LIH/BI-06-C [Ignore LLID = 0b00] 4.7-18
LMP/LIH/BI-07-C [Ignore LMP packets with the wrong packet type] 4.7-20
LMP/LIH/BI-08-C [Ignore LMP packets with the wrong packet type] 4.7-20
LMP/LIH/BI-09-C [Ignore APB Packets with LMP LLID 0b11, DM3/DM5] 4.7-21
LMP/LIH/BI-10-C [LMP PDU Incorrect Length] 4.7-22
4.8 Link Handling 4.8-1
4.8.1 References 4.8-1
4.8.2 Role Switch – Peripheral 4.8-1
LMP/LIH/BV-01-C [Initiate Role Switch] 4.8-1
LMP/LIH/BV-79-C [Role Switch at Setup, Peripheral] 4.8-3
LMP/LIH/BV-144-C [Rejected Role Switch request at Setup, Peripheral] 4.8-5
LMP/LIH/BI-04-C [Reject Role Switch at Setup, Peripheral] 4.8-7
4.8.3 Role Switch – Central 4.8-11
LMP/LIH/BV-02-C [Accept Role Switch] 4.8-11
LMP/LIH/BV-78-C [Role Switch at Setup, Central] 4.8-13
LMP/LIH/BV-151-C [Role Switch at Setup, Central] 4.8-13
LMP/LIH/BV-142-C [Reject Role Switch Request, Central] 4.8-16
4.8.4 Role Switch – Both Central and Peripheral 4.8-17
LMP/LIH/BV-143-C [Rejected Role Switch Request, Peripheral] 4.8-17
LMP/LIH/BV-149-C [Rejected Role Switch Request, Central] 4.8-17
LMP/LIH/BV-03-C [Unsupported Role Switch] 4.8-19
4.8.5 Detach – Both Central and Peripheral 4.8-20
LMP/LIH/BV-04-C [Close Link on Request] 4.8-20
LMP/LIH/BV-05-C [Close Link, HCI Command] 4.8-21
LMP/LIH/BV-82-C [Setup Rejected] 4.8-22
4.8.6 Hold mode – Peripheral 4.8-24
LMP/LIH/BV-06-C [Hold Mode, Peripheral] 4.8-24
LMP/LIH/BV-09-C [Hold Mode Request, Peripheral] 4.8-26
4.8.7 Hold mode – Central 4.8-29
LMP/LIH/BV-10-C [Hold Mode, Central] 4.8-29
LMP/LIH/BV-11-C [Hold Mode Request, Central] 4.8-32
4.8.8 Hold mode – Both Central and Peripheral 4.8-34
LMP/LIH/BV-12-C [Hold Mode Unsupported] 4.8-34
4.8.9 Sniff mode – Peripheral 4.8-35
LMP/LIH/BV-14-C [Enter Sniff Mode] 4.8-35
LMP/LIH/BV-15-C [Initiate Sniff Mode, Peripheral] 4.8-38
LMP/LIH/BV-16-C [Exit Sniff Mode] 4.8-40

LMP/LIH/BV-17-C [Accept Sniff Reject] 4.8-41
4.8.10 Sniff mode – Central 4.8-43
LMP/LIH/BV-18-C [Initiate Sniff Mode, Central] 4.8-43
LMP/LIH/BV-19-C [Request Sniff Mode Exit] 4.8-47
4.8.11 Sniff mode – Both Central and Peripheral 4.8-49
LMP/LIH/BV-20-C [Sniff Mode Reject] 4.8-49
4.8.12 Power control – Both Central and Peripheral 4.8-49
LMP/LIH/BV-35-C [Lowest Power Report] 4.8-49
LMP/LIH/BV-36-C [Highest Power Report] 4.8-51
LMP/LIH/BV-76-C [Request Decreased Power] 4.8-52
LMP/LIH/BV-77-C [Request Increased Power] 4.8-53
LMP/LIH/BV-127-C [Respond to EPC Increment Request] 4.8-55
LMP/LIH/BV-128-C [Respond to EPC Decrement Request] 4.8-56
LMP/LIH/BV-129-C [Respond to EPC go to Maximum Power Level] 4.8-58
LMP/LIH/BV-130-C [Request an EPC Increment] 4.8-60
LMP/LIH/BV-131-C [Request an EPC Decrement] 4.8-61
LMP/LIH/BV-133-C [Power Response Reports Unsupported Modulation Correctly] 4.8-63
LMP/LIH/BV-152-C [Power Control Request – Not Supported] 4.8-64
4.8.13 Quality of Service (QoS) – Peripheral 4.8-65
LMP/LIH/BV-39-C [Accept Polling Interval Notification] 4.8-65
LMP/LIH/BV-40-C [Accept Polling Interval Request] 4.8-66
LMP/LIH/BV-41-C [Polling Interval Rejected] 4.8-67
4.8.14 Quality of Service (QoS) – Central 4.8-69
LMP/LIH/BV-42-C [Set Polling Interval] 4.8-69
4.8.15 SCO links – Peripheral 4.8-70
LMP/LIH/BV-43-C 4.8-71
LMP/LIH/BV-44-C 4.8-71
LMP/LIH/BV-45-C 4.8-71
LMP/LIH/BV-46-C 4.8-71
LMP/LIH/BV-47-C [Accept Change to HV2 as Peripheral] 4.8-73
LMP/LIH/BV-48-C [Accept Change to HV3] 4.8-73
LMP/LIH/BV-49-C [Request Change to HV2] 4.8-74
LMP/LIH/BV-50-C [HV2 Request Rejected by Central] 4.8-75
4.8.16 SCO links – Central 4.8-77
LMP/LIH/BV-53-C 4.8-77
LMP/LIH/BV-54-C 4.8-77
LMP/LIH/BV-55-C [Request change to HV3] 4.8-79
LMP/LIH/BV-56-C [HV3 request rejected by Peripheral] 4.8-80
LMP/LIH/BV-57-C [Accept change to HV2 as Central] 4.8-81
4.8.17 SCO links – Both Central and Peripheral 4.8-84
LMP/LIH/BV-60-C [Reject SCO Request] 4.8-84
LMP/LIH/BI-01-C [Rejecting SCO Connection request when AES-CCM encryption is enabled, Peripheral] 4.8-85
LMP/LIH/BI-02-C [Rejecting SCO Connection request when AES-CCM encryption is enabled, Central] 4.8-85
LMP/LIH/BV-51-C [Accept SCO Closure as Peripheral] 4.8-86
LMP/LIH/BV-59-C [Accept SCO Closure as Central] 4.8-86
LMP/LIH/BV-52-C [Request SCO Closure as Peripheral] 4.8-87
LMP/LIH/BV-58-C [Request SCO Closure as Central] 4.8-87
4.8.18 eSCO links – Peripheral 4.8-87
LMP/LIH/BV-100-C [Accept EV3 eSCO request] 4.8-88
LMP/LIH/BV-101-C [Accept EV4 eSCO request] 4.8-88
LMP/LIH/BV-102-C [Accept EV5 eSCO request] 4.8-88
LMP/LIH/BV-103-C [Request eSCO as Peripheral] 4.8-90
LMP/LIH/BV-104-C [Accept change from EV3 to higher length packet, EV4] 4.8-92
LMP/LIH/BV-105-C [Accept change from EV3 to higher length packet, EV5] 4.8-92

LMP/LIH/BV-106-C [Request change to EV4] 4.8-94
LMP/LIH/BV-107-C [EV4 request rejected] 4.8-95
LMP/LIH/BV-108-C [Accept eSCO Closure as Peripheral] 4.8-97
LMP/LIH/BV-109-C [Request eSCO Closure as Peripheral] 4.8-97
4.8.19 eSCO links – Central 4.8-98
LMP/LIH/BV-110-C [Request eSCO as Central] 4.8-98
LMP/LIH/BV-111-C [Accept eSCO request] 4.8-100
LMP/LIH/BV-112-C [Request eSCO change] 4.8-102
LMP/LIH/BV-113-C [eSCO change rejected] 4.8-104
LMP/LIH/BV-114-C [Request to close eSCO link] 4.8-105
LMP/LIH/BV-115-C [Accept eSCO closure as Central] 4.8-106
LMP/LIH/BV-116-C [Reject eSCO Request] 4.8-107
4.8.20 Sniff Subrating 4.8-108
LMP/LIH/BV-117-C [LMP Feature Bits] 4.8-108
LMP/LIH/BV-118-C [Entering Sniff Subrating Mode from Sniff Mode with Lower Tester as the Initiator] 4.8-109
LMP/LIH/BV-119-C [Entering Sniff Subrating Mode From Sniff Mode With IUT As The Initiator] 4.8-111
LMP/LIH/BV-120-C [IUT Rejects Sniff Subrating Request When In Active Mode] 4.8-113
LMP/LIH/BV-121-C [Entering Sniff Subrating Mode With Lower Tester Initiating Sniff Mode Request] 4.8-114
LMP/LIH/BV-122-C [Entering Sniff Subrating Mode with IUT Initiating Sniff Mode Request] 4.8-119
LMP/LIH/BV-123-C [IUTs Transitioning from Existing Sniff Subrating Mode to A New Set of Subrating Parameters] 4.8-124
LMP/LIH/BV-124-C [Sniff Subrating Mode to Active Mode Transition Initiated By Lower Tester] 4.8-126
LMP/LIH/BV-125-C [Sniff Subrating Mode to Active Mode Transition Initiated By IUT] 4.8-128
4.8.21 Multi-slot Packets – Peripheral 4.8-129
LMP/LIH/BV-63-C [Accept Maximum Slot Request] 4.8-129
LMP/LIH/BV-146-C [Maximum Slot after Role Switch as Peripheral] 4.8-130
4.8.22 Multi-slot Packets – Central 4.8-132
LMP/LIH/BV-148-C [Maximum Slot after Role Switch as Central] 4.8-132
4.8.23 Multi-slot Packets – Both Central and Peripheral 4.8-134
LMP/LIH/BV-61-C [Request Maximum Slots as Peripheral] 4.8-134
LMP/LIH/BV-64-C [Request Maximum Slots as Central] 4.8-134
LMP/LIH/BV-145-C [Maximum Slot after a Connection as Peripheral] 4.8-137
LMP/LIH/BV-147-C [Maximum Slot after a Connection as Central] 4.8-137
4.8.24 Paging_Scheme – Both Central and Peripheral 4.8-138
LMP/LIH/BV-71-C [Reject Page Mode Negotiation] 4.8-138
LMP/LIH/BV-72-C [Reject Page Scan Negotiation] 4.8-138
4.8.25 Link Supervision 4.8-139
LMP/LIH/BV-74-C [Set Supervision Timer as Central] 4.8-139
LMP/LIH/BV-126-C [Set Supervision Timer as Peripheral] 4.8-140
4.8.26 Deadlock Avoidance 4.8-142
LMP/LIH/BV-80-C [Avoid Deadlock as Central] 4.8-142
LMP/LIH/BV-81-C [Avoid Deadlock as Peripheral] 4.8-143
4.8.27 Test for Devices that do not Support Enhanced Data Rate 4.8-144
LMP/LIH/BV-83-C [Test for Devices that do not support Enhanced Data_Rate] 4.8-144
4.8.28 Setting up and Removing Enhanced Data_Rate ACL connection 4.8-145
LMP/LIH/BV-84-C [EDR ACL Link Setup] 4.8-146
LMP/LIH/BV-85-C [EDR ACL Link Remove] 4.8-147
4.8.29 Setting an Enhanced Data Rate eSCO Connection 4.8-148
LMP/LIH/BV-86-C [EDR 2-EV3 eSCO Link Setup] 4.8-148
LMP/LIH/BV-87-C [EDR eSCO Link Remove] 4.8-150
LMP/LIH/BV-150-C [APB Ignores PDUs Other Than Clock Adjustment] 4.8-151
4.9 Test Modes 4.9-1
4.9.1 References 4.9-1
4.9.2 Enabled Mode – Peripheral 4.9-1
LMP/TEM/BV-01-C [Reject Test Mode Request] 4.9-1

4.10 Adaptive Frequency Hopping 4.10-1
4.10.1 References 4.10-1
4.10.2 Adaptive Frequency Hopping test cases 4.10-1
LMP/AFH/BV-01-C [AFH Enable – Peripheral] 4.10-1
LMP/AFH/BV-02-C [AFH Disable - Peripheral] 4.10-2
LMP/AFH/BV-03-C [AFH Switch – Peripheral] 4.10-3
LMP/AFH/BV-04-C [Classification Reporting – Normal Operation] 4.10-4
LMP/AFH/BV-05-C [Classification Reporting – After Successful Role Switch] 4.10-6
LMP/AFH/BV-06-C [Classification Reporting – After Unsuccessful Role Switch] 4.10-8
LMP/AFH/BV-08-C [Classification Reporting – After Unhold] 4.10-10
LMP/AFH/BV-09-C [Peripheral device does not send an LMP_SET_AFH PDU – after successful role switch] 4.10-12
4.11 Simple Pairing procedures 4.11-1
4.11.1 References 4.11-1
4.11.2 Backward Compatibility procedures 4.11-1
LMP/SP/BV-01-C [Secure Simple Pairing Capable Controller - Pairing - IUT Initiator] 4.11-1
LMP/SP/BV-02-C [Secure Simple Pairing Capable Controller - Pairing - IUT Responder] 4.11-2
LMP/SP/BV-03-C [Secure Simple Pairing Capable Controller - Legacy Host- IUT Initiator] 4.11-3
LMP/SP/BV-04-C [Secure Simple Pairing Capable Controller - Legacy Host - IUT Responder] 4.11-4
LMP/SP/BV-05-C [Secure Simple Pairing Capable Controller - Legacy Remote Host - IUT Initiator] 4.11-5
LMP/SP/BV-37-C [Secure Connections Capable Controller – Legacy Host – IUT Initiator – Unauthenticated Link Key] 4.11-8
LMP/SP/BV-38-C [Secure Connections Capable Controller – Legacy Host – IUT Initiator – Authenticated Link Key] 4.11-8
LMP/SP/BV-39-C [Secure Connections Capable Controller – Host has no P256 OOB data available – IUT Initiator – OOB] 4.11-14
LMP/SP/BV-40-C [Secure Connections Capable Controller and Host – Legacy Remote Controller – IUT Initiator] 4.11-20
4.11.3 Numeric Comparison procedures 4.11-21
LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success] 4.11-21
LMP/SP/BV-07-C [Numeric Comparison - IUT Responder – Success] 4.11-26
LMP/SP/BV-08-C [Numeric Comparison - IUT Initiator - Failure on Initiating Side] 4.11-31
LMP/SP/BV-43-C [Numeric Comparison – IUT Initiator, Failure on Initiating side, P-256] 4.11-31
LMP/SP/BV-09-C [Numeric Comparison - IUT Responder - Failure on Initiating Side] 4.11-35
LMP/SP/BV-44-C [Numeric Comparison – IUT Responder – Failure on Initiating Side, P-256] 4.11-35
LMP/SP/BV-10-C [Numeric Comparison - IUT Initiator - Failure on Responding Side] 4.11-38
LMP/SP/BV-45-C [Numeric Comparison – IUT Initiator – Failure on Responding Side, P-256] 4.11-38
LMP/SP/BV-11-C [Numeric Comparison - IUT Responder - Failure on Responding Side] 4.11-42
LMP/SP/BV-46-C [Numeric Comparison – IUT Responder – Failure on Responding Side, P-256] 4.11-42
LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256] 4.11-45
LMP/SP/BV-42-C [Numeric Comparison – IUT Responder – Success, P-256] 4.11-51
LMP/SP/BV-47-C [Pairing on encrypted ACL – Numeric Comparison – IUT Initiator – Success, P-256] 4.11-56
4.11.4 Passkey Entry procedures 4.11-58
LMP/SP/BV-12-C [Passkey Entry - IUT Initiator – Success] 4.11-58
LMP/SP/BV-13-C [Passkey Entry - IUT Responder – Success] 4.11-63
LMP/SP/BV-14-C [Passkey Entry - IUT Initiator - Failure on Initiating Side] 4.11-68
LMP/SP/BV-50-C [Passkey Entry – IUT Initiator – Failure on Initiating Side, P-256] 4.11-68
LMP/SP/BV-15-C [Passkey Entry - IUT Responder - Failure on Initiating Side] 4.11-71
LMP/SP/BV-51-C [Passkey Entry – IUT Responder – Failure on Initiating Side, P-256] 4.11-71
LMP/SP/BV-16-C [Passkey Entry - IUT Initiator - Failure on Responding Side] 4.11-74
LMP/SP/BV-52-C [Passkey Entry – IUT Initiator – Failure on Responding Side, P-256] 4.11-74
LMP/SP/BV-17-C [Passkey Entry - IUT Responder - Failure on Responding Side] 4.11-77
LMP/SP/BV-53-C [Passkey Entry – IUT Responder – Failure on Responding Side, P-256] 4.11-77
LMP/SP/BV-48-C [Passkey Entry – IUT Initiator – Success, P-256] 4.11-80
LMP/SP/BV-49-C [Passkey Entry – IUT Responder – Success, P-256] 4.11-86

LMP/SP/BV-67-C [Passkey Entry – IUT Initiator, Verify Random Passkeys] 4.11-91
LMP/SP/BV-68-C [Passkey Entry – IUT Responder – Verify Random Passkeys] 4.11-94
4.11.5 Out-of-Band procedures 4.11-97
LMP/SP/BV-18-C [OOB Protocol - IUT Initiator - IUT with OOB_Auth_Data – Success, v6.2 and earlier] ... 4.11-97
LMP/SP/BV-20-C [OOB Protocol - IUT Initiator – Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier] 4.11-97
LMP/SP/BV-22-C [OOB Protocol - IUT Initiator - IUT and Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier] 4.11-97
LMP/SP/BV-70-C [OOB Protocol - IUT Initiator - IUT with OOB_Auth_Data – Success, v6.3 and later] 4.11-97
LMP/SP/BV-71-C [OOB Protocol - IUT Initiator – Lower Tester with OOB_Auth_Data – Success, v6.3 and later] 4.11-97
LMP/SP/BV-72-C [OOB Protocol - IUT Initiator - IUT and Lower Tester with OOB_Auth_Data – Success, v6.3 and later] 4.11-97
LMP/SP/BV-19-C [OOB Protocol - IUT Responder - IUT with OOB_Auth_Data – Success, v6.2 and earlier] 4.11-104
LMP/SP/BV-21-C [OOB Protocol - IUT Responder – Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier] 4.11-104
LMP/SP/BV-23-C [OOB Protocol - IUT Responder - IUT and Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier] 4.11-104
LMP/SP/BV-73-C [OOB Protocol - IUT Responder - IUT with OOB_Auth_Data – Success, v6.3 and later] 4.11-104
LMP/SP/BV-74-C [OOB Protocol - IUT Responder – Lower Tester with OOB_Auth_Data – Success, v6.3 and later] 4.11-104
LMP/SP/BV-75-C [OOB Protocol - IUT Responder - IUT and Lower Tester with OOB_Auth_Data – Success, v6.3 and later] 4.11-104
LMP/SP/BV-24-C [OOB Protocol - IUT Initiator - IUT with OOB_Auth_Data – Failure] 4.11-110
LMP/SP/BV-60-C [OOB Protocol – IUT Initiator - IUT with OOB_Auth_Data – Failure, P-256] 4.11-110
LMP/SP/BV-25-C [OOB Protocol - IUT Responder - IUT with OOB_Auth_Data – Failure] 4.11-114
LMP/SP/BV-61-C [OOB Protocol – IUT Responder - IUT with OOB_Auth_Data – Failure, P-256] 4.11-114
LMP/SP/BV-26-C [OOB Protocol - IUT Initiator – Lower Tester with OOB_Auth_Data – Failure] 4.11-117
LMP/SP/BV-62-C [OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Failure, P-256] .. 4.11-117
LMP/SP/BV-27-C [OOB Protocol - IUT Responder – Lower Tester with OOB_Auth_Data – Failure] 4.11-121
LMP/SP/BV-63-C [OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Failure, P-256] 4.11-121
LMP/SP/BV-54-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Success, P-256, v6.2 |and earlier] 4.11-124
LMP/SP/BV-56-C [OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier] 4.11-124
LMP/SP/BV-58-C [OOB Protocol – IUT Initiator – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier] 4.11-124
LMP/SP/BV-76-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Success, P-256, v6.3 |and later] 4.11-124
LMP/SP/BV-77-C [OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later] 4.11-124
LMP/SP/BV-78-C [OOB Protocol – IUT Initiator – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later] 4.11-124
LMP/SP/BV-55-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Success, P-256, v6.2 and earlier] 4.11-132
LMP/SP/BV-57-C [OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier] 4.11-132
LMP/SP/BV-59-C [OOB Protocol – IUT Responder – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier] 4.11-132
LMP/SP/BV-79-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Success, P-256, v6.3 and later] 4.11-133
LMP/SP/BV-80-C [OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later] 4.11-133

LMP/SP/BV-81-C [OOB Protocol – IUT Responder – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later] 4.11-133
4.11.6 Test Mode procedures 4.11-140
LMP/SP/BV-28-C [Secure Simple Pairing Debug Mode - Fixed Private Key] 4.11-140
LMP/SP/BV-29-C [Secure Simple Pairing Debug Mode - Responding Device Uses Fixed Private Key] 4.11-144
LMP/SP/BV-64-C [Secure Simple Pairing Debug Mode – Fixed Private Key, P256] 4.11-148
LMP/SP/BV-65-C [Secure Simple Pairing Debug Mode – Responding Device Uses Fixed Private Key, P256] 4.11-152
4.11.7 Simple Pairing Failure procedures 4.11-156
LMP/SP/BV-30-C [Secure Simple Pairing Failed - IUT Responder] 4.11-156
LMP/SP/BV-31-C [Secure Simple Pairing Capable Controller–Host rejects Secure Simple Pairing–IUT Initiator] 4.11-158
LMP/SP/BV-32-C [Secure Simple Pairing Capable Controller–Host rejects Secure Simple Pairing–IUT Responder] 4.11-159
LMP/SP/BV-33-C [Passkey Entry with Keypress notification - IUT Initiator - Success] 4.11-161
LMP/SP/BV-34-C [Passkey Entry with Keypress notification - IUT Responder – Success] 4.11-166
LMP/SP/BV-35-C [Passkey Entry with Keypress notification - IUT Initiator - Failure on Responding Side] 4.11-171
LMP/SP/BV-36-C [Passkey Entry with Keypress notification - IUT Responder - Failure on Responding Side] 4.11-174
LMP/SP/BI-01-C [Numeric Comparison – IUT Initiator – Invalid Public Key Failure, P-192] 4.11-177
LMP/SP/BI-07-C [Numeric Comparison – IUT Initiator – Invalid Public Key Failure, P-256] 4.11-177
LMP/SP/BI-02-C [Numeric Comparison – IUT Responder – Invalid Public Key Failure, P-192] 4.11-182
LMP/SP/BI-08-C [Numeric Comparison – IUT Responder – Invalid Public Key Failure, P-256] 4.11-182
LMP/SP/BI-03-C [Passkey Entry - IUT Initiator – Invalid Public Key Failure, P-192] 4.11-186
LMP/SP/BI-09-C [Passkey Entry - IUT Initiator – Invalid Public Key Failure, P-256] 4.11-186
LMP/SP/BI-04-C [Passkey Entry – IUT Responder – Invalid Public Key Failure, P-192] 4.11-191
LMP/SP/BI-10-C [Passkey Entry – IUT Responder – Invalid Public Key Failure, P-256] 4.11-191
LMP/SP/BI-05-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-192] 4.11-195
LMP/SP/BI-11-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-256] 4.11-195
LMP/SP/BI-06-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-192] 4.11-200
LMP/SP/BI-12-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-256] 4.11-200
LMP/SP/BV-66-C [Simple Pairing Capable Controller – Host is in Non-Bondable Mode – IUT Responder] 4.11-204
LMP/SP/BI-13-C [Authentication Stage 2, Invalid DHKey Check, Responder] 4.11-206
LMP/SP/BI-14-C [Authentication Stage 2, Invalid DHKey Check, Initiator] 4.11-206
LMP/SP/BV-69-C [Simple Pairing Capable Controller – Reject Secure Simple Pairing PDUs When SSP is Disabled] 4.11-207
4.12 Piconet Clock Adjust 4.12-1
4.12.1 References 4.12-1
4.12.2 Piconet Clock Adjust test cases 4.12-1
LMP/XCL/BV-01-C [Central Initiates Coarse Clock Adjustment] 4.12-1
LMP/XCL/BV-02-C [Peripheral Coarse Clock Adjustment Request] 4.12-3
LMP/XCL/BV-03-C [Test that Central does not reuse Clk_Adj_ID within LSTO] 4.12-4
LMP/XCL/BV-04-C [Test that a Clock Adjust is ignored on ACL-C] 4.12-6
4.13 Slot Availability Mask 4.13-1
4.13.1 References 4.13-1
4.13.2 Slot Availability Mask test cases 4.13-1
LMP/SAM/BV-01-C [Respond to three SAM instances] 4.13-1
LMP/SAM/BV-02-C [Initiate three SAM instances] 4.13-4
LMP/SAM/BI-03-C [Respond to invalid SAM type 0 submap configuration sequence] 4.13-7

LMP/SAM/BI-04-C [Respond to invalid SAM type 0 submap] 4.13-10
LMP/SAM/BI-05-C [Respond to invalid SAM index] 4.13-13
LMP/SAM/BV-06-C [SAM is disabled after a successful role switch] 4.13-16
LMP/SAM/BV-07-C [SAM is resumed after a failed role switch] 4.13-18
LMP/SAM/BV-08-C [SAM and sniff mode] 4.13-20
LMP/SAM/BV-09-C [Respond to request for SAM Anchor Point using Initialization Procedure 2] 4.13-23
LMP/SAM/BV-10-C [Initiate SAM Anchor Point using Initialization Procedure 2] 4.13-25

5 Test case mapping 5-1

5.1 References 5-1

5.2 Test Case Mapping Table 5-2

6 Revision history and acknowledgments 6-1

1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and test cases to test the implementation of the Bluetooth Link Manager Protocol layer with the objective to provide a high probability of air interface interoperability between the tested implementation and other manufacturers' Bluetooth devices.

2 References, definitions, and abbreviations

2.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

2.2 Definitions

In this Bluetooth document, the definitions from [1] and [5] apply.

Certain terms that were identified as inappropriate have been replaced. For a list of the original terms and their replacement terms, see the Appropriate Language Mapping Tables document [15].

2.3 Acronyms and abbreviations

In this Bluetooth document, the definitions, acronyms, and abbreviations from [1] and [5] apply.

Acronyms and abbreviations	Definition
SSR	Sniff Subrating

Table 2.1: Acronyms and abbreviations

The following tables contain parameter name and/or abbreviation changes (or lack thereof) as found in Erratum 14646 of the LMP Specification.

Previous name	Current name
access scheme	Access_Scheme
AFH_channel_classification	AFH_Channel_Classification
AFH_channel_map	AFH_Channel_Map
AFH_instant	AFH_Instant
AFH_max_interval	AFH_Max_Interval
AFH_min_interval	AFH_Min_Interval
AFH_mode	AFH_Mode
AFH_reporting_mode	AFH_Reporting_Mode
air mode	Air_Mode
authentication response	Authentication_Response
clk_adj_clk	Clk_Adj_Clk
clk_adj_id	Clk_Adj_ID
clk_adj_instant	Clk_Adj_Instant
clk_adj_mode	Clk_Adj_Mode
clk_adj_period	Clk_Adj_Period
clk_adj_slots	Clk_Adj_Slots
clk_adj_us	Clk_Adj_Offset
clock offset	Clock_Offset
Commitment value	Commitment_Value
CompId	Company_Identifier
Confirmation value	Confirmation_Value
data rate	Data_Rate
drift	Drift
D_{sniff}	D_{Sniff}
encapsulated data	Encap_Data
encapsulated major type	Encap_Major_Type
encapsulated minor type	Encap_Minor_Type
encapsulated payload length	Encap_Payload_Length
encryption mode	Encryption_Mode
error code	Error_Code
escape op code	Escape_Opcode
eSCO handle	eSCO_Handle
eSCO LT_ADDR	eSCO_LT_ADDR
eSCO packet type	eSCO_Packet_Type
extended features	Extended_Features
extended op code	Extended_Opcode
features	Features
features page	Features_Page
hold instant	Hold_Instant

Previous name	Current name
hold time	Hold_Time
jitter	Jitter
key	Key
key size	Key_Size
key size mask	Key_Size_Mask
max slots	Max_Slots
max supported page	Max_Supported_Page
max_sniff_subrate	Max_Sniff_Subrate
min_sniff_mode_timeout	Min_Sniff_Mode_Timeout
name fragment	Name_Fragment
name length	Name_Length
name offset	Name_Offset
negotiation state	Negotiation_State
Nonce Value	Nonce_Value
Notification Type	Notification_Type
N_{poll}	N_{Poll}
N_{SAM-SM}	N_{SAM_SM}
OOB Authentication Data	OOB_Auth_Data
op code	Opcode
packet length	Packet_Length
packet type table	Packet_Type_Table
paging scheme	Paging_Scheme
paging scheme settings	Paging_Scheme_Settings
poll interval	Poll_Interval
power_adjustment_request	Power_Adj_Request
power_adjustment_response	Power_Adj_Response
random number	Random_Number
SAM_Submaps	SAM_Submaps
SAM_Type0-Submap	SAM_Type0_Submap
SCO handle	SCO_Handle
SCO packet	SCO_Packet
slot offset	Slot_Offset
sniff attempt	Sniff_Attempt
sniff timeout	Sniff_Timeout
sniff_subrating_instant	Sniff_Subrating_Instant
SubVersNr	Subversion
supervision timeout	Supervision_Timeout
switch instant	Switch_Instant
timing control flags	Timing_Control_Flags
T_{sniff}	T_{Sniff}

Previous name	Current name
Update Mode	Update_Mode
VersNr	Version

Table 2.2: Parameter names changed under Erratum 14646 of the LMP Specification

Unchanged parameter names
Authentication_Requirements
BD_ADDR
D_{eSCO}
D_{SAM}
D_{SCO}
IO_Capabilities
LT_ADDR
SAM_Index
SAM_Instant
T_{eSCO}
$T_{SAM\text{-}SM}$
T_{SCO}
W_{eSCO}

Table 2.3: Parameter names unchanged by Erratum 14646 of the LMP Specification

3 Test Suite Structure (TSS)

3.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

3.2 Overview

The Link Manager is layer 3 of the Bluetooth protocol stack.

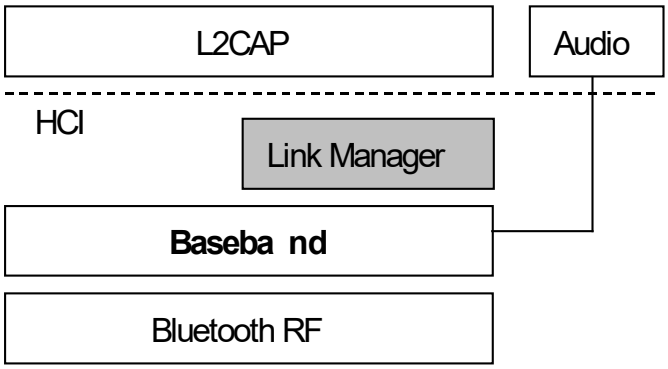


Figure 3.1: Bluetooth protocol stack, basic layers

The Link Manager specifies seven groups of services:

- Authentication Procedures
- Encryption
- Information Requests
- Link Handling
- Test Mode
- Adaptive Frequency Hopping
- Secure Simple Pairing

Figure 3.2 shows the Link Manager Test Suite Structure (TSS) including its subgroups for the conformance testing.

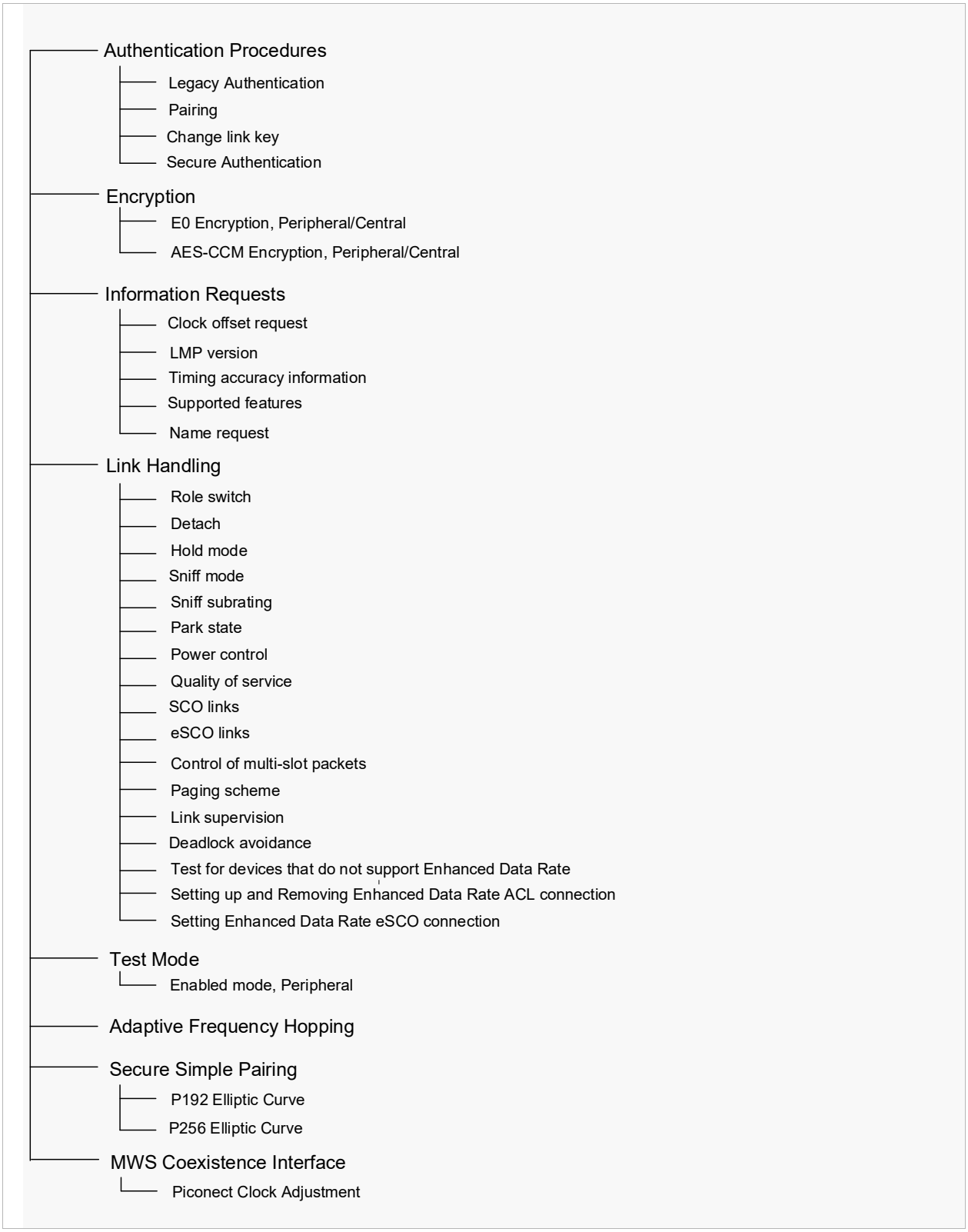


Figure 3.2: TSS for Link Manager

3.3 Test Suite Structure (TSS)

The Test Suite Structure is structured as a tree with a first level defined as LM representing the protocol groups: Authentication Procedures, Encryption, Information Requests, Link Handling, Test Mode, Adaptive Frequency Hopping, Secure Simple Pairing, and MWS Coexistence Interface.

3.3.1 Test groups

The test groups are organized in three levels. The first level defines the protocol groups representing the protocol services. The second level separates the protocol services in functional modules. The last level in each branch contains the standard ISO subgroups BV and BI.

The main test groups are the capability group, the valid behavior group, and the invalid behavior group.

3.3.2 Protocol groups

The protocol groups identify the Bluetooth Link Manager services: Authentication Procedures, Encryption, Information Requests, Link Handling, Test Mode, Adaptive Frequency Hopping, Secure Simple Pairing, and Piconet Clock Adjustment as defined in [1].

3.3.2.1 Authentication procedures

The authentication procedures module covers the whole authentication procedure for two devices.

3.3.2.2 Encryption

The encryption module covers the optional encryption procedure so that two devices can use encrypted traffic.

3.3.2.3 Information Requests

The information requests module covers the information procedure between two devices.

3.3.2.4 Link Handling

The link handling module covers the link handling procedures such as Enhanced Data_Rate link setup.

3.3.2.5 Test Mode

The test mode module verifies that a Central cannot set a Peripheral into test mode unless it is locally enabled.

3.3.2.6 Adaptive Frequency Hopping

The Adaptive Frequency Hopping (AFH) module covers adaptive frequency hopping control functions.

3.3.2.7 Secure Simple Pairing

The Secure Simple Pairing module covers simple pairing functions.

3.3.2.8 MWS Coexistence

The MWS Coexistence module verifies that the Central of a piconet can adjust the piconet clock.

3.3.2.9 Slot Availability Mask

3.3.3 Behavior testing groups

The TSS accommodates both valid and invalid behaviors.

3.3.3.1 Valid Behavior (BV) tests

This subgroup provides testing to verify that the IUT reacts in conformity with the Bluetooth standard, after receipt or exchange of valid Protocol Data Units (PDUs). Valid PDUs means that the exchange of messages and the content of the exchanged messages are considered as valid.

3.3.3.2 Invalid Behavior (BI) tests

This subgroup provides testing to verify that the IUT reacts in conformity with the Bluetooth standard, after receipt of a syntactically or semantically invalid PDU.

3.3.4 HCI Command and Event Version

If a command or event has more than one version and the test does not explicitly say otherwise:

- A reference to a command specifying the version number means that that version or any higher-numbered version supported by the IUT may be used.
- A reference to an event specifying the version number means that that version or at least one higher-numbered version supported by the IUT is unmasked (other versions, including lower-numbered versions, may also be unmasked).
- A reference to a command or event that does not specify the version number is equivalent to specifying [v1].

4 Test cases (TC)

4.1 Introduction

4.1.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.1.2 Test case identification conventions

Test cases are assigned unique identifiers per the conventions in [5]. The convention used here is: **<spec abbreviation>/<IUT role>/**<class>/**<feat>**/<func>/<subfunc>/<cap>/**<xx>-<nn>-<y>**.

Identifier Abbreviation	**Spec Identifier <spec abbreviation>**
LMP	Link Manager Protocol
Identifier Abbreviation	**Feature Identifier <feat>**
AFH	Adaptive Frequency Hopping
AUT	Authentication Procedure
ENC	Encryption
INF	Information Requests
LIH	Link Handling
SAM	Slot Availability Mask
SP	Secure Simple Pairing
TEM	Test Mode
XCL	Coexistence Piconet Clock Adjustment

Table 4.1-1: LMP TC feature naming conventions

4.1.3 Conformance

When conformance is claimed for a particular specification, all capabilities are to be supported in the specified manner. The mandated tests from this Test Suite depend on the capabilities to which conformance is claimed.

The Bluetooth Qualification Program may employ tests to verify implementation robustness. The level of implementation robustness that is verified varies from one specification to another and may be revised for cause based on interoperability issues found in the market.

Such tests may verify:

- That claimed capabilities may be used in any order and any number of repetitions not excluded by the specification
- That capabilities enabled by the implementations are sustained over durations expected by the use case
- That the implementation gracefully handles any quantity of data expected by the use case
- That in cases where more than one valid interpretation of the specification exists, the implementation complies with at least one interpretation and gracefully handles other interpretations
- That the implementation is immune to attempted security exploits

A single execution of each of the required tests is required to constitute a Pass verdict. However, it is noted that to provide a foundation for interoperability, it is necessary that a qualified implementation consistently and repeatedly pass any of the applicable tests.

In any case, where a member finds an issue with the test plan generated by the Bluetooth SIG qualification tool, with the test case as described in the Test Suite, or with the test system utilized, the member is required to notify the responsible party via an erratum request such that the issue may be addressed.

4.1.4 Baseband assumptions

Subsections Test cases are built upon having a Baseband Link up and running. The IUT and the Lower Tester must be in connection state (Active mode). DM1 packets are used where not otherwise specified. See the Preambles section.

All test cases are built upon a connection between two (2) devices a Central and a Peripheral.

4.1.5 Role Switch

To force the IUT to become Central of the piconet, Paging of the Lower Tester must be used as PDU LMP_SWITCH_REQ is optional and all IUTs will not support this. See the Preambles section.

4.1.6 Applicable Parameter Values

The parameter values indicated in the test cases are thought to be reasonable. However, what is reasonable ultimately depends on the user scenario the IUT is intended for. In those cases, where the Bluetooth System Specification does not require the implementation of a specific value, and the IUT cannot support the value indicated in a test case, it is allowed to test the IUT with another value. The selected value has to be given as IXIT information. When a value deviates from what is indicated in the test case, select as close as possible to the value indicated in the test case. The selected value must not be such that the test purpose for the test case cannot be verified or the test case is not applicable. All test cases applicable as determined by the combination of Test Case Reference List, Implementation Conformance Statement, and Test Case Mapping Table must be executed successfully to complete the qualification of the IUT.

4.1.7 Advertisement of Features for test cases

It is favorable to avoid LMP traffic that could create situations in which a test case is not designed to be executed or which may add complexity to the test system implementation. This can be achieved by proper selection of which Features are advertised by the Lower Tester. In some test cases this is exactly specified in the Test Suite but in most cases it is not. As a general rule, for each test case the Lower Tester should not advertise more Features than necessary to facilitate execution of the test purpose. Specifically, with the introduction of Enhanced Data_Rate, this feature is only advertised by the Lower Tester in those test cases where it is necessary for the test purpose.

4.2 Default settings

The default settings must be carried out before each test case to guarantee a correct set up each time the tests are performed. Please see Default settings for the set up messages used.

4.2.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.2.2 Authentication

This default setting will be used for the different authentication test cases.

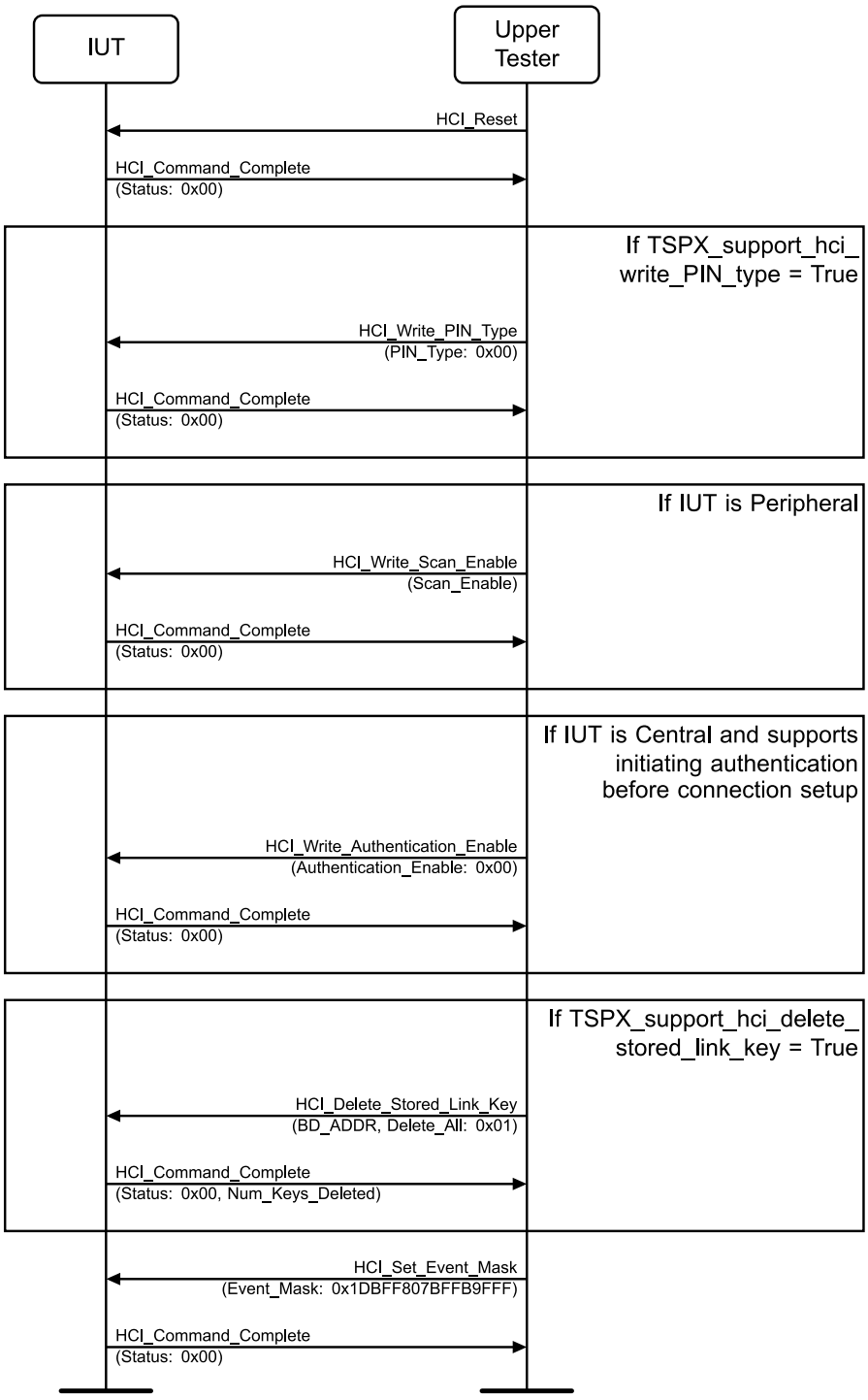


Figure 4.2-1: Default settings used for authentication test cases

4.2.3 Encryption

This default setting will be used for the different encryption test cases.

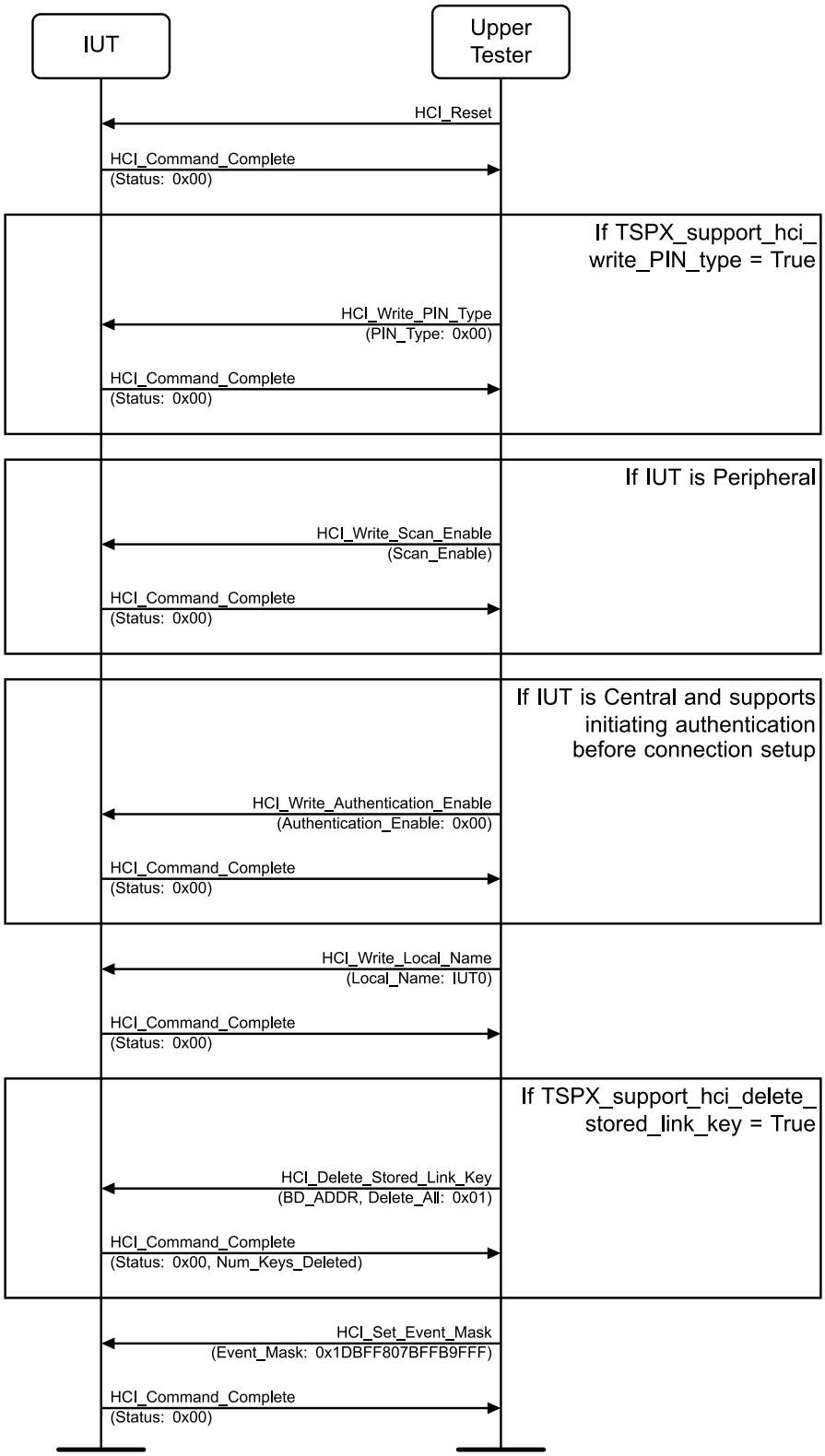


Figure 4.2-2: Default settings used for encryption test cases

4.2.4 Information Requests

This default setting will be used for the different information requests test cases.

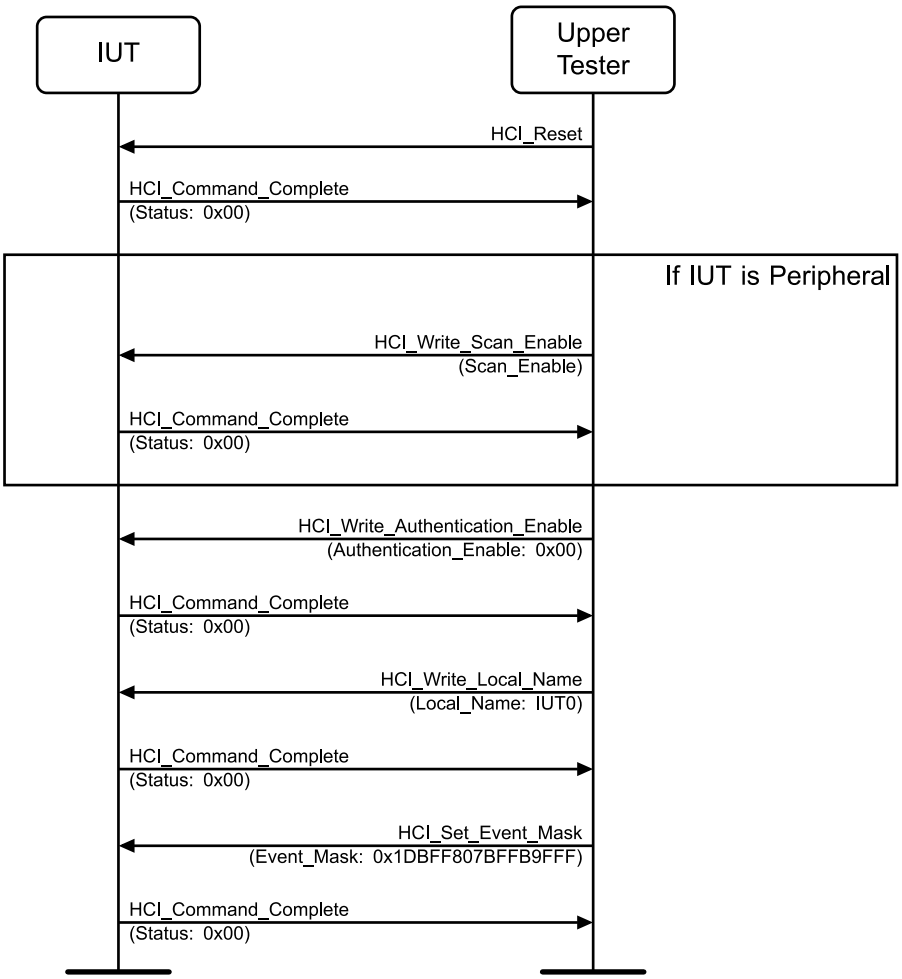


Figure 4.2-3: Default settings used for information requests test cases

4.2.5 Link Handling

This default setting will be used for the different link handling test cases.

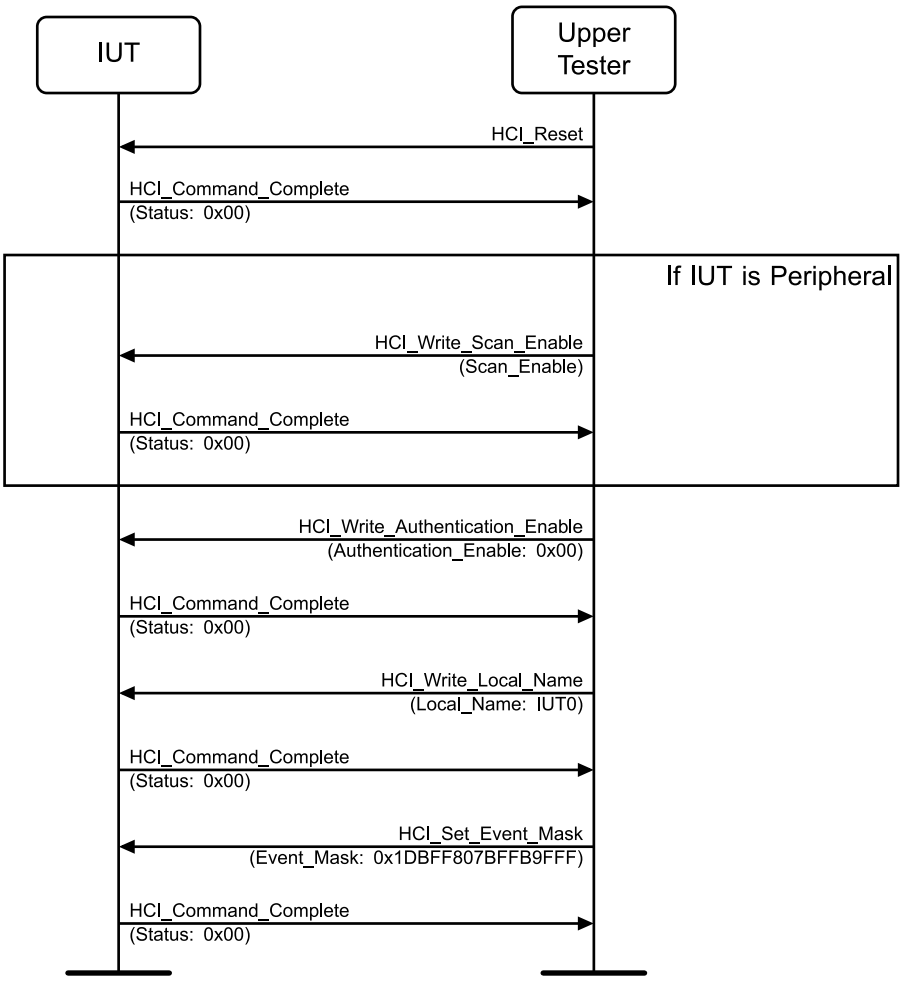


Figure 4.2-4: Default settings used for link handling test cases

4.2.6 Secure Simple Pairing

The default settings used for the Secure Simple Pairing test cases.

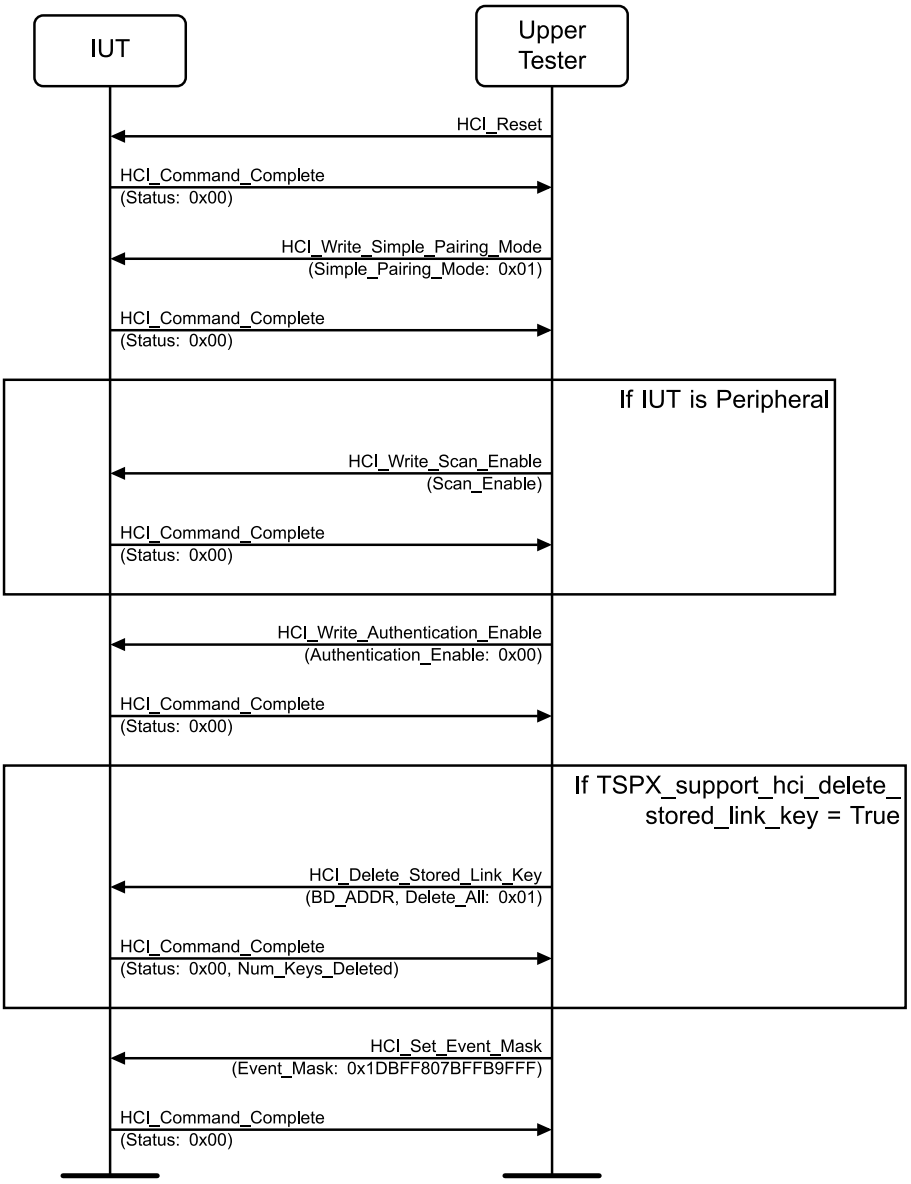


Figure 4.2-5: Default setting used for Secure Simple Pairing test cases

4.2.7 AES-CCM Encryption

The default settings used for the AES-CCM encryption test cases.

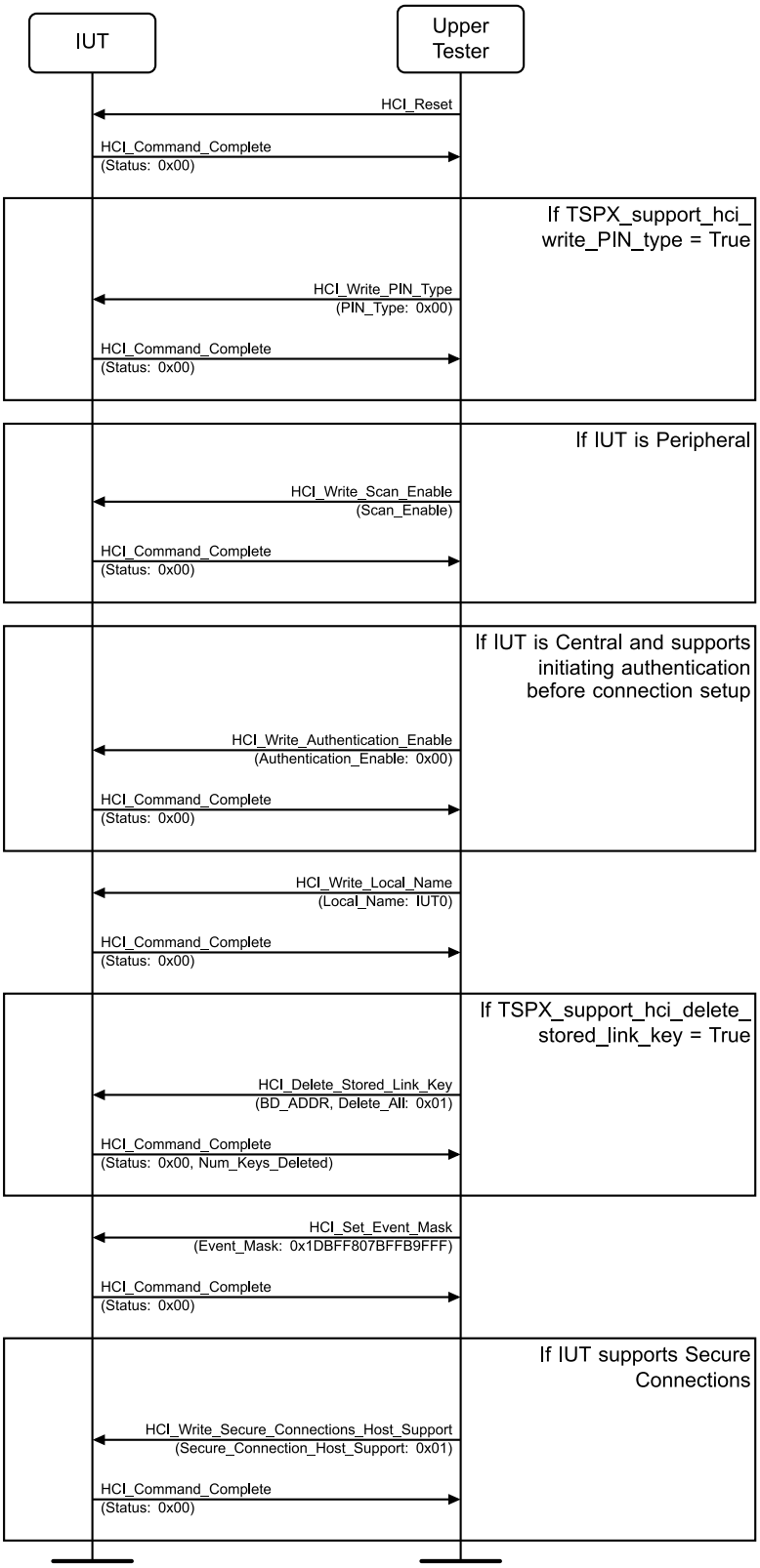


Figure 4.2-6: Default setting used for AES-CCM encryption test cases

4.2.8 Secure Simple Pairing P-256

The default settings used for the Secure Simple Pairing test cases using the P-256 Elliptic Curve.

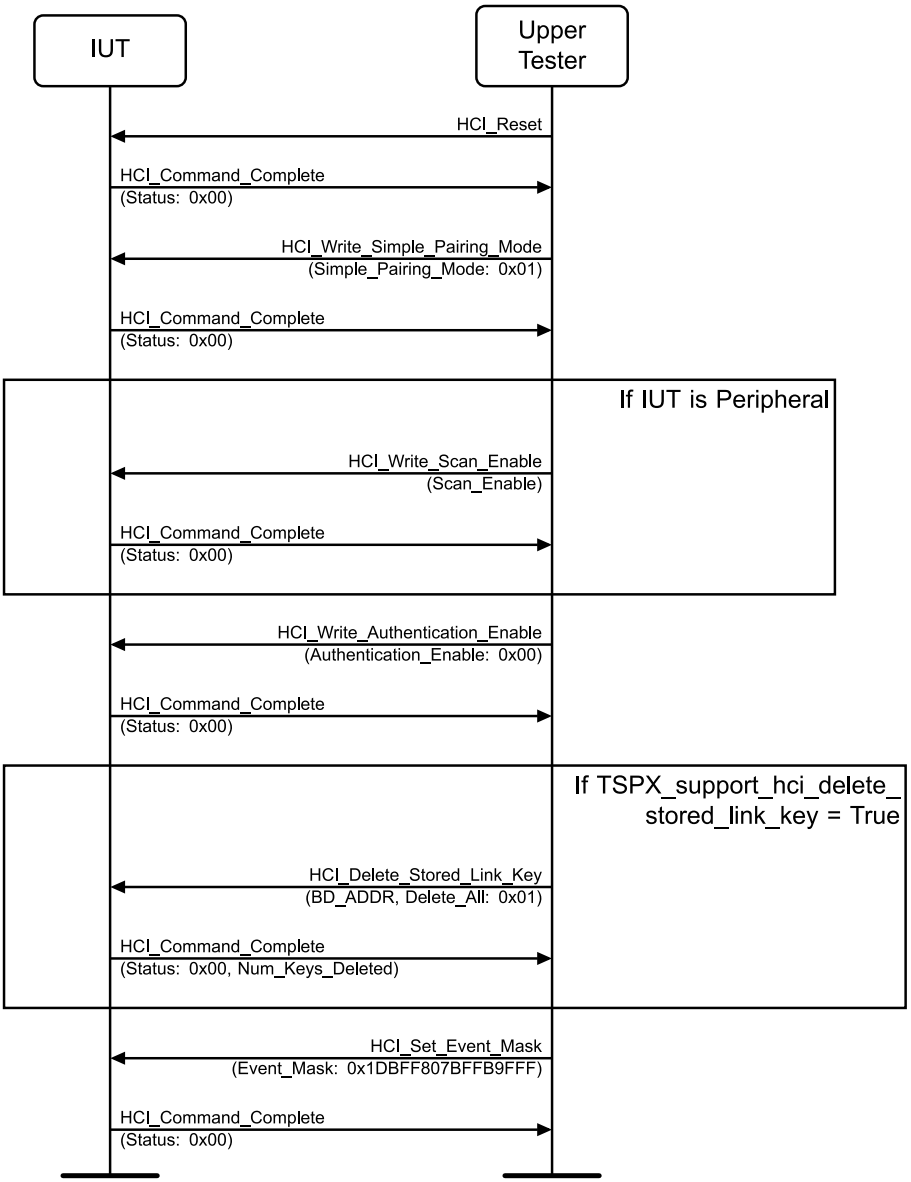


Figure 4.2-7: Default setting used for the Secure Simple Pairing test cases using the P-256 Elliptic Curve

4.3 Preambles

The MSCs in this section are provided for information, as they are used by test equipment in achieving the initial conditions in certain tests.

4.3.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.3.2 Connection Establishment IUT Central

This Preamble will be used when the IUT will act as Central.

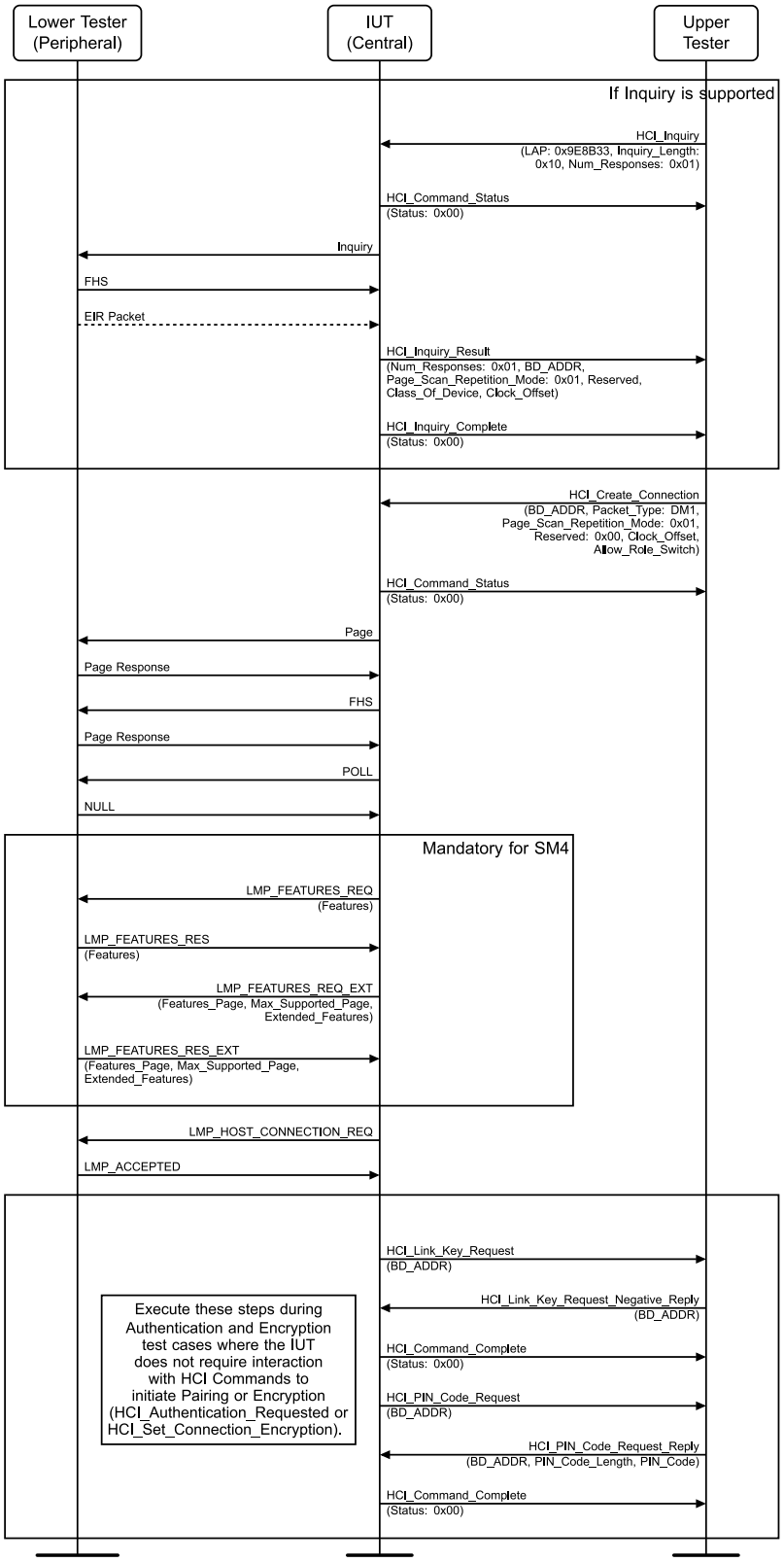


Figure 4.3-1: Preamble used when the IUT will act as Central

4.3.3 Connection Establishment Lower Tester

This Preamble will be used in all cases when the IUT will act as a Peripheral.

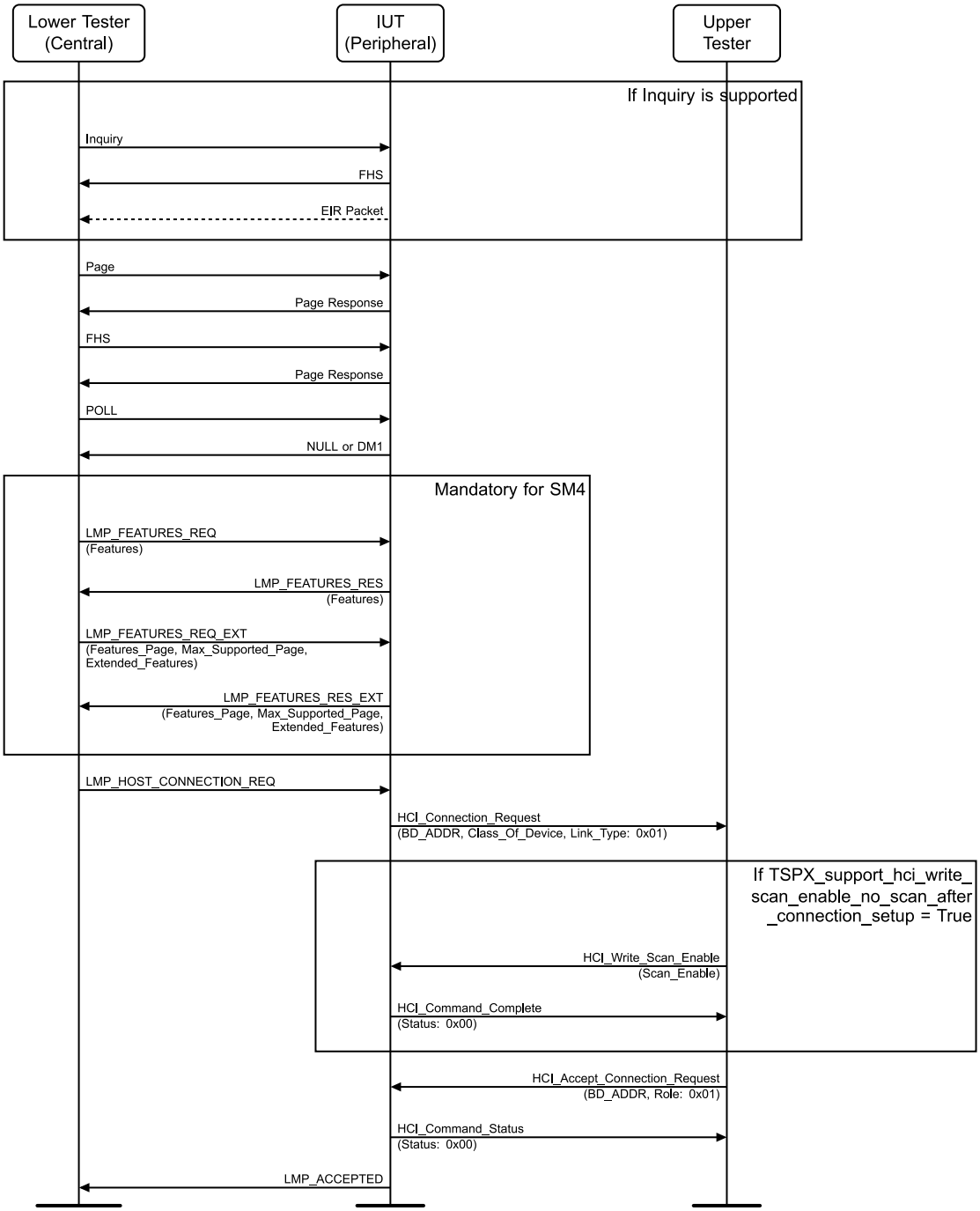


Figure 4.3-2: Preamble used when the IUT will act as Peripheral

4.3.4 Default settings

Connection setup with the Enhanced Data_Rate ACL link enabled.

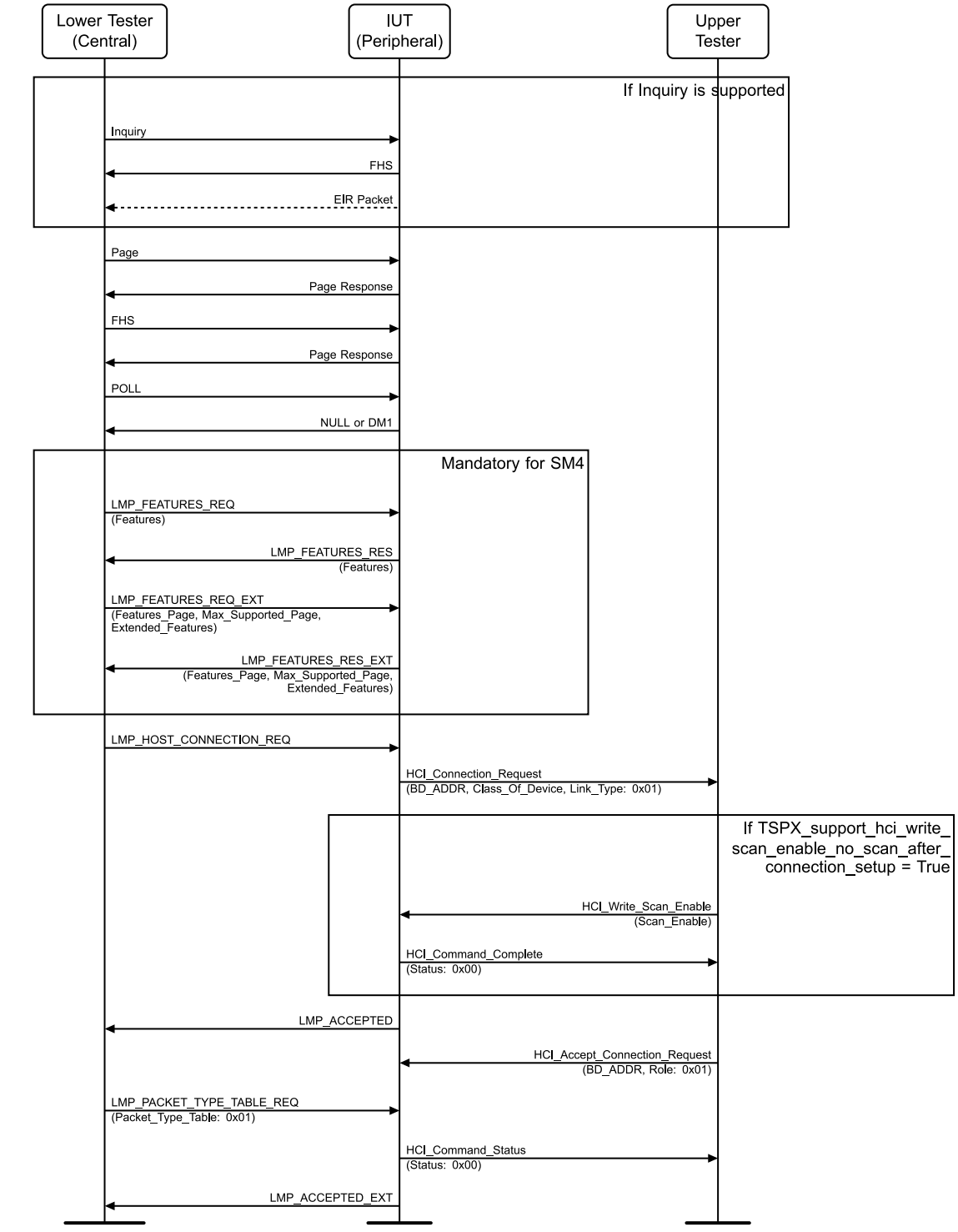


Figure 4.3-3: Connection setup with the Enhanced Data_Rate established

4.3.5 External Frame Configuration

This preamble will be used for external frame configuration for Piconet Clock Adjust test cases. The IUT may use the specified HCI or any equivalent method to set up the test parameters.

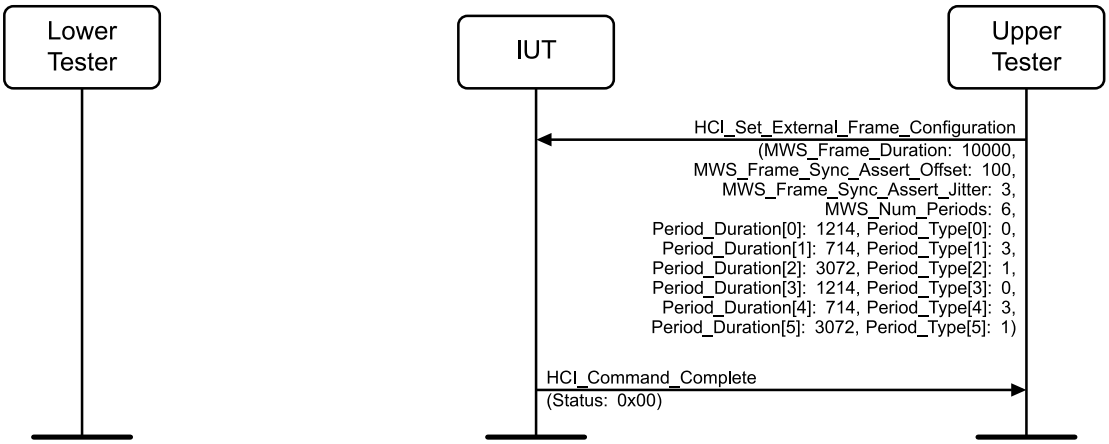


Figure 4.3-4: External Frame Configuration

4.3.6 Pass/Inconclusive/Fail verdict conventions

Each test case has an Expected Outcome section. The IUT is granted the Pass verdict when all the detailed pass criteria conditions within the Expected Outcome section are met.

Certain test cases also have an Inconclusive verdict defined. If the conditions for this verdict are met, then the test provides evidence that the IUT neither meets nor violates the test case; instead, it means that the test case was not applicable to the IUT, and therefore a Pass verdict is not required in order to achieve Qualification of the IUT. Implementers are encouraged to provide mechanisms to avoid the behavior leading to an Inconclusive condition during testing.

The convention in this Test Suite is that, unless there is a specific set of fail conditions outlined in the test case, the IUT fails the test case as soon as one of the pass criteria conditions cannot be met. If this occurs, then the outcome of the test is a Fail verdict.

For an Inconclusive verdict, all the pass criteria conditions apply up to the point in the test procedure where an Inconclusive verdict is identified. If one of the pass criteria in a step prior to the Inconclusive verdict cannot be met, then the outcome of the test is the Fail verdict and not the Inconclusive verdict.

4.4 Common Packet Contents

4.4.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.4.2 Fields and Bits Reserved for Future Use

Unless a specific test states otherwise, all fields within packets and all bits within fields that are described as reserved for future use are set to 0 in packets sent by the Upper and Lower Testers.

4.5 Authentication procedures

Verify the correct implementation of the Authentication services.

4.5.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.5.2 Authentication – Both Central and Peripheral

Verify the authentication procedure. The role of the IUT is of no importance.

LMP/AUT/BI-01-C [Error Return When a Unit Key is Requested]

- Test Purpose

  Verify that the IUT properly returns an error when the Lower Tester requests the Unit Key.

- Reference

  [1] 4.2.2.1, 4.2.2.4

- Initial Condition
  - See the “Baseband assumptions” section.
  - The IUT is the Initiator and the Lower Tester is the Responder.
  - ACL connection establishment has started.
- Test Procedure

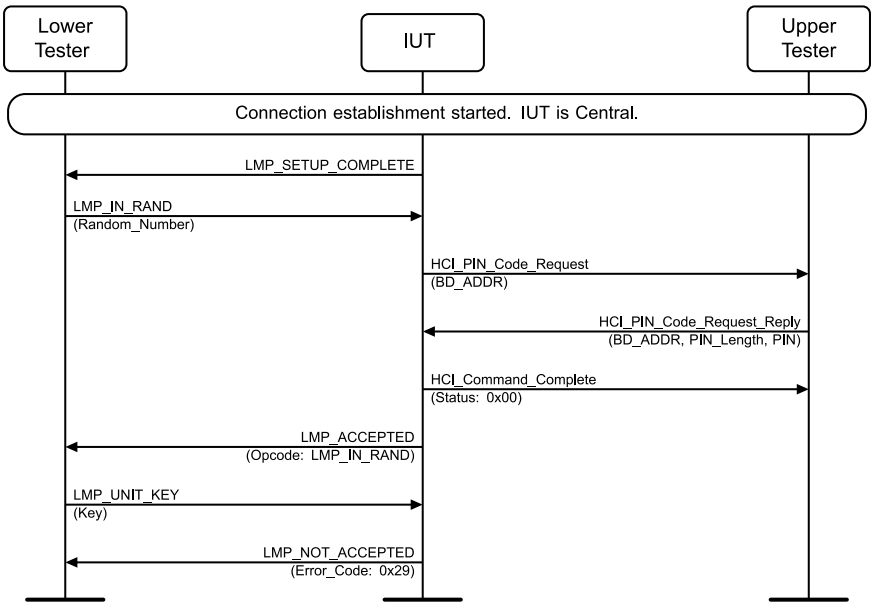


Figure 4.5-1: LMP/AUT/BI-01-C [Error Return When a Unit Key is Requested] MSC

1. After the IUT sends the LMP_SETUP_COMPLETE PDU, the Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_IN_RAND PDU Opcode.
5. The Lower Tester sends an LMP_UNIT_KEY PDU to the IUT with a Key.
6. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the Error_Code set to 0x29 (Pairing with Unit Key Not Supported).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with the Error_Code set to 0x29 (Pairing with Unit Key Not Supported) upon reception of the LMP_UNIT_KEY PDU from the Lower Tester.

LMP/AUT/BV-01-C [Authentication Reject, No Link Key]

- Test Purpose

  Verify that the IUT rejects the authentication when the IUT has no link key associated with the Lower Tester.

- Reference

  [1] 4.2.1.2

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the Responder and has no link key associated with the Lower Tester, which is the Initiator.
  - ACL connection establishment has started.
  - Authentication and encryption are disabled.

- Test Procedure

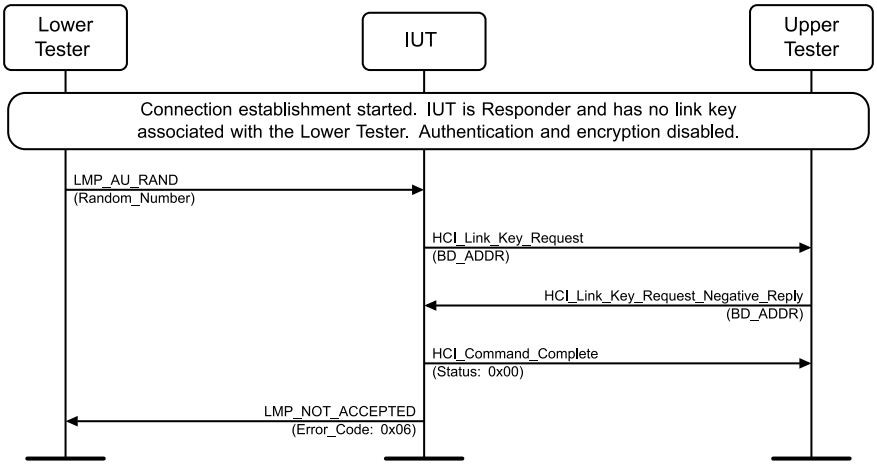


Figure 4.5-2: LMP/AUT/BV-01-C [Authentication Reject, No Link Key] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester sends an HCI_Link_Key_Request_Negative_Reply command to the IUT with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_AU_RAND PDU Opcode and Error_Code set to 0x06 (PIN or Key Missing).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_AU_RAND PDU Opcode and Error_Code set to 0x06 (PIN or Key Missing) upon reception of the LMP_AU_RAND PDU from the Lower Tester.

LMP/AUT/BV-36-C [Legacy Authentication of Previously Authenticated or Stored Link Key]

- Test Purpose

  Verify that the IUT performs the legacy authentication procedure when requested by the Host on an active connection when the link key has been previously authenticated or a link key is stored on the IUT.

- Reference

  [1] 4.2.1.1

  [7] 7.1.15

- Initial Condition
  - See the “Baseband assumptions” section.
  - An ACL connection has been established.
  - The IUT is the Initiator and has a link key associated with the Lower Tester. The Lower Tester is the Responder.
  - Authentication has been previously performed, and encryption has been enabled.
  - The Lower Tester does not support Secure Connections.

- Test Procedure

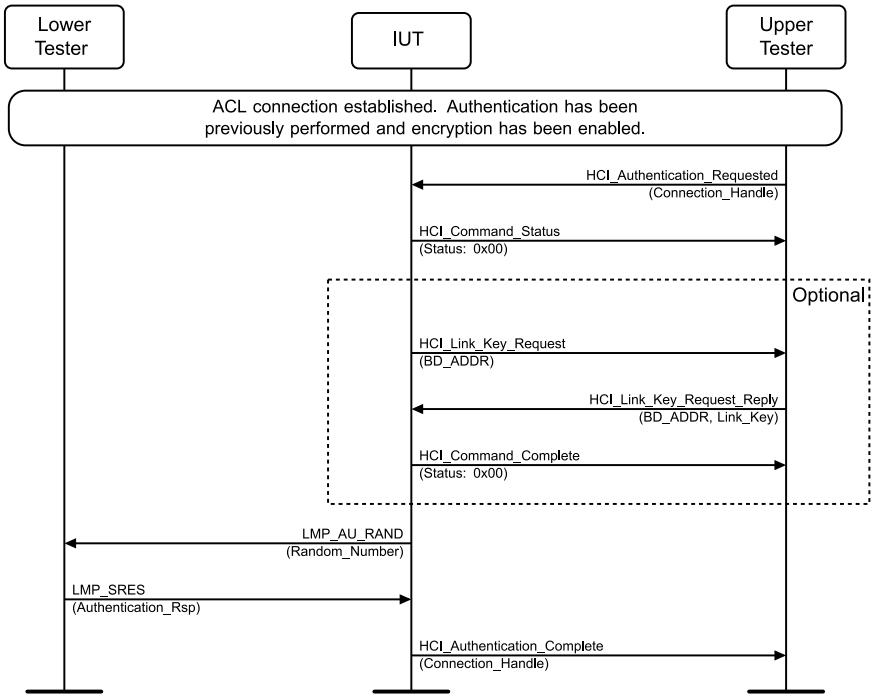


Figure 4.5-3: LMP/AUT/BV-36-C [Legacy Authentication of Previously Authenticated or Stored Link Key] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with Connection_Handle and receives a successful HCI_Command_Status event in return.
2. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.

3. If Step 2 occurs, the Upper Tester sends an HCI_Link_Key_Request_Reply command to the IUT with the BD_ADDR and Link_Key and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.
6. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_AU_RAND PDU to the Lower Tester containing the Random_Number parameter.

  The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester after receiving an LMP_SRES PDU with the Authentication_Rsp from the Lower Tester.

LMP/AUT/BV-40-C [Legacy Authentication of Previously Authenticated or Stored Link Key, v6.0]

- Test Purpose

  Verify that the IUT performs the legacy authentication procedure when requested by the Host on an active connection when the link key has been previously authenticated or a link key is stored on the IUT.

- Reference

  [1] 4.2.1.1

  [7] 7.1.15

- Initial Condition

  - See the “Baseband assumptions” section.
  - An ACL connection has been established.
  - Authentication has been previously performed, and encryption has been enabled.
  - The IUT is the Initiator and has a link key associated with the Lower Tester. The Lower Tester is the Responder.
  - The Lower Tester does not support Secure Connections.

- Test Procedure

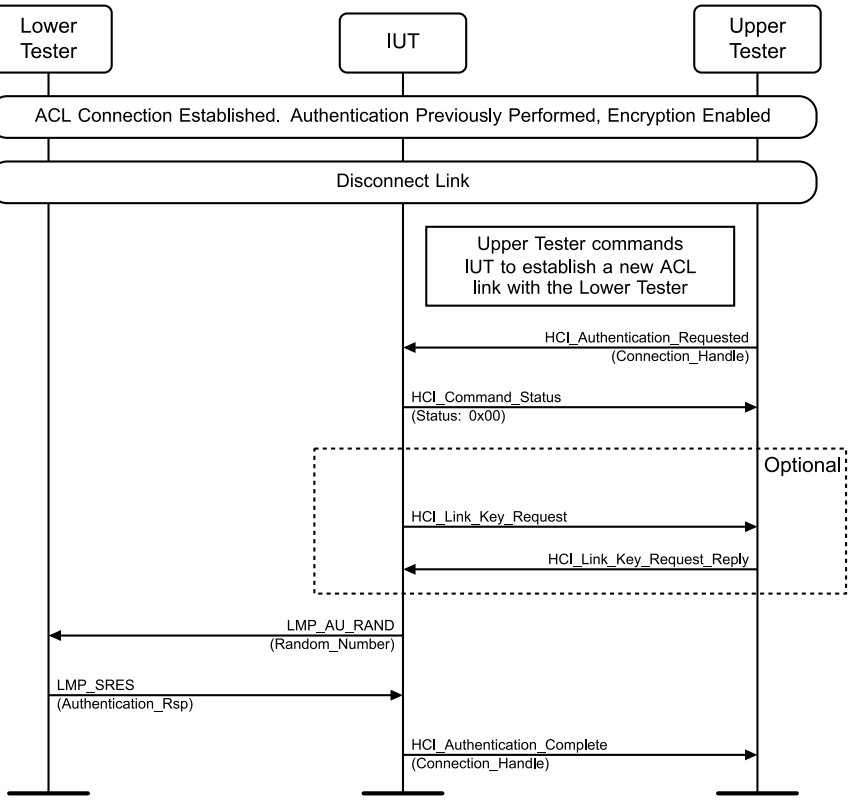


Figure 4.5-4: LMP/AUT/BV-40-C [Legacy Authentication of Previously Authenticated or Stored Link Key, v6.0] MSC

1. The Lower Tester and the IUT disconnect the ACL.
2. The Upper Tester commands the IUT to establish a new ACL connection.
3. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status in return.
4. Optionally, the IUT sends an HCI_Link_Key_Request command to the Upper Tester and receives an HCI_Link_Key_Request_Reply command with the stored link key in response.
5. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
6. The Lower Tester responds with an LMP_SRES PDU to the IUT.
7. The IUT sends an HCI_Authentication_Complete event to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 5, the IUT transmits the LMP_AU_RAND PDU to the Lower Tester containing the Random_Number parameter.

  In Step 7, the IUT transmits a successful HCI_Authentication_Complete event to the Upper Tester after receiving an LMP_SRES PDU with the Authentication_Rsp from the Lower Tester.

  Fail verdict

  The IUT sends an HCI_PIN_Code_Request to the Upper Tester at any point during the test.

4.5.3 Pairing – Both Central and Peripheral

Verify the pairing procedure. The role of the IUT is of no importance.

LMP/AUT/BV-03-C [Create Link Key]

- Test Purpose

  Verify that the IUT creates a valid link key.

- Reference

  [1] 4.2.2

- Initial Condition
  - See the “Baseband assumptions” section.
  - The IUT is the Responder and has a variable PIN code. The Lower Tester is the Initiator.
  - The Lower Tester does not support Secure Simple Pairing.
  - ACL connection establishment has started, and the IUT has sent an LMP_SETUP_COMPLETE PDU.
  - Authentication and encryption are disabled.

- Test Procedure

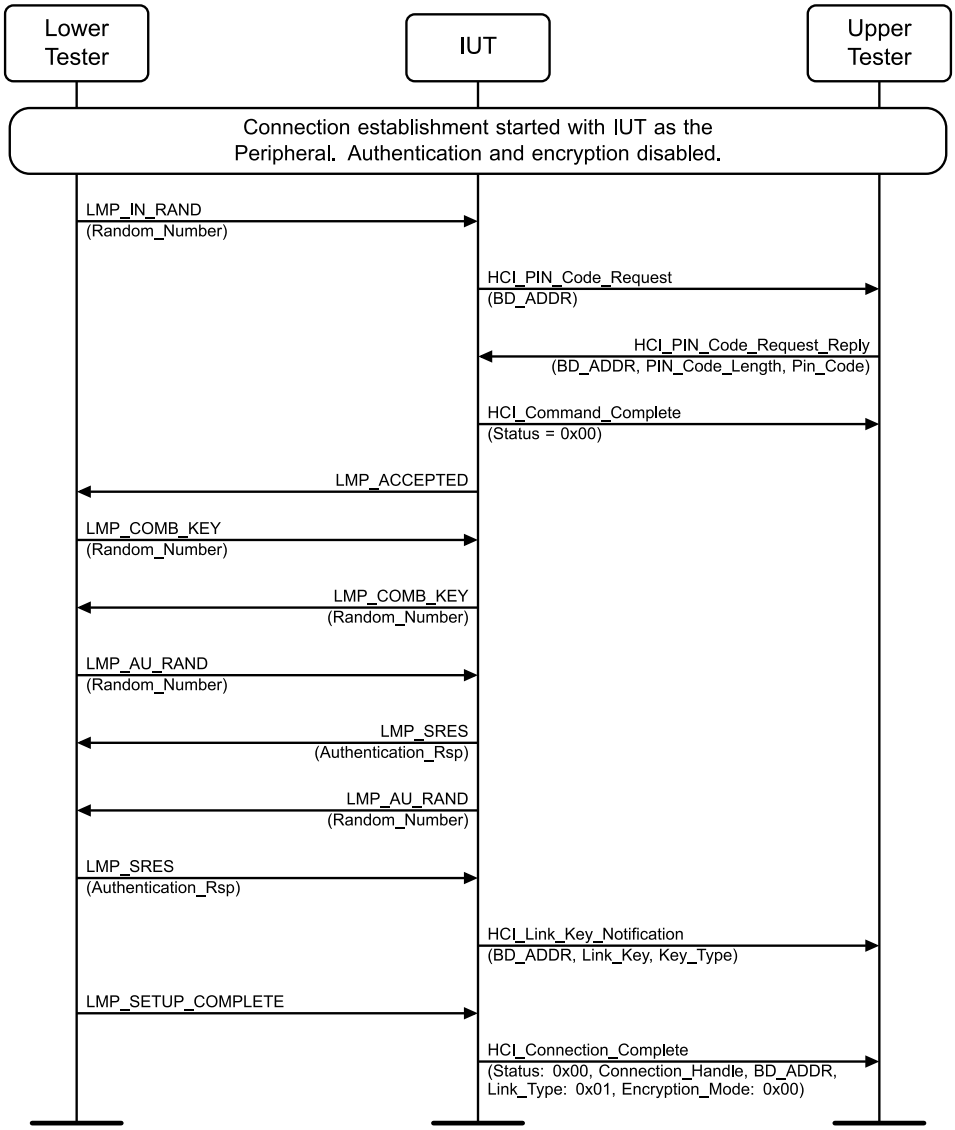


Figure 4.5-5: LMP/AUT/BV-03-C [Create Link Key] MSC

1. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_IN_RAND PDU Opcode.
5. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
6. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.
7. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
8. The IUT responds to the Lower Tester with an LMP_SRES PDU with an Authentication_Rsp and an LMP_AU_RAND PDU with a Random_Number.
9. The Lower Tester sends an LMP_SRES PDU to the IUT with an Authentication_Rsp.

10. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 9.
11. The Lower Tester sends the LMP_SETUP_COMPLETE PDU to the IUT.
12. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_IN_RAND PDU Opcode after receiving the LMP_IN_RAND PDU.

  In Step 6, the IUT sends the LMP_COMB_KEY PDU to the Lower Tester after receiving the LMP_COMB_KEY PDU.

4.5.3.1 Pairing, IUT Initiator

- Test Purpose

  Verify that the IUT initiates a complete pairing and authentication procedure.

- Reference

  [1] 4.2.2.1

- Initial Condition
  - See the “Baseband assumptions” section.
  - The IUT is the Initiator. The Lower Tester is the Responder.
  - The Lower Tester does not support Secure Simple Pairing.
  - If the HCI Command Required to Pair is No in Table 4.5-1, the Upper Tester sends an HCI_Write_Authentication_Enable command to the IUT with Enabled set to 0x01 before starting the connection.
  - ACL connection establishment has started.
  - Authentication and encryption are disabled.

- Test Case Configuration

Test Case	HCI Command Required to Pair
LMP/AUT/BV-04-C [Pairing, IUT Initiator]	No
LMP/AUT/BV-52-C [Pairing, IUT Initiator – HCI Command Required to Pair]	Yes

Table 4.5-1: Pairing, IUT Initiator test cases

- Test Procedure

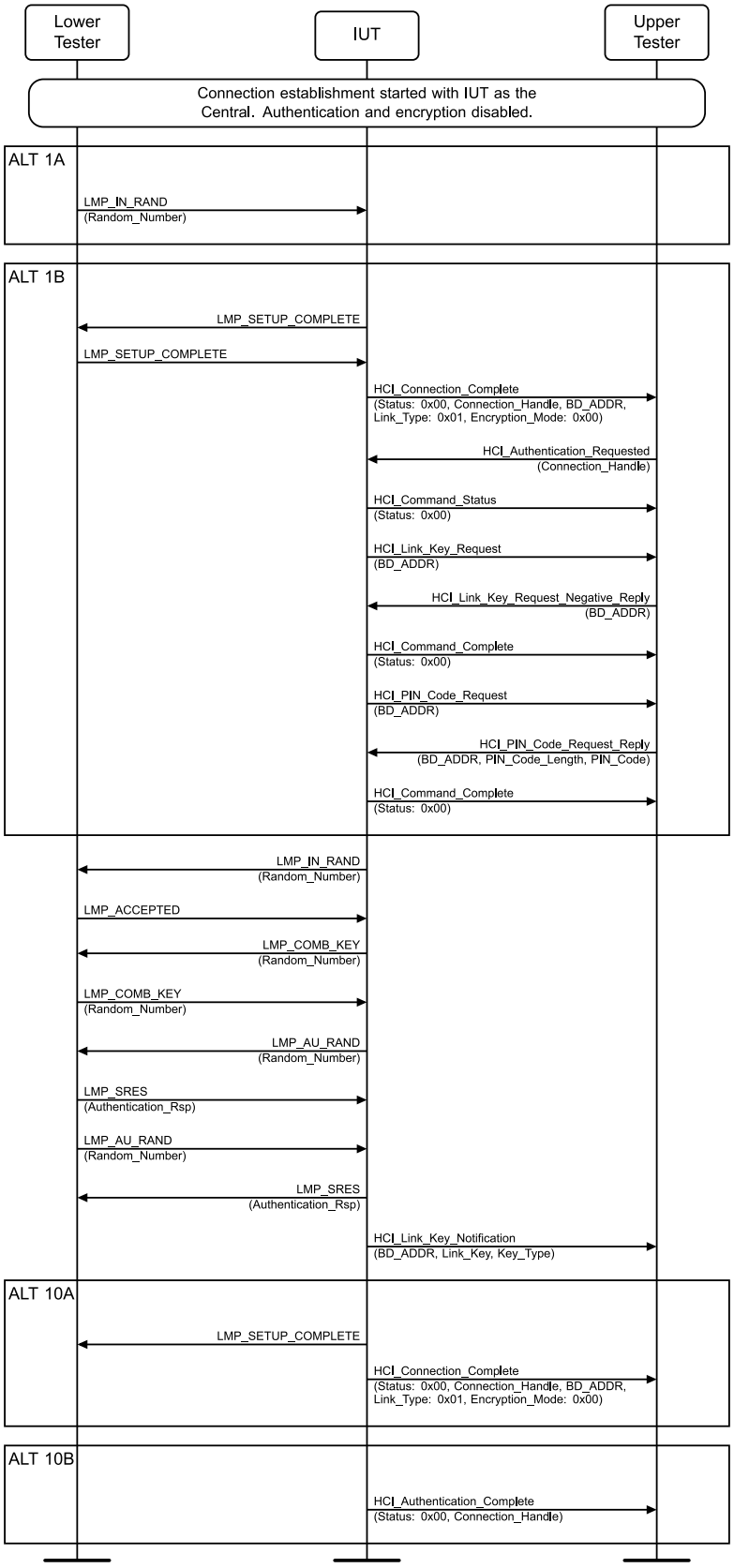


Figure 4.5-6: Pairing, IUT Initiator MSC

1. Perform either alternative 1A or 1B depending on whether the IUT requires an HCI command to pair as indicated in Table 4.5-1.
   Alternative 1A (The IUT does not require an HCI command to pair):
      1A.1. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.
   Alternative 1B (The IUT requires an HCI command to pair):
      1B.1. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.
      1B.2. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.
      1B.3. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).
      1B.4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
      1B.5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
      1B.6. The Upper Tester sends an HCI_Link_Key_Request_Negative_Reply command to the IUT with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
      1B.7. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
      1B.8. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
2. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_IN_RAND PDU Opcode.
4. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
6. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
7. The Lower Tester responds to the IUT with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random_Number.
8. The IUT sends an LMP_SRES PDU to the Lower Tester.
9. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 8.
10. Perform either alternative 10A or 10B depending on whether the IUT requires an HCI command to pair as indicated in Table 4.5-1.
    Alternative 10A (The IUT does not require an HCI command to pair):
       10A.1. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.
       10A.2. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).
    Alternative 10B (The IUT requires an HCI command to pair):
       10B.1. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_IN_RAND PDU.

  In Step 4, the IUT sends the LMP_COMB_KEY PDU.

  After Step 5, the link key is created and checked by a mutual authentication (SRES is checked).

LMP/AUT/BV-05-C [IUT Responder, Fixed PIN]

- Test Purpose

  Verify that when the IUT has a fixed PIN it can request to become the Initiator.

- Reference

  [1] 4.2.2.2

- Initial Condition
  - See the “Baseband assumptions” section.
  - The IUT is the Responder and has a fixed PIN code. The Lower Tester is the Initiator.
  - The Lower Tester does not support Secure Simple Pairing.
  - ACL connection establishment has started, and the IUT has sent the LMP_SETUP_COMPLETE PDU.
  - Authentication and encryption are disabled.

- Test Procedure

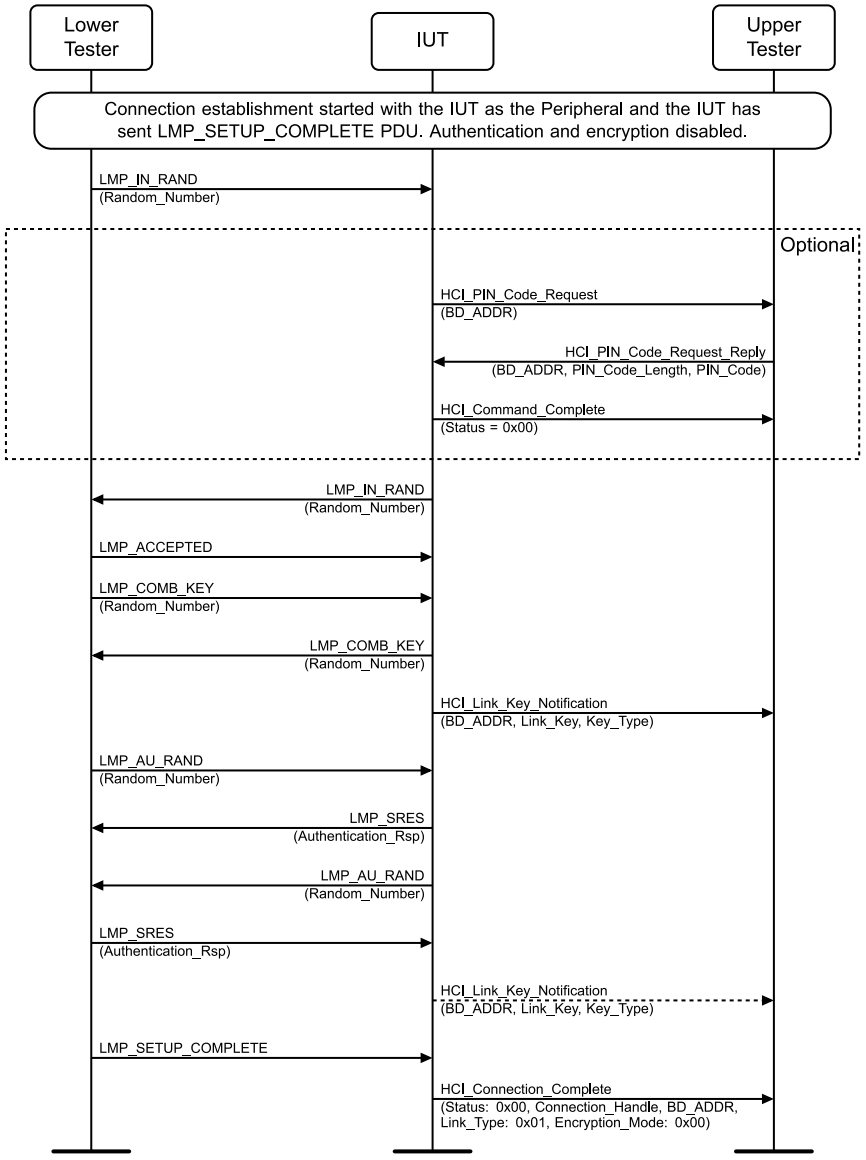


Figure 4.5-7: LMP/AUT/BV-05-C [IUT Responder, Fixed PIN] MSC

1. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
2. Optionally, the IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
3. If Step 2 occurs, the Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_IN_RAND PDU Opcode.
6. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
7. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.
8. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
9. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.

10. The IUT responds to the Lower Tester with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random_Number.
11. The Lower Tester sends an LMP_SRES PDU to the IUT.
12. Optionally, the IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
13. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.
14. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_IN_RAND PDU upon reception of the LMP_IN_RAND PDU from the Lower Tester.

  After Step 7, the correct K_{init} is generated.

4.5.3.2 IUT Initiator; Responder has Fixed PIN

- Test Purpose

  Verify that the IUT accepts that the Lower Tester has a fixed PIN.

- Reference

  [1] 4.2.2.2

- Initial Condition

  - See the “Baseband assumptions” section.
  - The Lower Tester is the Responder and has a fixed PIN code. The IUT is the Initiator and does not have a fixed PIN code.
  - The Lower Tester does not support Secure Simple Pairing.
  - If the HCI Command Required to Pair is No in Table 4.5-2, the Upper Tester sends an HCI_Write_Authentication_Enable command to the IUT with Enabled set to 0x01 before starting the connection.
  - ACL connection establishment has started.
  - Authentication and encryption are disabled.

- Test Case Configuration

Test Case	HCI Command Required to Pair
LMP/AUT/BV-06-C [IUT Initiator; Responder has Fixed PIN]	No
LMP/AUT/BV-53-C [IUT Initiator; Responder has Fixed PIN – HCI Command Required to Pair]	Yes

Table 4.5-2: IUT Initiator; Responder has Fixed PIN test cases

- Test Procedure

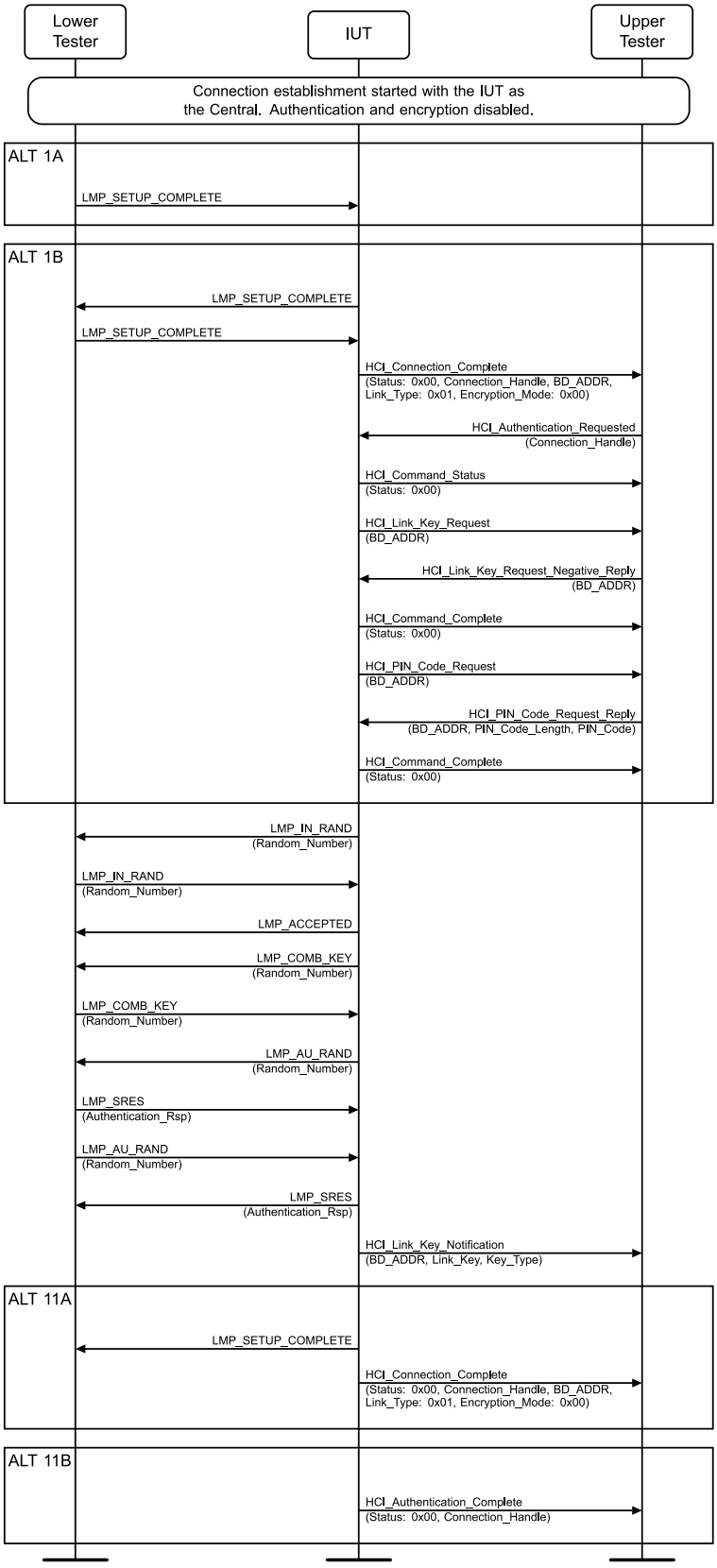


Figure 4.5-8: IUT Initiator; Responder has Fixed PIN MSC

1. Perform either alternative 1A or 1B depending on whether the IUT requires an HCI command to pair as indicated in Table 4.5-2.
   Alternative 1A (The IUT does not require an HCI command to pair):
   - 1A.1 The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.

   Alternative 1B (The IUT requires an HCI command to pair):
   - 1B.1. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.
   - 1B.2. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.
   - 1B.3. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).
   - 1B.4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
   - 1B.5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
   - 1B.6. The Upper Tester sends an HCI_Link_Key_Request_Negative_Reply command to the IUT with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
   - 1B.7. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
   - 1B.8. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
2. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
3. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_IN_RAND PDU Opcode.
5. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
6. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
7. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
8. The Lower Tester responds to the IUT with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random_Number.
9. The IUT sends an LMP_SRES PDU to the Lower Tester.
10. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 9.
11. Perform either alternative 11A or 11B depending on whether the IUT requires an HCI command to pair as indicated in Table 4.5-2.
    Alternative 11A (The IUT does not require an HCI command to pair):
    - 11A.1. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.
    - 11A.2. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).

    Alternative 11B (The IUT requires an HCI command to pair):
    - 11B.1. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_IN_RAND PDU from the Lower Tester.

LMP/AUT/BV-24-C [Create Link Key – Rejects Bad Authentication Response]

- Test Purpose

  Verify that the IUT creates the correct link key and rejects a bad authentication response.

- Reference

  [1] 4.2.2

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the Responder and has a variable PIN code. The Lower Tester is the Initiator.
  - The Lower Tester does not support Simple Pairing.
  - ACL connection establishment has started, and the IUT has sent an LMP_SETUP_COMPLETE PDU.

- Test Procedure

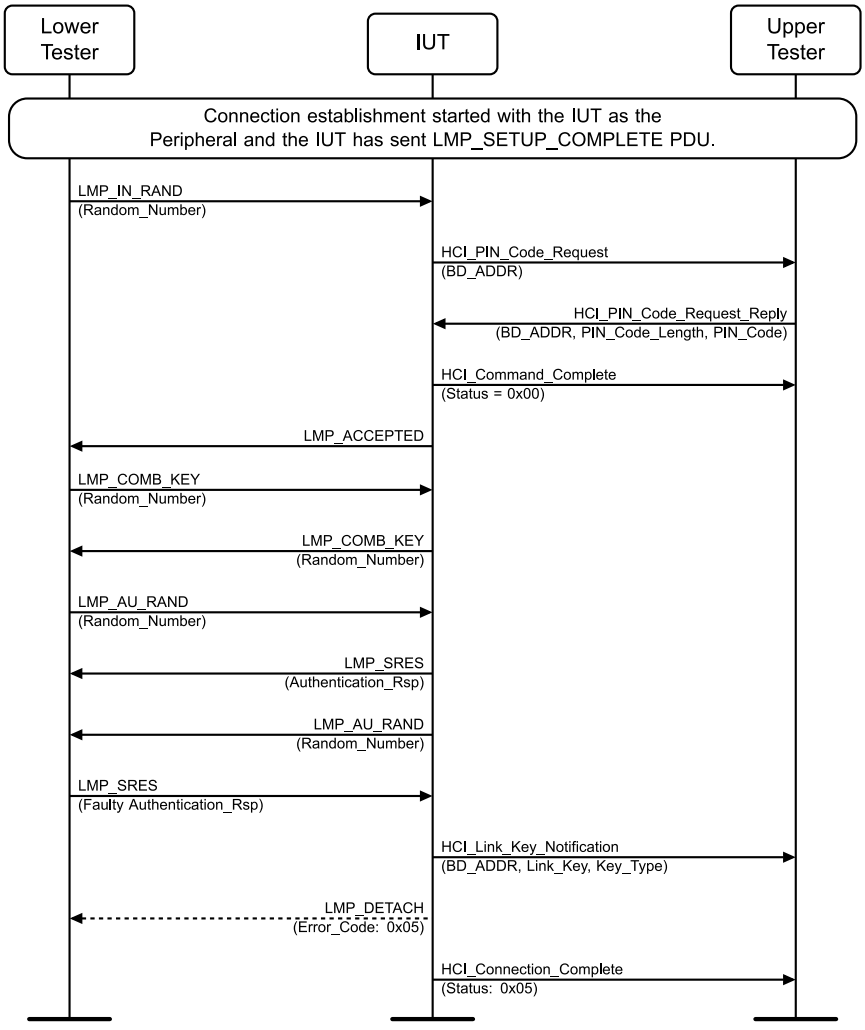


Figure 4.5-9: LMP/AUT/BV-24-C [Create Link Key – Rejects Bad Authentication Response] MSC

1. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.

3. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_IN_RAND PDU Opcode.
5. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
6. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.
7. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
8. The IUT responds to the Lower Tester with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random_Number.
9. The Lower Tester sends an LMP_SRES PDU to the IUT with one bit at random of the Authentication_Rsp inverted.
10. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 9.
11. Optionally, the IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).
12. The IUT sends an HCI_Connection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_ACCEPTED PDU to the Lower Tester with the LMP_IN_RAND PDU Opcode upon reception of the LMP_IN_RAND PDU from the Lower Tester.

  In Step 6, the IUT sends the LMP_COMB_KEY PDU to the Lower Tester upon reception of the LMP_COMB_KEY PDU from the Lower Tester.

  The IUT rejects the authentication after receiving a bad LMP_SRES PDU from the Lower Tester.

4.5.3.3 Pairing, IUT Initiator – Rejects Bad Authentication Response

- Test Purpose

  Verify that the IUT rejects a bad authentication response in a complete pairing and authentication procedure that it initiates.

- Reference

  [1] 4.2.2.1

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the Initiator. The Lower Tester is the Responder.
  - The Lower Tester does not support Simple Pairing.
  - If the HCI Command Required to Pair is No in Table 4.5-3, the Upper Tester sends an HCI_Write_Authentication_Enable command to the IUT with Enabled set to 0x01 before starting the connection.

- Test Case Configuration

Test Case	Detach	HCI Command Required to Pair
LMP/AUT/BV-25-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Optional]	ALT 10A or ALT 10B	No
LMP/AUT/BV-54-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Optional, HCI Command Required to Pair]	ALT 10A or ALT 10B	Yes
LMP/AUT/BV-45-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Mandatory]	ALT 10B	No
LMP/AUT/BV-55-C [Pairing, IUT Initiator – Rejects Bad Authentication Response, Detach Mandatory, HCI Command Required to Pair]	ALT 10B	Yes

Table 4.5-3: Pairing, IUT Initiator – Rejects Bad Authentication Response test cases

- Test Procedure

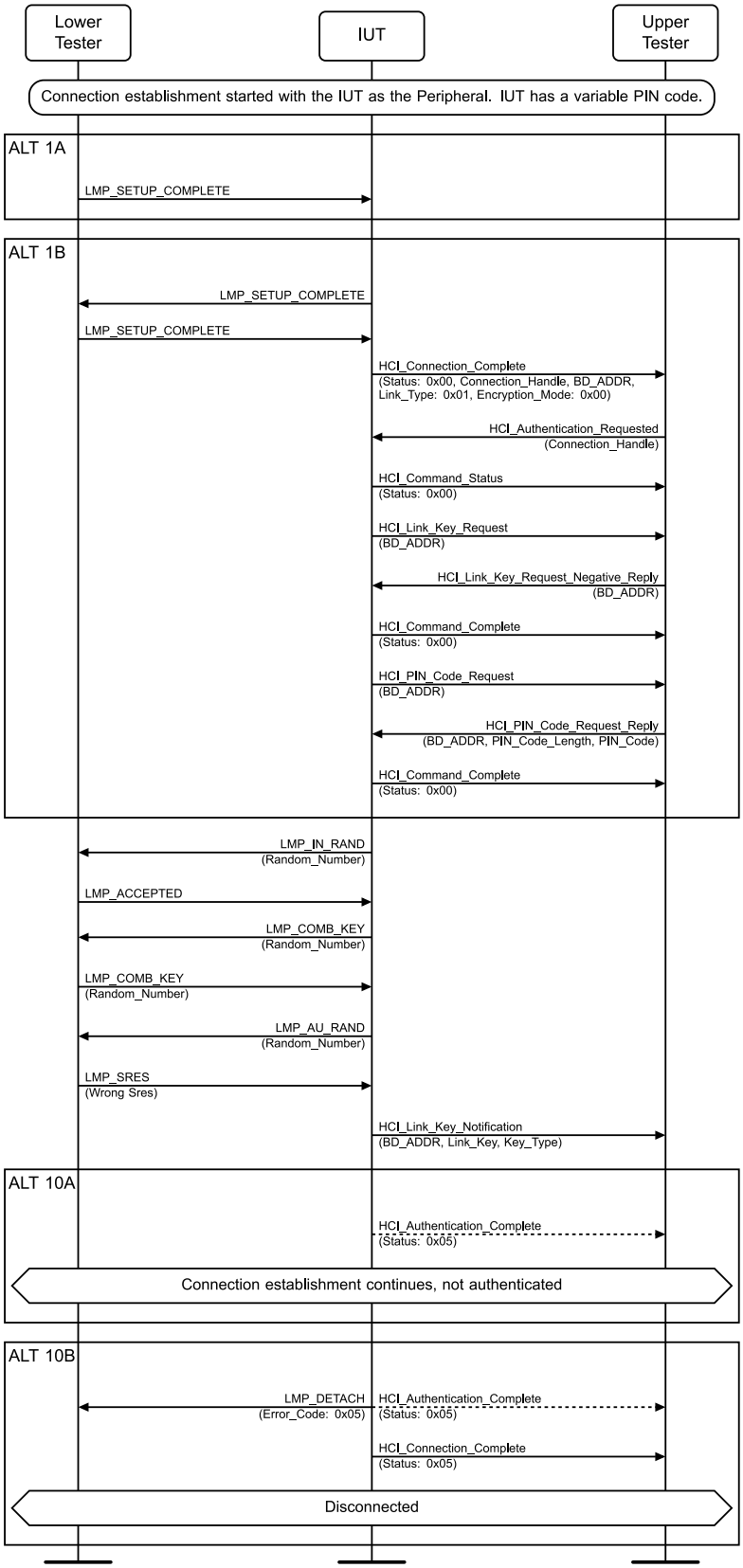


Figure 4.5-10: Pairing, IUT Initiator – Rejects Bad Authentication Response MSC

1. Perform either alternative 1A or 1B depending on whether the IUT requires an HCI command to pair as indicated in Table 4.5-3.
   Alternative 1A (The IUT does not require an HCI command to pair):
   - 1A.1. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.

   Alternative 1B (The IUT requires an HCI command to pair):
   - 1B.1. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.
   - 1B.2. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.
   - 1B.3. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).
   - 1B.4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
   - 1B.5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
   - 1B.6. The Upper Tester sends an HCI_Link_Key_Request_Negative_Reply command to the IUT with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
   - 1B.7. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
   - 1B.8. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
2. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_IN_RAND PDU Opcode.
4. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
6. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
7. The Lower Tester responds to the IUT with an LMP_SRES PDU with one bit at random of the Authentication_Rsp inverted.
8. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 7.
9. If the IUT does not require an HCI command to pair, the Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
10. Perform either alternative 10A or 10B as indicated in Table 4.5-3.
    Alternative 10A (Detach is optional):
    - 10A.1. If the IUT requires an HCI command to pair, the IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
    - 10A.2. The IUT and the Lower Tester continue connection establishment not authenticated.

    Alternative 10B (Detach is mandatory):
    - 10B.1. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).
    - 10B.2. If the IUT requires an HCI command to pair, the IUT optionally sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
    - 10B.3. The IUT sends an HCI_Connection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
    - 10B.4. The IUT and the Lower Tester disconnect.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the correct LMP_IN_RAND PDU.

  In Step 4, the IUT sends the correct LMP_COMB_KEY PDU.

  After Step 6, the IUT and the Lower Tester mutually authenticate the link key.

  After Step 8, the IUT rejects the authentication when receiving a bad LMP_SRES PDU from the Lower Tester.

  If the IUT sends the LMP_DETACH PDU in alternative 10B, then the IUT and the Lower Tester disconnect.

LMP/AUT/BV-26-C [IUT Responder, Fixed PIN – Rejects Bad Authentication Response]

- Test Purpose

  Verify that when the IUT has a fixed PIN it can request to become the Initiator and rejects a bad authentication response.

- Reference

  [1] 4.2.3

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the Responder and has a fixed PIN code. The Lower Tester is the Initiator.
  - The Lower Tester does not support Simple Pairing.
  - ACL connection establishment has started and the IUT has sent the LMP_SETUP_COMPLETE PDU.

- Test Procedure

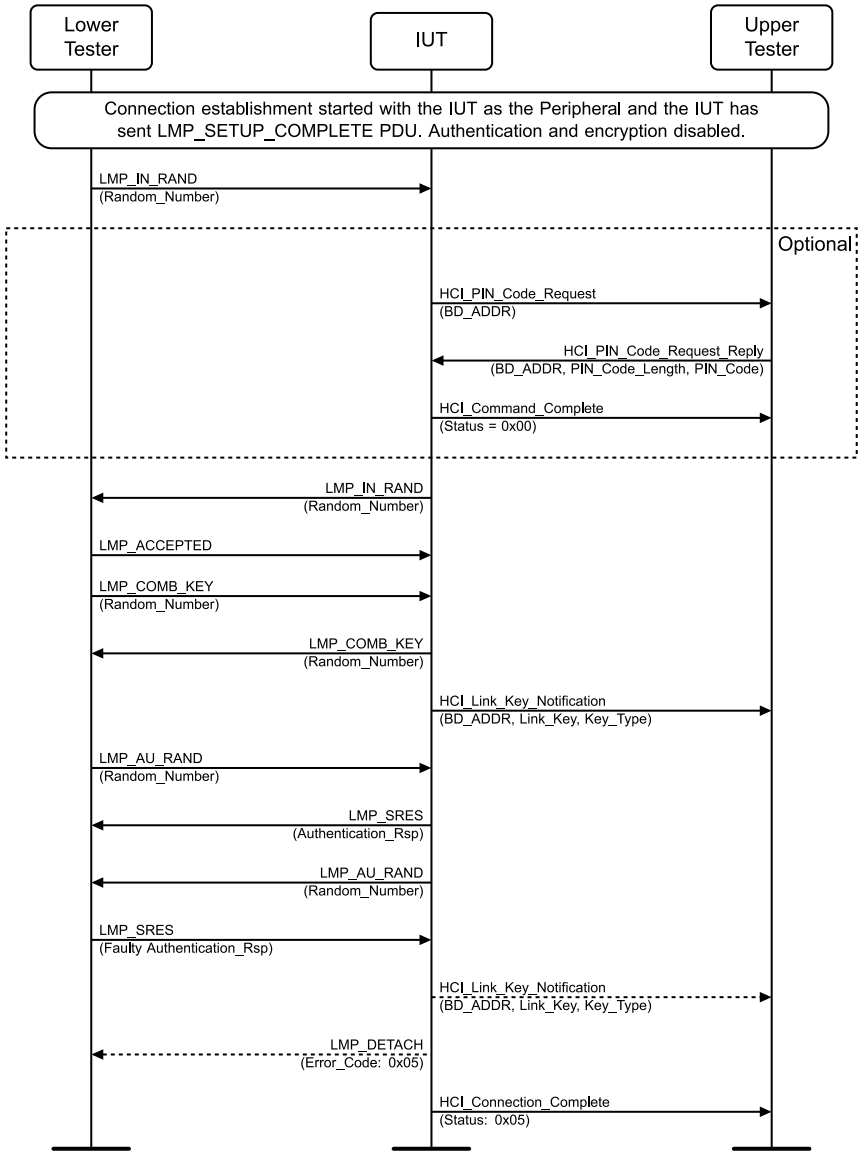


Figure 4.5-11: LMP/AUT/BV-26-C [IUT Responder, Fixed PIN – Rejects Bad Authentication Response] MSC

1. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
2. Optionally, the IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
3. If Step 2 occurs, the Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_IN_RAND PDU Opcode.
6. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
7. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.
8. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
9. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.

10. The IUT responds to the Lower Tester with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random_Number.
11. The Lower Tester sends an LMP_SRES PDU to the IUT with one bit at random of the Authentication_Rsp inverted.
12. Optionally, the IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
13. Optionally, the IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).
14. The IUT sends an HCI_Connection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_IN_RAND PDU to the Lower Tester upon reception of the LMP_IN_RAND PDU from the Lower Tester.

  The IUT rejects the authentication after receiving a bad LMP_SRES PDU from the Lower Tester.

4.5.3.4 IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response

- Test Purpose

  Verify that the IUT accepts that the Lower Tester has a fixed PIN and rejects a bad authentication response.

- Reference

  [1] 4.2.2.2

- Initial Condition

  - See the “Baseband assumptions” section.
  - The Lower Tester is the Responder and has a fixed PIN code. The IUT is the Initiator and does not have a fixed PIN code.
  - The Lower Tester does not support Simple Pairing.
  - ACL connection establishment has started.
  - Authentication and encryption are disabled.
  - If the HCI Command Required to Pair is No in Table 4.5-4, the Upper Tester sends an HCI_Write_Authentication_Enable command with Enabled set to 0x01 before starting the connection.

- Test Case Configuration

Test Case	Detach	HCI Command Required to Pair
LMP/AUT/BV-27-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Optional]	ALT 10A or ALT 10B	No
LMP/AUT/BV-56-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Optional, HCI Command Required to Pair]	ALT 10A or ALT 10B	Yes

Test Case	Detach	HCI Command Required to Pair
LMP/AUT/BV-46-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Mandatory]	ALT 10B	No
LMP/AUT/BV-57-C [IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response, Detach Mandatory, HCI Command Required to Pair]	ALT 10B	Yes

Table 4.5-4: IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response test cases

- Test Procedure

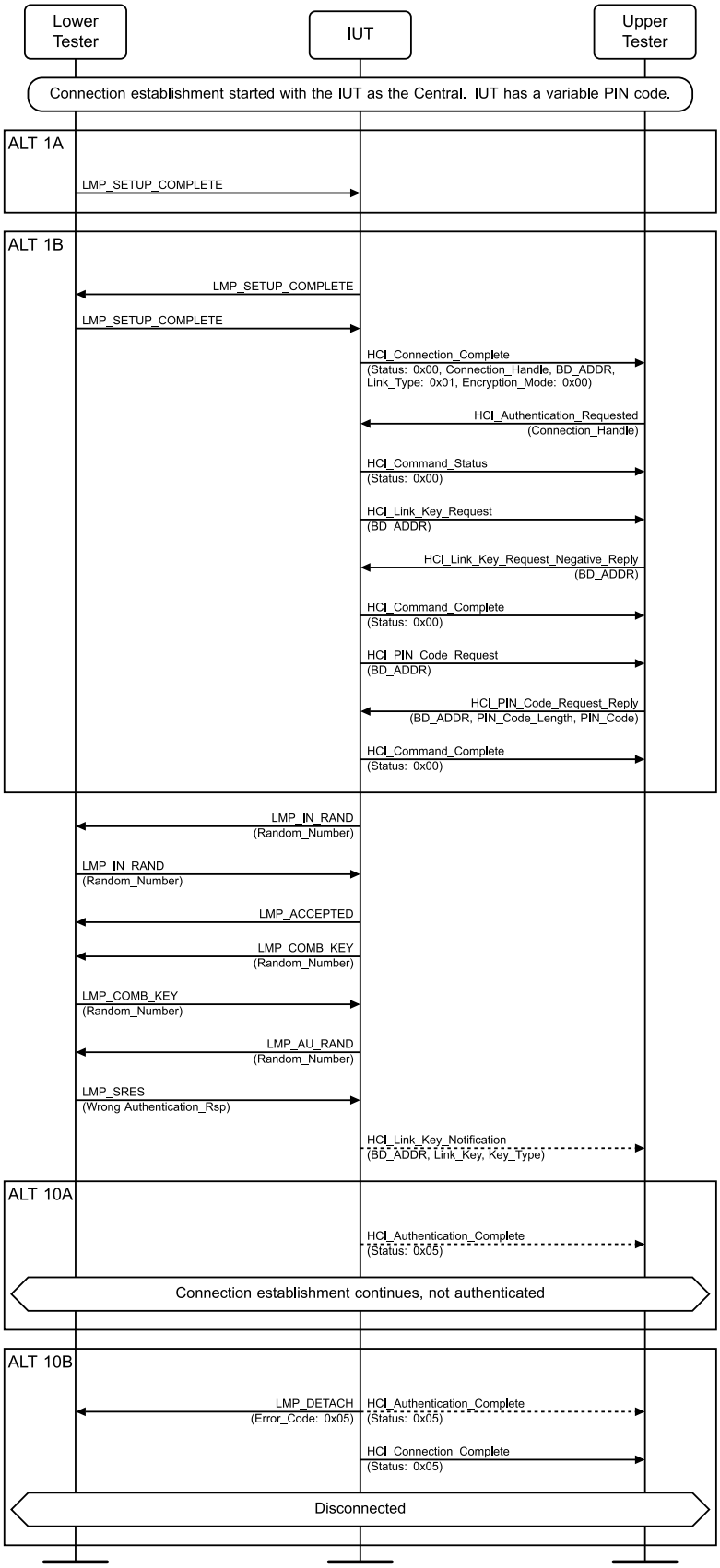


Figure 4.5-12: IUT Initiator; Responder has Fixed PIN – Rejects Bad Authentication Response MSC

1. Perform either alternative 1A or 1B depending on whether the IUT requires an HCI command to pair as indicated in Table 4.5-4.
   Alternative 1A (The IUT does not require an HCI command to pair):
   - 1A.1 The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.

   Alternative 1B (The IUT requires an HCI command to pair):
   - 1B.1. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.
   - 1B.2. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.
   - 1B.3. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).
   - 1B.4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
   - 1B.5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
   - 1B.6. The Upper Tester sends an HCI_Link_Key_Request_Negative_Reply command to the IUT with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
   - 1B.7. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
   - 1B.8. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
2. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
3. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_IN_RAND PDU Opcode.
5. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
6. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
7. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
8. The Lower Tester responds to the IUT with an LMP_SRES PDU with one bit at random of the Authentication_Rsp inverted.
9. The IUT may send an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 9.
10. Perform either alternative 10A or 10B as indicated in Table 4.5-4.
    Alternative 10A (Detach is optional):
    - 10A.1. If the IUT requires an HCI command to pair, the IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
    - 10A.2. The IUT and the Lower Tester continue connection establishment not authenticated.

    Alternative 10B (Detach is mandatory):
    - 10B.1. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).
    - 10B.2. If the IUT requires an HCI command to pair, the IUT optionally sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
    - 10B.3. The IUT sends an HCI_Connection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
    - 10B.4. The IUT and the Lower Tester disconnect.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_ACCEPTED PDU upon reception of the LMP_IN_RAND PDU from the Lower Tester.

  After Step 9, the IUT rejects the authentication when receiving a bad LMP_SRES PDU from the Lower Tester.

LMP/AUT/BV-34-C [Pairing, IUT rejects Pairing Procedure – Host is in Non-Pairable Mode]

- Test Purpose

  Verify that the IUT rejects a pairing procedure if the Host is in non-pairable mode (Host sends PIN Code Request Negative Reply command).

- Reference

  [1] 4.2.2, 4.2.2.3

- Initial Condition

  - The IUT is the Peripheral and the Claimant. The Lower Tester is the Central and the Verifier of the pairing procedure.
  - The Lower Tester does not support Simple Pairing.
  - The IUT is connected to the Lower Tester through LMP_HOST_CONNECTION_REQ and LMP_ACCEPTED. The Upper Tester does not allow pairing.
  - The “Connection Establishment Lower Tester” preamble may be used for a Peripheral IUT; otherwise, a comparable initialization sequence should be used.

- Test Procedure

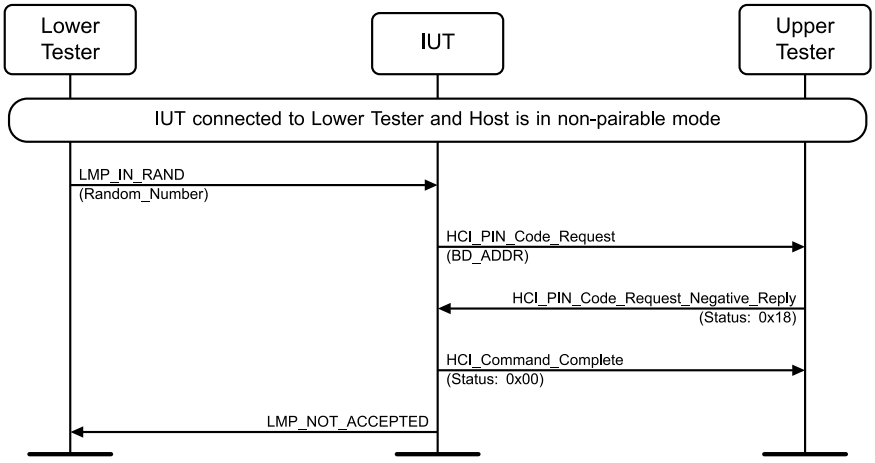


Figure 4.5-13: LMP/AUT/BV-34-C [Pairing, IUT rejects Pairing Procedure – Host is in Non-Pairable Mode] MSC

1. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_PIN_Code_Request to the Upper Tester.
3. The Upper Tester sends an HCI_PIN_Code_Request_Negative_Reply to the IUT with Status set to 0x18 (Pairing Not Allowed) and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_IN_RAND PDU Opcode and Error_Code set to 0x18 (Pairing Not Allowed).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x18 (Pairing Not Allowed) after it has received an LMP_IN_RAND PDU.

LMP/AUT/BI-04-C [Reject Role Switch, Pairing, Responder]

- Test Purpose

  Verify that the IUT rejects a role switch request during the pairing process.

- Reference

  [1] 4.2.3, 4.4.2

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble with Allow_Role_Switch set to 0x01.
  - The IUT is the Responder and has a fixed PIN code. The Lower Tester is the Initiator.
  - The Lower Tester does not support Secure Simple Pairing.

- Test Procedure

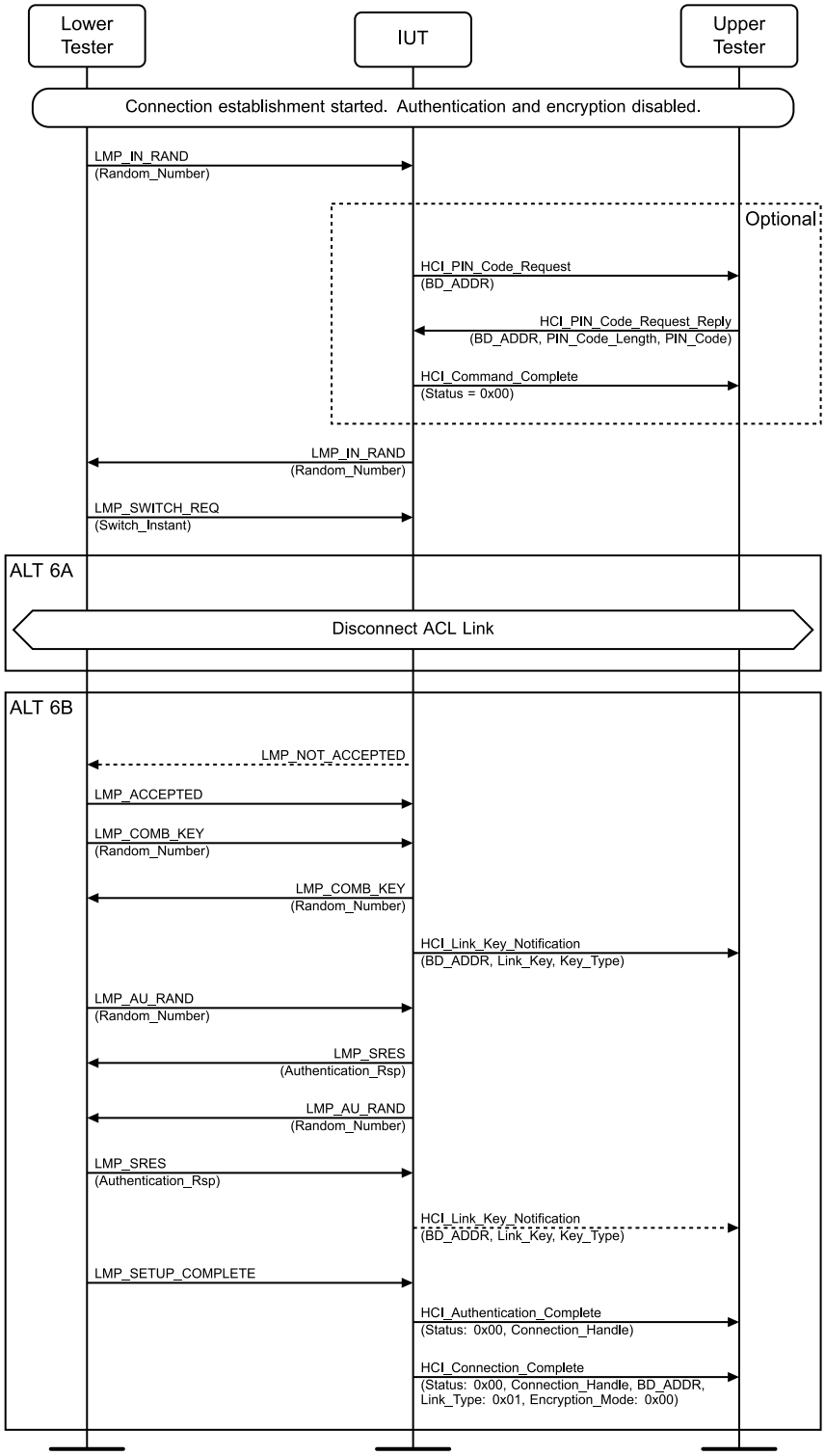


Figure 4.5-14: LMP/AUT/BI-04-C [Reject Role Switch, Pairing, Responder] MSC

1. The Lower Tester sends an LMP_IN_RAND PDU to the IUT with a Random_Number.
2. Optionally, the IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.

3. If Step 2 occurs, the Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
6. Perform either alternative 6A or 6B depending on the IUT's response.

   Alternative 6A (The IUT disconnects the ACL Link):

   6A.1. The IUT disconnects the ACL Link with the Lower Tester.

   Alternative 6B (The IUT continues with the pairing procedure):

   6B.1. Optionally, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
   6B.2. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_IN_RAND PDU Opcode.
   6B.3. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
   6B.4. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
   6B.5. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
   6B.6. The IUT responds to the Lower Tester with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random_Number.
   6B.7. The Lower Tester sends an LMP_SRES PDU to the IUT.
   6B.8. Optionally, the IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
   6B.9. The Lower Tester sends an LMP_SETUP_COMPLETE PDU to the IUT.
   6B.10. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).

- Expected Outcome

  Pass verdict

  In alternative 6A, the IUT disconnects the ACL Link with the Lower Tester.

  In alternative 6B, the IUT sends an HCI_Connection_Complete event to the Upper Tester.

LMP/AUT/BI-08-C [Pairing, IUT Initiator, Invalid Combination Key]

- Test Purpose

  Verify that the IUT rejects an invalid combination key.

- Reference

  [1] 4.2.2.1

- Initial Condition

  - See the "Baseband assumptions" section.
  - If the IUT supports authentication before connection completion, the Upper Tester sends the HCI_Write_Authentication_Enable command with Enabled set to 0x01 before starting the connection.

- Test Procedure

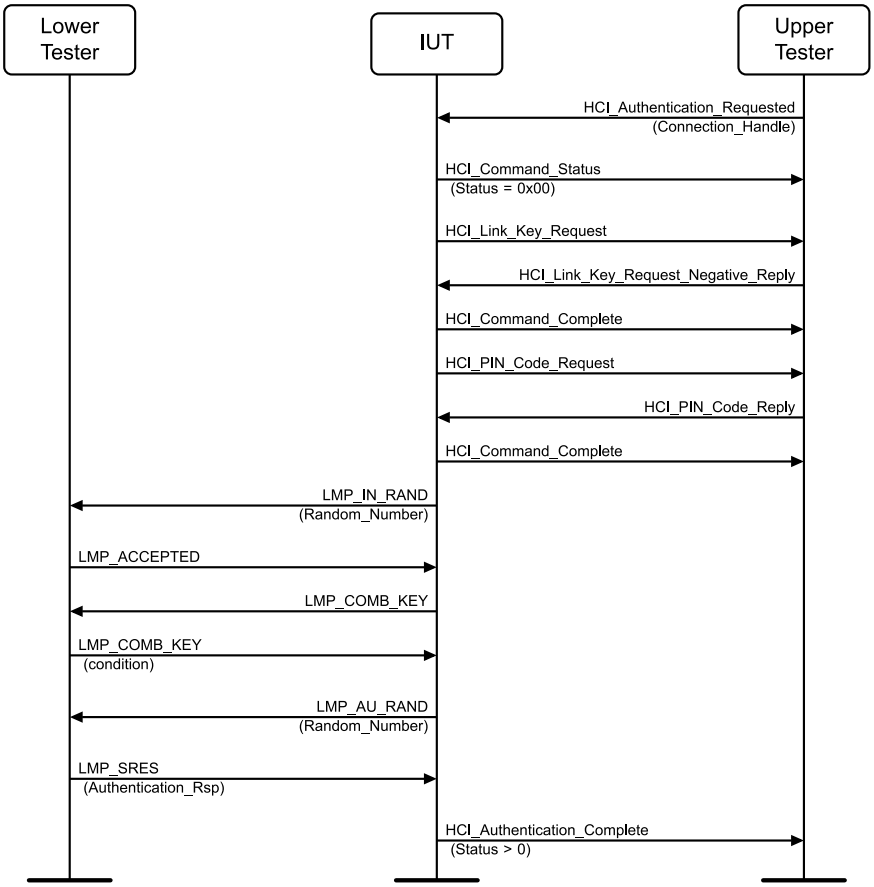


Figure 4.5-15: LMP/AUT/BI-08-C [Pairing, IUT Initiator, Invalid Combination Key] MSC

If the IUT initiates pairing autonomously, skip to Step 7.

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester.
3. The Upper Tester sends an HCI_Link_Key_Request_Negative_Reply command to the IUT.
4. The IUT sends an HCI_Command_Complete event to the Upper Tester.
5. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester.
6. The Upper Tester sends an HCI_PIN_Code_Request_Reply command to the IUT and receives a successful HCI_Command_Complete event in response.
7. The IUT sends an LMP_IN_RAND PDU to the Lower Tester and receives an LMP_ACCEPTED PDU in return.
8. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester.
9. The Lower Tester responds with an LMP_COMB_KEY PDU with CA = CB.
10. Any time after Step 9, the IUT may send an LMP_NOT_ACCEPTED or LMP_DETACH PDU to the Lower Tester. If so, the IUT ends with a Pass verdict.
11. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with Random_Number.
12. The Lower Tester sends an LMP_SRES to the IUT with Authentication_Rsp.
13. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status > 0.

- Expected Outcome

  Pass verdict

  Any time after Step 9, the IUT sends an LMP_NOT_ACCEPTED PDU.

  In Step 13, the IUT sends an error to the Upper Tester.

4.5.4 Change Link Key – Both Central and Peripheral

Verify the Change Link Key procedure. The role of the IUT is of no importance.

LMP/AUT/BV-12-C [Change Link Key, IUT Responder]

- Test Purpose

  Verify that the IUT accepts changing the link key and that the IUT creates the new link key correctly.

- Reference

  [1] 4.2.3

- Initial Condition

  - See the “Baseband assumptions” section.
  - The Lower Tester is the initiating unit configured to use a combination key.
  - The IUT is configured to use a combination key.
  - Encryption is not used.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

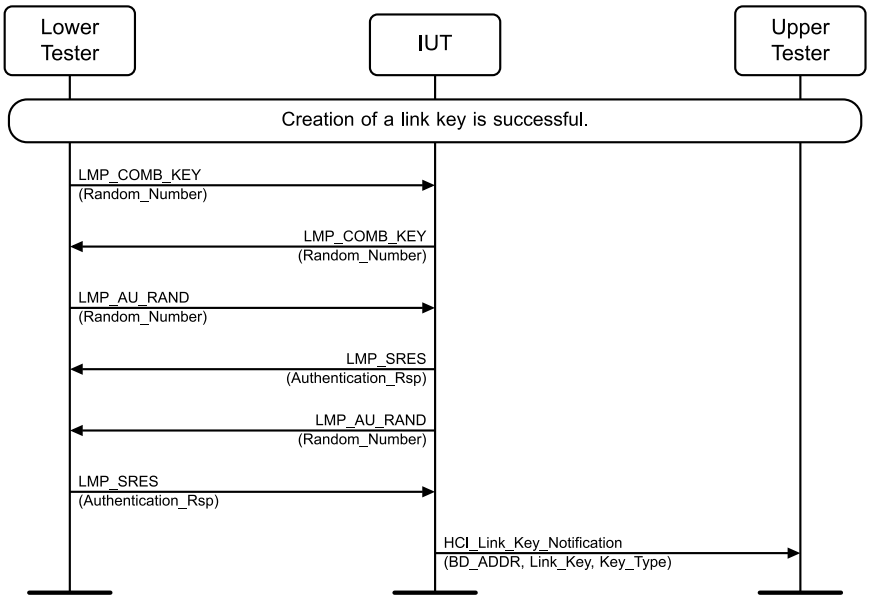


Figure 4.5-16: LMP/AUT/BV-12-C [Change Link Key, IUT Responder] MSC

1. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
2. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.
3. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.

4. The IUT responds to the Lower Tester with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random Number.
5. The Lower Tester sends an LMP_SRES PDU to the IUT.
6. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 5.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_COMB_KEY PDU to the Lower Tester upon reception of the LMP_COMB_KEY PDU from the Lower Tester. The new link key matches the calculated link key.

LMP/AUT/BV-13-C [Change Link Key, IUT Initiator]

- Test Purpose

  Verify that the IUT can change the link key and that the IUT creates the new link key correctly.

- Reference

  [1] 4.2.3

- Initial Condition
  - See the “Baseband assumptions” section.
  - The IUT is the initiating unit configured to use a combination key.
  - Encryption is not used.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

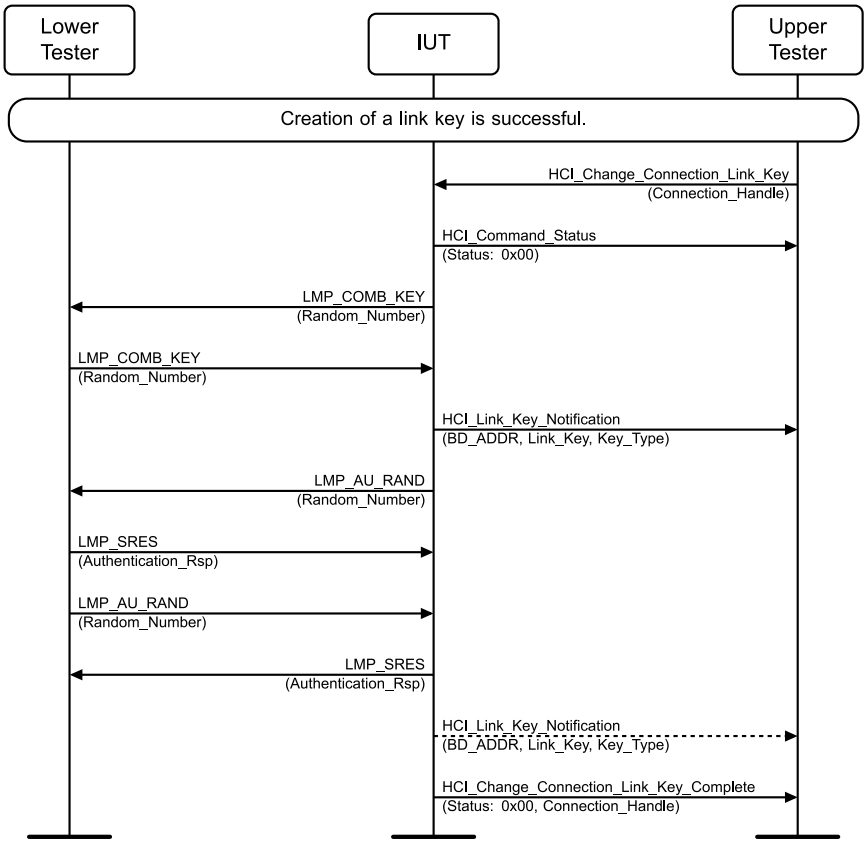


Figure 4.5-17: LMP/AUT/BV-13-C [Change Link Key, IUT Initiator] MSC

1. The Upper Tester sends an HCI_Change_Connection_Link_Key command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event from the IUT.
2. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
4. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
5. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
6. The Lower Tester responds to the IUT with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random_Number.
7. The IUT sends an LMP_SRES PDU to the Lower Tester.
8. Optionally, the IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 7.
9. The IUT sends a successful HCI_Change_Connection_Link_Key_Complete event to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_COMB_KEY PDU to the Lower Tester. The new link key matches the calculated link key.

LMP/AUT/BV-28-C [Change Link Key, IUT Responder – Rejects Bad Authentication Response]

- Test Purpose

  Verify that the IUT accepts changing the link key, creates the new link key correctly and rejects a bad authentication response.

- Reference

  [1] 4.2.3

- Initial Condition

  - See the “Baseband assumptions” section.
  - The Lower Tester is the initiating unit configured to use a combination key.
  - The IUT is configured to use a combination key.
  - No encryption is used.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

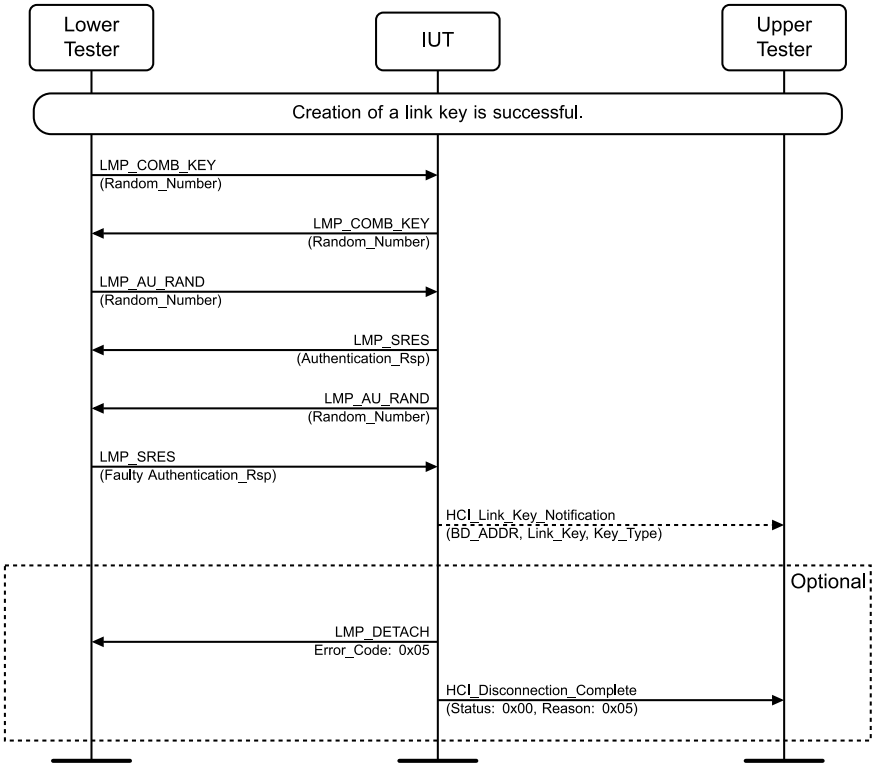


Figure 4.5-18: LMP/AUT/BV-28-C [Change Link Key, IUT Responder – Rejects Bad Authentication Response] MSC

1. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
2. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.
3. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
4. The IUT responds to the Lower Tester with an LMP_SRES PDU and an LMP_AU_RAND PDU with a Random Number.

5. The Lower Tester sends an LMP_SRES PDU to the IUT with one bit at random of the Authentication_Rsp inverted.
6. The IUT may send an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 6.
7. Optionally, the IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure) and sends a successful HCI_Disconnection_Complete event to the Upper Tester with Reason set to 0x05 (Authentication Failure).

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_COMB_KEY PDU to the Lower Tester upon reception of the LMP_COMB_KEY PDU from the Lower Tester. The new link key matches the calculated link key.

  The IUT rejects the authentication after receiving a bad LMP_SRES PDU from the Lower Tester.

4.5.4.1 Change Link Key, IUT Initiator – Rejects Bad Authentication Response

- Test Purpose

  Verify that the IUT can change the link key, creates the new link key correctly, and rejects a bad authentication response.

- Reference

  [1] 4.2.3

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the initiating unit configured to use a combination key.
  - Encryption is not used.
  - The IUT and the Lower Tester have already created a link key.

- Test Case Configuration

Test Case	Detach
LMP/AUT/BV-29-C [Change Link Key, IUT Initiator – Rejects Bad Authentication Response, Detach Optional]	ALT A or ALT B
LMP/AUT/BV-47-C [Change Link Key, IUT Initiator – Rejects Bad Authentication Response, Detach Mandatory]	ALT B

Table 4.5-5: Change Link Key, IUT Initiator – Rejects Bad Authentication Response test cases

- Test Procedure

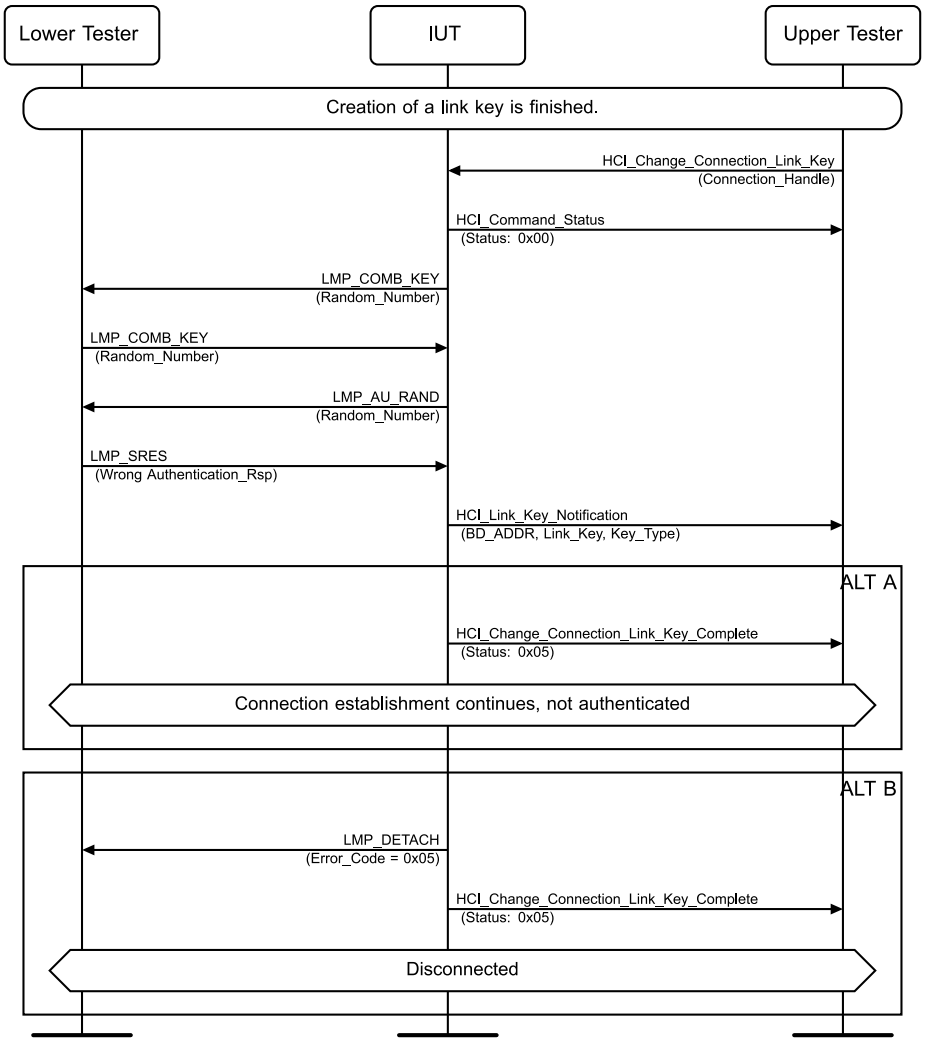


Figure 4.5-19: Change Link Key, IUT Initiator – Rejects Bad Authentication Response MSC

1. The Upper Tester sends an HCI_Change_Connection_Link_Key command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event from the IUT.
2. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_SRES PDU with one bit at random of the Authentication_Rsp inverted.
6. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type.
7. Perform either alternative 7A or 7B depending on the IUT's response.

   Alternative 7A (The IUT does not send an LMP_DETACH PDU):

   7A.1. The IUT sends an HCI_Change_Connection_Link_Key_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

   7A.2. The IUT and the Lower Tester continue connection establishment not authenticated.

   Alternative 7B (The IUT sends an LMP_DETACH PDU):

   7B.1. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).

7B.2. The IUT sends an HCI_Change_Connection_Link_Key_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
7B.3. The IUT and the Lower Tester disconnect.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_COMB_KEY PDU to the Lower Tester. The new link key matches the calculated link key.

  The IUT rejects the authentication after receiving a bad LMP_SRES PDU from the Lower Tester.

4.5.5 Secure Authentication procedures

Verify the Secure Authentication procedure.

LMP/AUT/BV-14-C [Secure Authentication, Responder (IUT) has link key, Initiator (Lower Tester) is Central]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Initiator, the IUT is the Responder, and the IUT has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

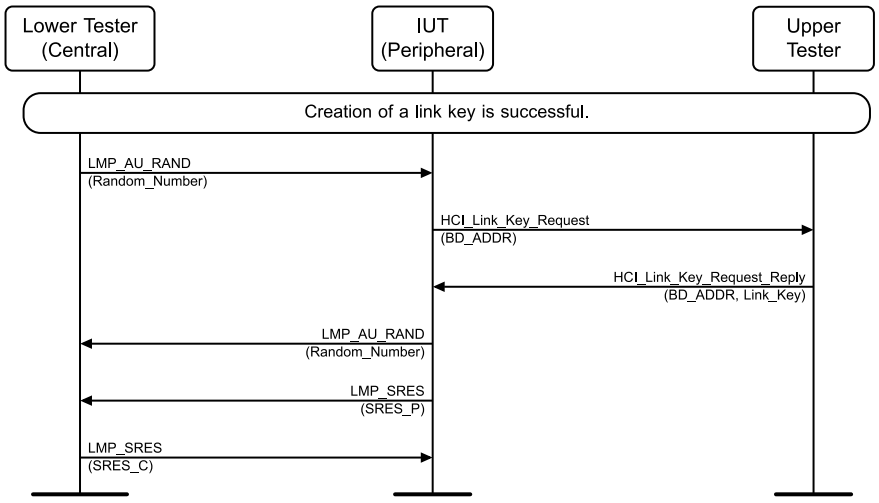


Figure 4.5-20: LMP/AUT/BV-14-C [Secure Authentication, Responder (IUT) has link key, Initiator (Lower Tester) is Central] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.

3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply event with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
6. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES_C.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 5, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct SRES.

LMP/AUT/BV-15-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Peripheral]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Initiator, the IUT is the Responder, and the IUT has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

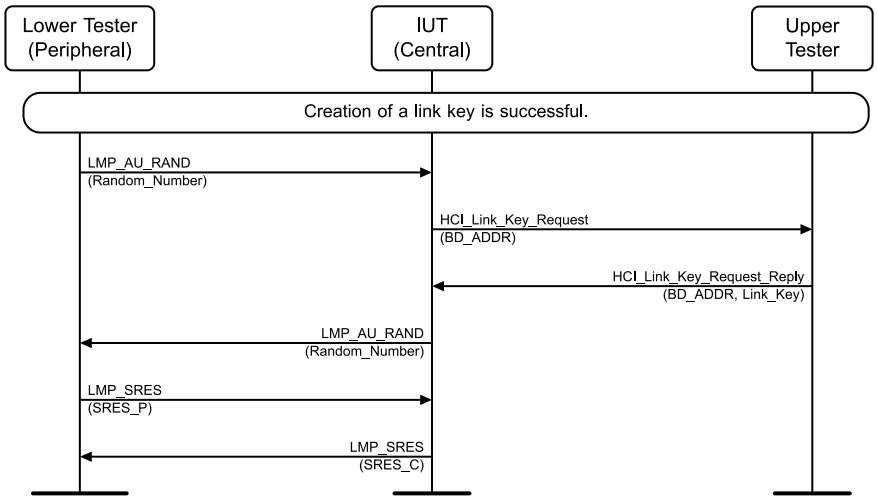


Figure 4.5-21: LMP/AUT/BV-15-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Peripheral] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.

3. The Upper Tester responds with an HCI_Link_Key_Request_Reply event to the IUT with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P.
6. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES_C.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 6, the IUT sends the LMP_SRES PDU containing the correct SRES upon reception of the LMP_SRES PDU from the Lower Tester.

LMP/AUT/BV-16-C [Secure Authentication, Responder (Lower Tester) has link Key, Initiator (IUT) is Central]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder, the IUT is the Initiator, and the IUT has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

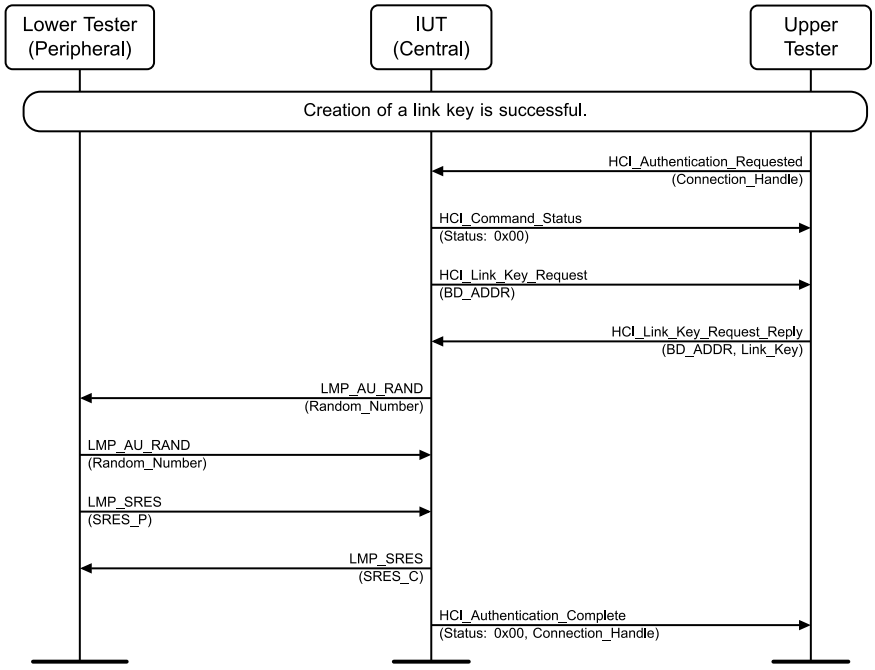


Figure 4.5-22: LMP/AUT/BV-16-C [Secure Authentication, Responder (Lower Tester) has link Key, Initiator (IUT) is Central] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds with an HCI_Link_Key_Request_Reply command to the IUT with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number and an LMP_SRES PDU with the SRES_P.
6. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_C.
7. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon receiving the link key from the Upper Tester.

  In Step 6, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct SRES_C upon reception of the LMP_SRES PDU from the Lower Tester.

LMP/AUT/BV-17-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder, the IUT is the Initiator, and the IUT has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

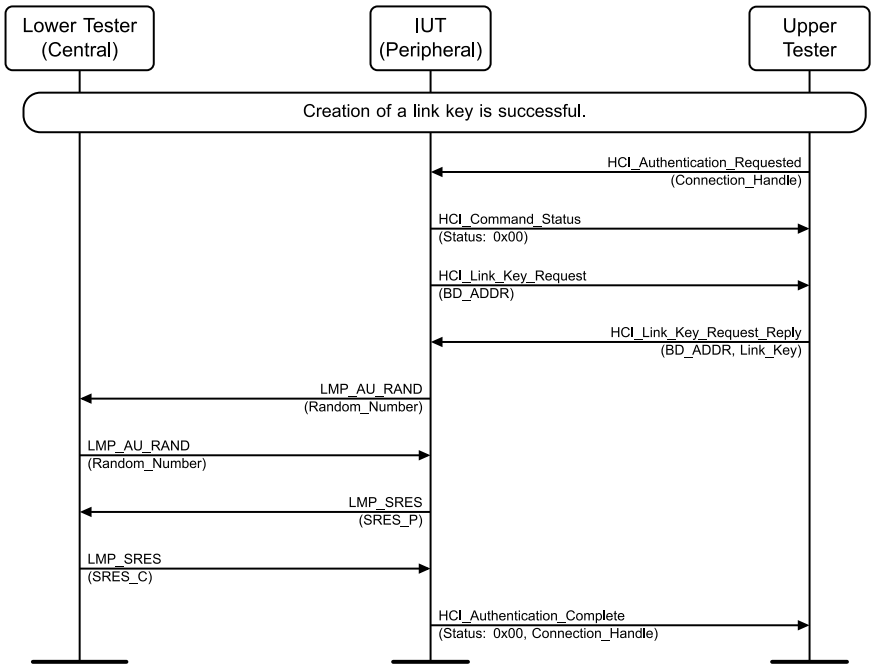


Figure 4.5-23: LMP/AUT/BV-17-C [Secure Authentication, Responder (Lower Tester) has link Key, Initiator (IUT) is Peripheral] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
6. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
7. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES_C.
8. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon receiving the link key from the Upper Tester.

  In Step 6, the IUT sends the LMP_SRES PDU containing the correct SRES_P upon reception of the LMP_AU_RAND PDU from the Lower Tester.

LMP/AUT/BV-18-C [Role switch during Secure Authentication before Authentication Response, Responder (IUT) has link key, Initiator (Lower Tester) is Central]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Initiator and initiates a role switch before it sends the authentication response, the IUT is the Responder, and the IUT has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

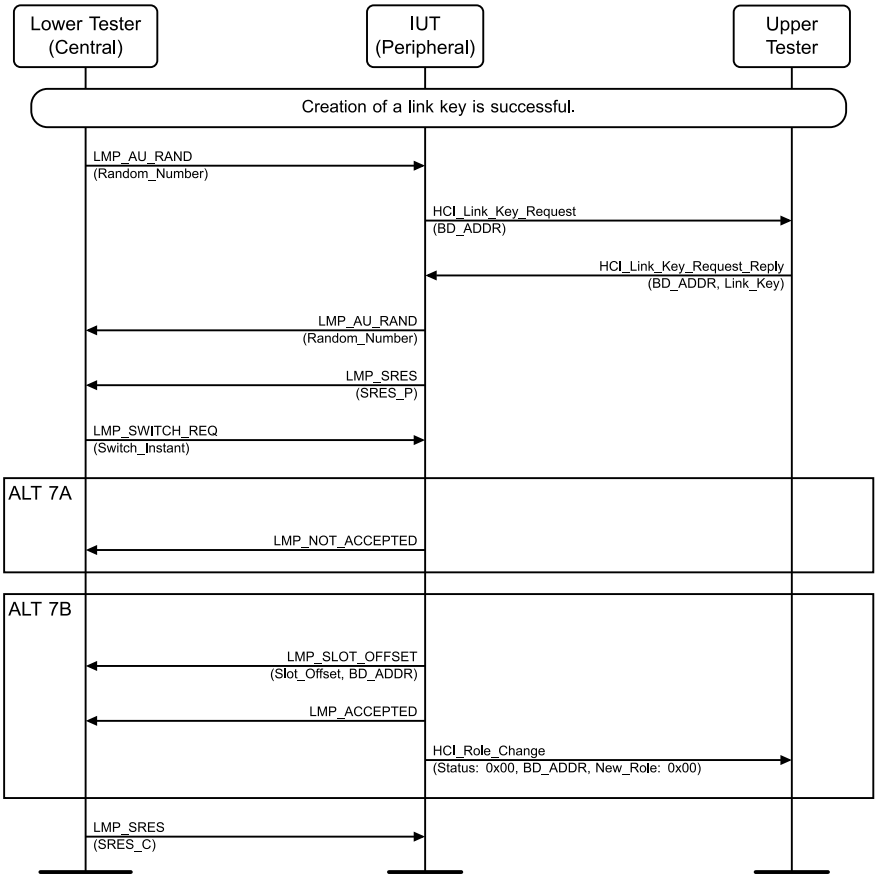


Figure 4.5-24: LMP/AUT/BV-18-C [Role switch during Secure Authentication before Authentication Response, Responder (IUT) has link key, Initiator (Lower Tester) is Central] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
6. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
7. Perform either alternative 7A or 7B depending on the IUT's response.

   Alternative 7A (The IUT denies the role switch):

   7A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   Alternative 7B (The IUT accepts the role switch):

   7B.1. The IUT responds to the Lower Tester with an LMP_SLOT_OFFSET PDU with the Slot_Offset and the BD_ADDR.

   7B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   7B.3. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).
8. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_C.

- Expected Outcome

  Pass verdict

  It is acceptable for the IUT to reject or to accept the role switch, as long as it continues and successfully completes the authentication.

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 5, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct SRES_P.

LMP/AUT/BV-19-C [Role switch during Secure Authentication before Authentication Response, Responder (IUT) has link key, Initiator (Lower Tester) is Peripheral]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Initiator and initiates a role switch before it sends the authentication response, the IUT is the Responder, and the IUT has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

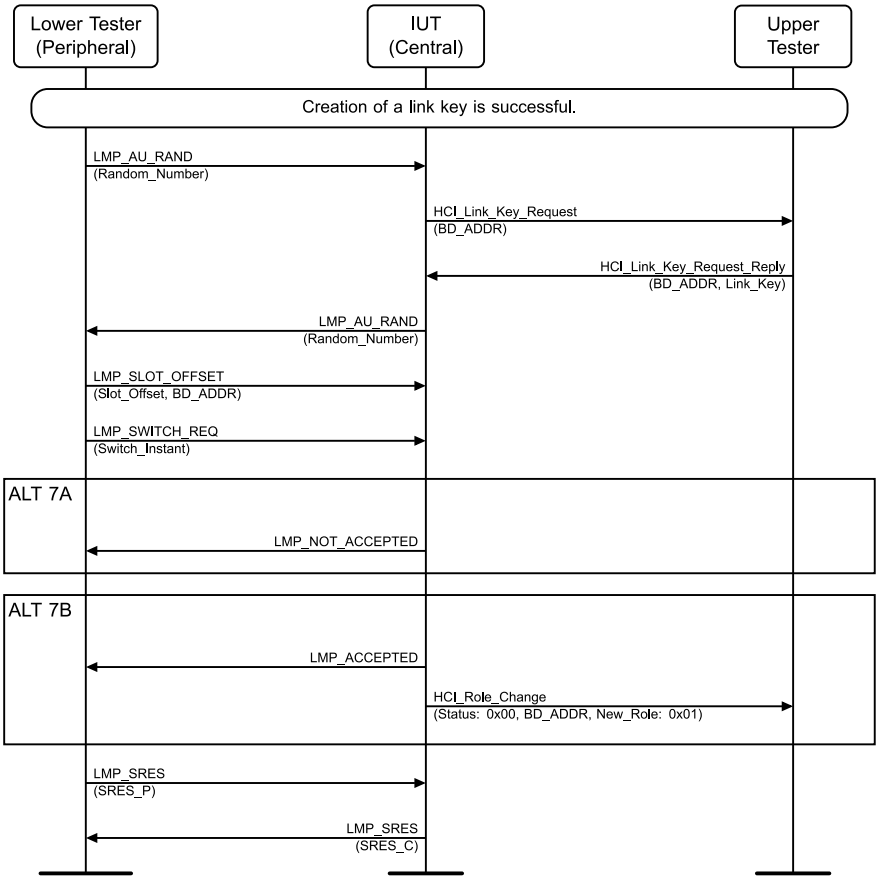


Figure 4.5-25: LMP/AUT/BV-19-C [Role switch during Secure Authentication before Authentication Response, Responder (IUT) has link key, Initiator (Lower Tester) is Peripheral] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and the BD_ADDR.
6. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
7. Perform either alternative 7A or 7B depending on the IUT's response.

   Alternative 7A (The IUT denies the role switch):

   7A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   Alternative 7B (The IUT accepts the role switch):

   7B.1. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   7B.2. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x01 (Peripheral).
8. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P.
9. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES_C.

- Expected Outcome

  Pass verdict

  It is acceptable for the IUT to reject or to accept the role switch, as long as it continues and successfully completes the authentication.

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 9, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response upon reception of the LMP_SRES PDU from the Lower Tester.

LMP/AUT/BV-20-C [Role switch during Secure Authentication before Authentication Response, Responder (Lower Tester) has link key, Initiator (IUT) is Central]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder and initiates a role switch before it sends the authentication response, the IUT is the Initiator, and the Lower Tester has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

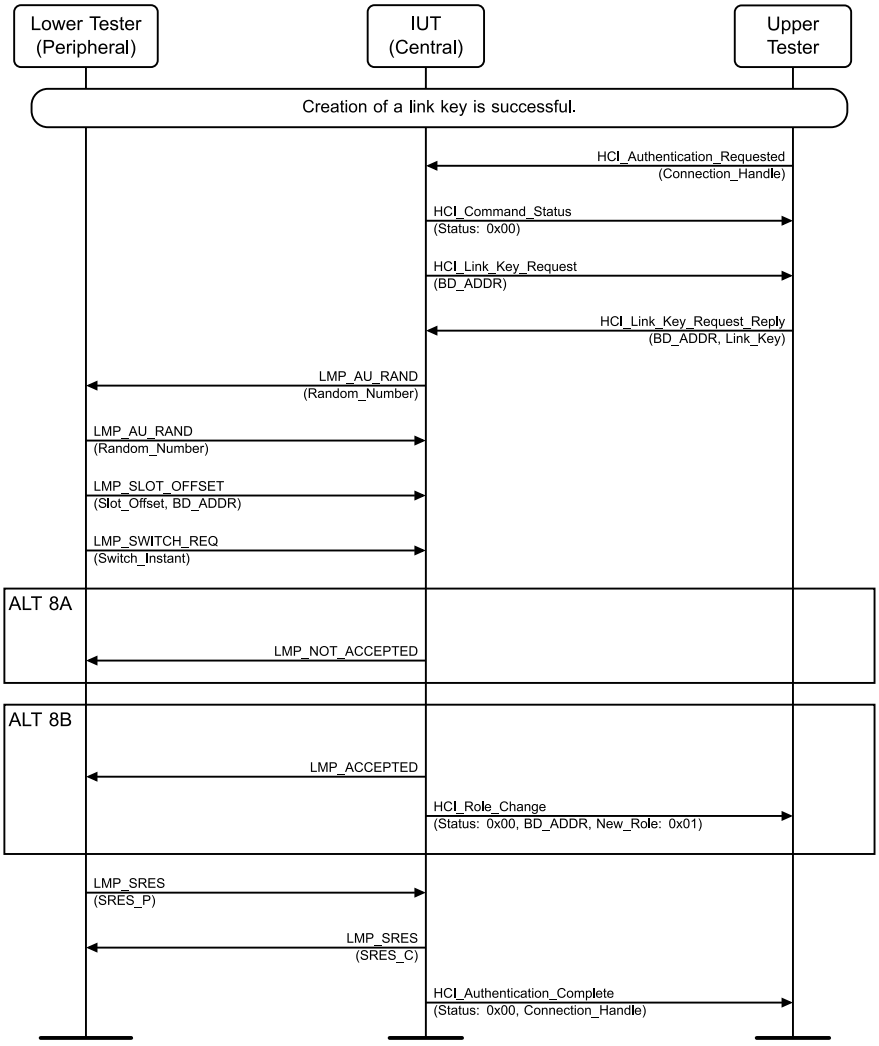


Figure 4.5-26: LMP/AUT/BV-20-C [Role switch during Secure Authentication before Authentication Response, Responder (Lower Tester) has link key, Initiator (IUT) is Central] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
6. The Lower Tester sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and the BD_ADDR.
7. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
8. Perform either alternative 8A or 8B depending on the IUT's response.
   Alternative 8A (The IUT rejects the role switch):
   - 8A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

Alternative 8B (The IUT accepts the role switch):

8B.1. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

8B.2. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x01 (Peripheral).

9. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P.
10. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES_C.
11. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  It is acceptable for the IUT to reject or to accept the role switch, as long as it continues and successfully completes the authentication.

  In Step 4, the IUT sends the LMP_AU_RAND PDU upon receiving the link key from the Upper Tester.

  In Step 10, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response upon reception of the LMP_SRES PDU from the Lower Tester.

  In Step 11, the IUT sends a successful HCI_Authentication_Complete event to the Upper Tester.

LMP/AUT/BV-21-C [Role switch during Secure Authentication before Authentication Response, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder and initiates a role switch before it sends the authentication response, the IUT is the Initiator, and the Lower Tester has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

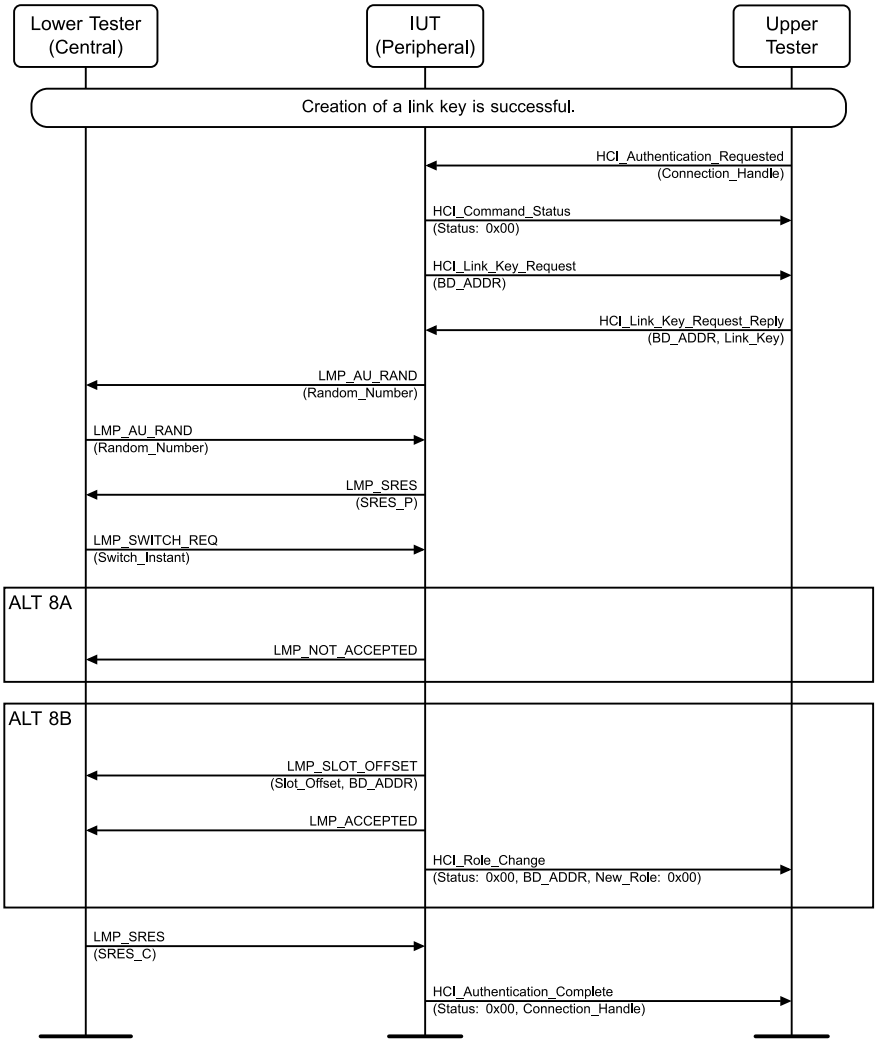


Figure 4.5-27: LMP/AUT/BV-21-C [Role switch during Secure Authentication before Authentication Response, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
6. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
7. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
8. Perform either alternative 8A or 8B depending on the IUT's response.

   Alternative 8A (The IUT rejects the role switch):

   8A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

Alternative 8B (The IUT accepts the role switch):

8B.1. The IUT responds to the Lower Tester with an LMP_SLOT_OFFSET PDU with the Slot_Offset and the BD_ADDR.

8B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

8B.3. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).

9. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_C.
10. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  It is acceptable for the IUT to reject or to accept the role switch, as long as it continues and successfully completes the authentication.

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon receiving the link key from the Upper Tester.

  In Step 6, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 10, the IUT sends a successful HCI_Authentication_Complete event to the Upper Tester.

LMP/AUT/BV-22-C [Role switch during Secure Authentication before Random Number, Responder (Lower Tester) has link key, Initiator (IUT) is Central]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder and initiates a role switch before it sends the random number, the IUT is the Initiator, and the Lower Tester has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

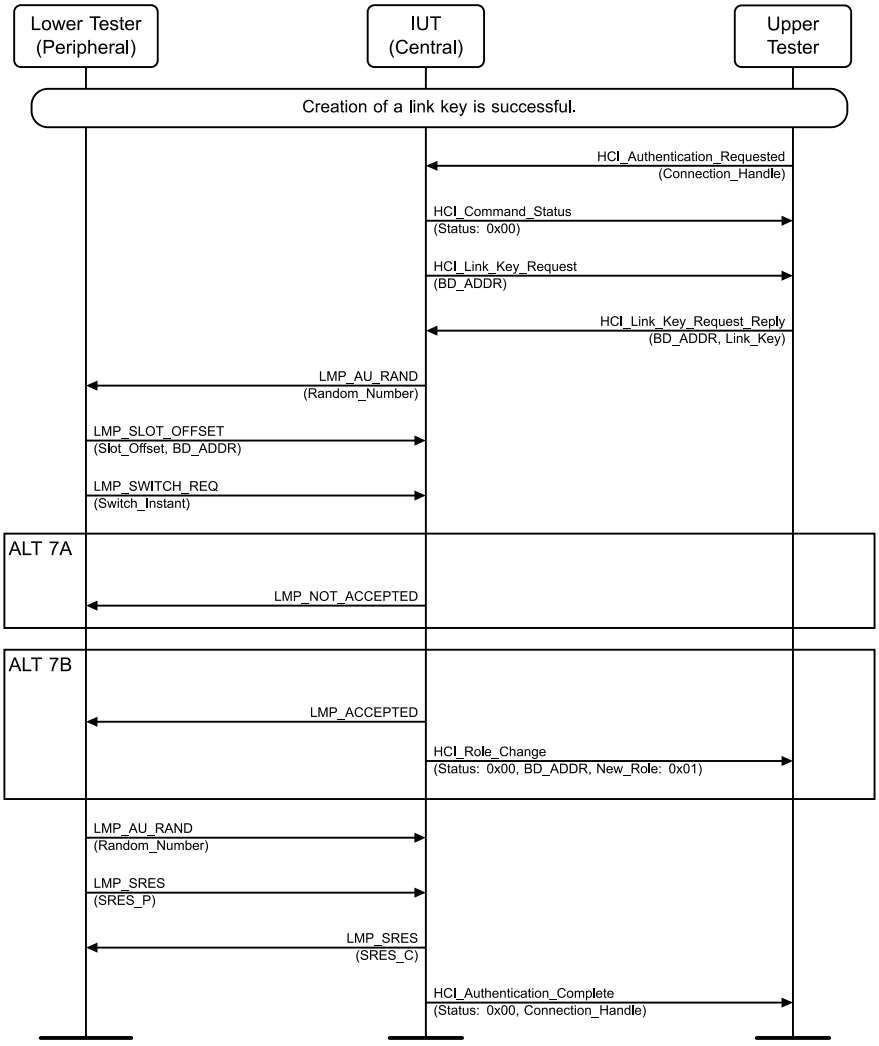


Figure 4.5-28: LMP/AUT/BV-22-C [Role switch during Secure Authentication before Random Number, Responder (Lower Tester) has link key, Initiator (IUT) is Central] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and the BD_ADDR.
6. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
7. Perform either alternative 7A or 7B depending on the IUT's response.
   Alternative 7A (The IUT rejects the role switch):
   7A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

Alternative 7B (The IUT accepts the role switch):

7B.1. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

7B.2. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x01 (Peripheral).

8. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
9. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P.
10. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES_C.
11. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  It is acceptable for the IUT to reject or to accept the role switch, as long as it continues and successfully completes the authentication.

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon receiving the link key from the Upper Tester.

  In Step 10, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response upon reception of the LMP_SRES PDU from the Lower Tester.

  In Step 11, the IUT sends a successful HCI_Authentication_Complete event to the Upper Tester.

LMP/AUT/BV-23-C [Role switch during Secure Authentication before Random Number, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder and initiates a role switch before it sends the random number, the IUT is the Initiator, and the Lower Tester has the link key.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

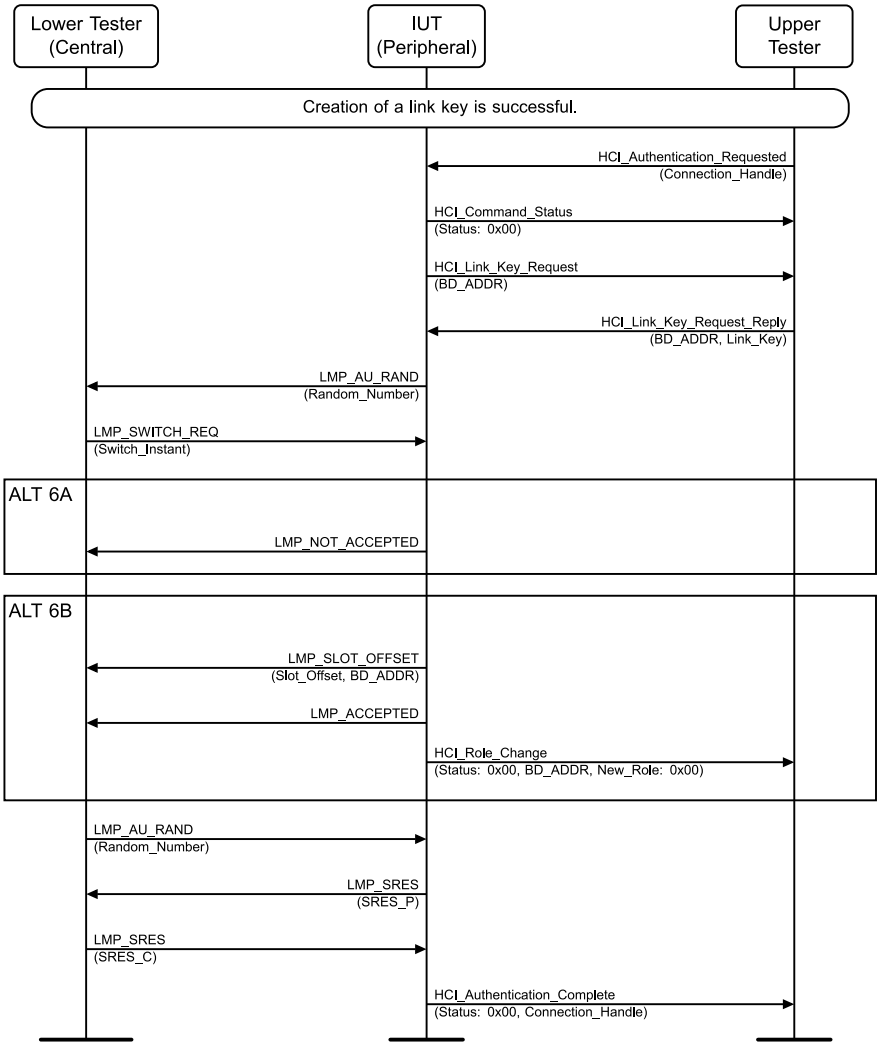


Figure 4.5-29: LMP/AUT/BV-23-C [Role switch during Secure Authentication before Random Number, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
6. Perform either alternative 6A or 6B depending on the IUT's response.

   Alternative 6A (The IUT rejects the role switch):

   6A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   Alternative 6B (The IUT accepts the role switch):

   6B.1. The IUT responds to the Lower Tester with an LMP_SLOT_OFFSET PDU with the Slot_Offset and the BD_ADDR.

   6B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

6B.3. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).

7. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
8. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
9. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES_C.
10. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  It is acceptable for the IUT to reject or to accept the role switch, as long as it continues and successfully completes the authentication.

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon receiving the link key from the Upper Tester.

  In Step 8, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 10, the IUT sends a successful HCI_Authentication_Complete event to the Upper Tester.

LMP/AUT/BI-02-C [Mistimed role switch during Secure Authentication, Responder (IUT) has link key, Initiator (Lower Tester) is Central]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Initiator, the IUT is the Responder, and the IUT has the link key. The Lower Tester initiates a role switch after the IUT sends the authentication response and the Lower Tester sends back the key that was sent by the IUT.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

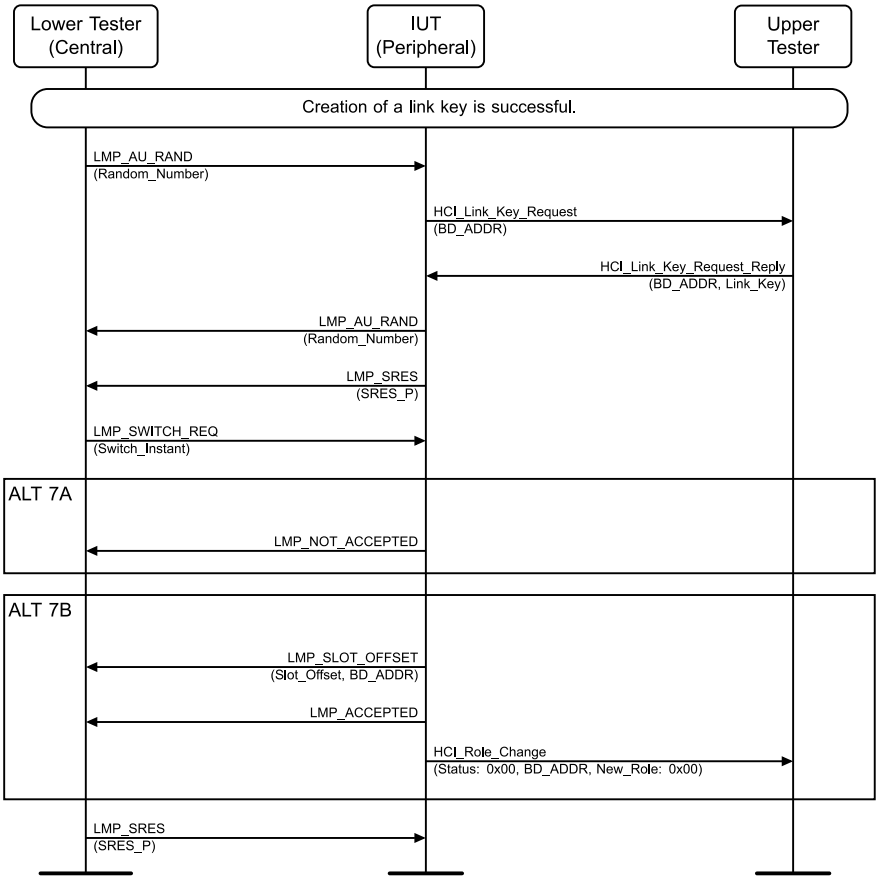


Figure 4.5-30: LMP/AUT/BI-02-C [Mistimed role switch during Secure Authentication, Responder (IUT) has link key, Initiator (Lower Tester) is Central] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
6. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
7. Perform either alternative 7A or 7B depending on the IUT's response.

   Alternative 7A (The IUT rejects the role switch):

   7A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   Alternative 7B (The IUT accepts the role switch):

   7B.1. The IUT responds to the Lower Tester with an LMP_SLOT_OFFSET PDU with the Slot_Offset and the BD_ADDR.

   7B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   7B.3. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).
8. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P that was sent by the IUT in Step 5.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 5, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response.

  After Step 8, the secure authentication procedure fails.

LMP/AUT/BI-03-C [Mistimed role switch during Secure Authentication, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral]

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder, the IUT is the Initiator, and the Lower Tester has the link key. The Lower Tester initiates a role switch after the IUT sends the authentication response and the Lower Tester sends back the key that was sent by the IUT.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

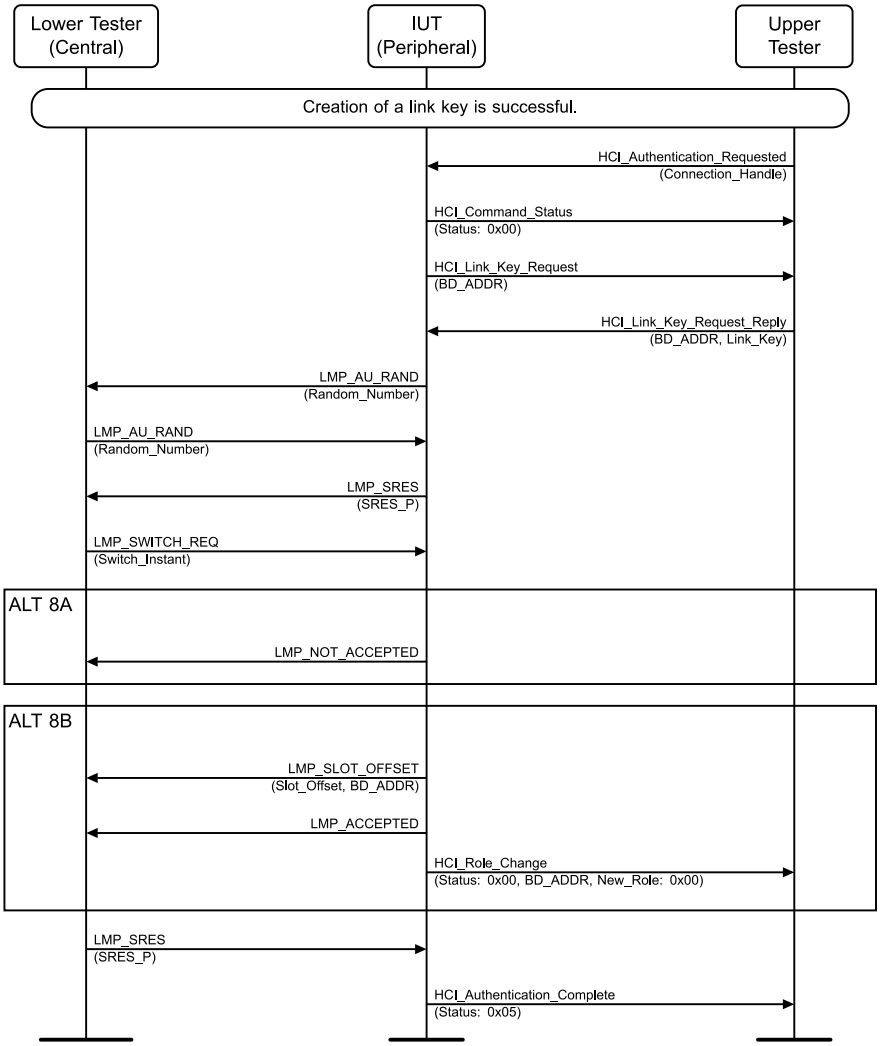


Figure 4.5-31: LMP/AUT/BI-03-C [Mistimed role switch during Secure Authentication, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
6. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
7. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
8. Perform either alternative 8A or 8B depending on the IUT's response.

   Alternative 8A (The IUT rejects the role switch):

   8A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   Alternative 8B (The IUT accepts the role switch):

   8B.1. The IUT responds to the Lower Tester with an LMP_SLOT_OFFSET PDU with the Slot_Offset and the BD_ADDR.

8B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

8B.3. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).

9. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P that was sent by the IUT in Step 6.
10. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with the Status set to 0x05 (Authentication Failure).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon receiving the link key from the Upper Tester.

  In Step 6, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 10, the IUT sends an HCI_Authentication_Complete event to the Upper Tester with the Status set to 0x05 (Authentication Failure).

4.5.5.1 Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Central – Rejects Bad Authentication Response

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Initiator, the IUT is the Responder, and the IUT has the link key. The IUT rejects a bad authentication response.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Case Configuration

Test Case	Detach
LMP/AUT/BV-30-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Central – Rejects Bad Authentication Response, Detach Optional]	ALT 7A or ALT 7B
LMP/AUT/BV-48-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Central – Rejects Bad Authentication Response, Detach Mandatory]	ALT 7B

Table 4.5-6: Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Central – Rejects Bad Authentication Response test cases

- Test Procedure

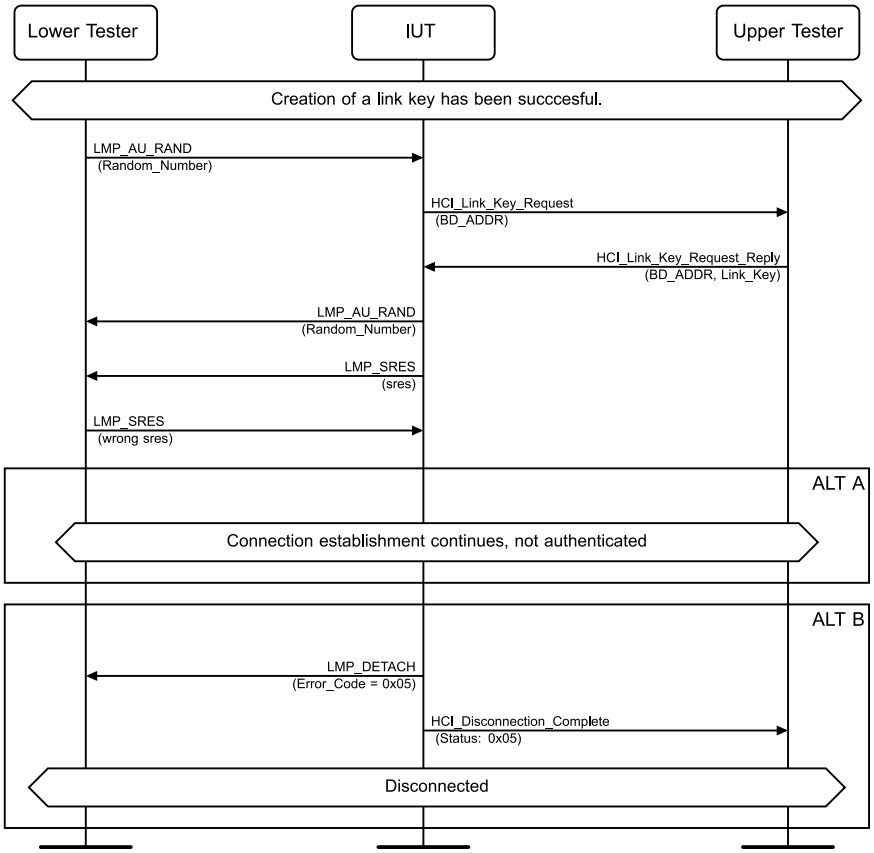


Figure 4.5-32: Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Central – Rejects Bad Authentication Response MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply event with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
6. The Lower Tester responds to the IUT with an LMP_SRES PDU with one bit at random of the SRES_C inverted.
7. Perform either alternative 7A or 7B as indicated in Table 4.5-6.

   Alternative 7A (Detach is optional):

   7A.1. The IUT and the Lower Tester continue connection establishment not authenticated.

   Alternative 7B (Detach is mandatory):

   7B.1. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).

   7B.2. The IUT sends an HCI_Disconnection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

   7B.3. The IUT and the Lower Tester disconnect.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  In Step 5, the IUT sends the LMP_SRES PDU to the Lower Tester containing the correct authentication response.

  The IUT rejects the authentication when receiving a bad LMP_SRES PDU from the Lower Tester via alternative 7A or 7B as indicated in Table 4.5-6.

4.5.5.2 Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Peripheral – Rejects Bad Authentication Response

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Initiator, the IUT is the Responder, and the IUT has the link key. The IUT rejects a bad authentication response.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Case Configuration

Test Case	Detach
LMP/AUT/BV-31-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Peripheral – Rejects Bad Authentication Response, Detach Optional]	ALT 6A or ALT 6B
LMP/AUT/BV-49-C [Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Peripheral – Rejects Bad Authentication Response, Detach Mandatory]	ALT 6B

Table 4.5-7: Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Peripheral – Rejects Bad Authentication Response test cases

- Test Procedure

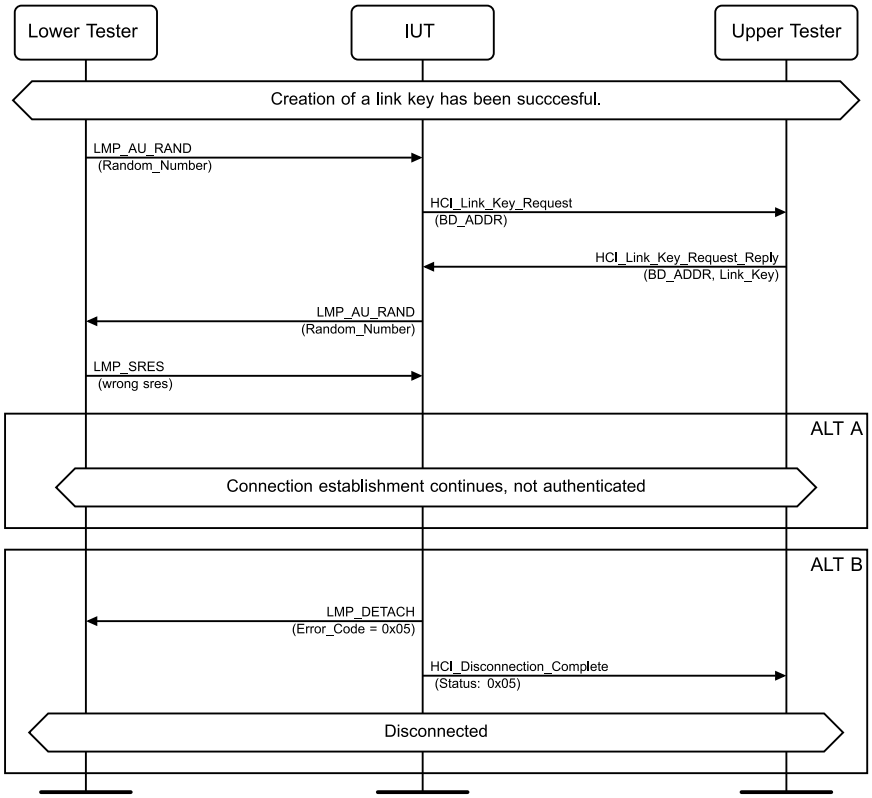


Figure 4.5-33: Secure Authentication, Responder (IUT) has Link Key, Initiator (Lower Tester) is Peripheral – Rejects Bad Authentication Response MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply event with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_SRES PDU with one bit at random of the SRES_P inverted.
6. Perform either alternative 6A or 6B as indicated in Table 4.5-7.

   Alternative 6A (Detach is optional):

   6A.1. The IUT and the Lower Tester continue connection establishment not authenticated.

   Alternative 6B (Detach is mandatory):

   6B.1. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).

   6B.2. The IUT sends an HCI_Disconnection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

   6B.3. The IUT and the Lower Tester disconnect.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon reception of the LMP_AU_RAND PDU from the Lower Tester.

  The IUT rejects the authentication when receiving a bad LMP_SRES PDU from the Lower Tester via alternative 6A or 6B as indicated in Table 4.5-7.

4.5.5.3 Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Central – Rejects Bad Authentication Response

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder, the IUT is the Initiator, and the IUT has the link key. The IUT rejects a bad authentication response.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Case Configuration

Test Case	Detach
LMP/AUT/BV-32-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Central – Rejects Bad Authentication Response, Detach Optional]	ALT 6A or ALT 6B
LMP/AUT/BV-50-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Central – Rejects Bad Authentication Response, Detach Mandatory]	ALT 6B

Table 4.5-8: Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Central – Rejects Bad Authentication Response test cases

- Test Procedure

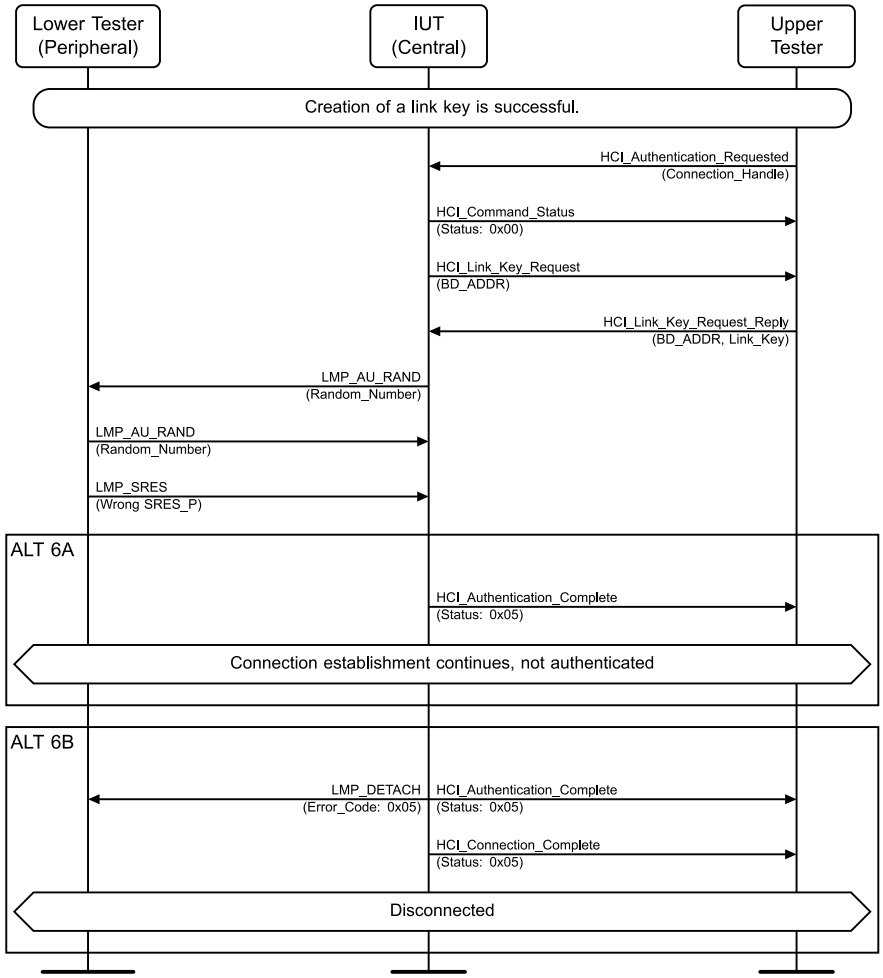


Figure 4.5-34: Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Central – Rejects Bad Authentication Response MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number and an LMP_SRES PDU with one bit at random of the SRES_P inverted.
6. Perform either alternative 6A or 6B as indicated in Table 4.5-8.

   Alternative 6A (Detach is optional):

   6A.1. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with the Status set to 0x05 (Authentication Failure).

   6A.2. The IUT and the Lower Tester continue connection establishment not authenticated.

   Alternative 6B (Detach is mandatory):

   6B.1. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).

   6B.2. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

6B.3. The IUT sends an HCI_Connection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).
6B.4. The IUT and the Lower Tester disconnect.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester upon receiving the link key from the Upper Tester.

  The IUT rejects the authentication when receiving a bad LMP_SRES PDU from the Lower Tester via alternative 6A or 6B as indicated in Table 4.5-8.

4.5.5.4 Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral – Rejects Bad Authentication Response

- Test Purpose

  Verify the Secure Authentication procedure when the Lower Tester is the Responder, the IUT is the Initiator, and the IUT has the link key. The IUT rejects a bad authentication response.

- Reference

  [1] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Case Configuration

Test Case	Detach
LMP/AUT/BV-33-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral – Rejects Bad Authentication Response, Detach Optional]	Optional
LMP/AUT/BV-51-C [Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral – Rejects Bad Authentication Response, Detach Mandatory]	Mandatory

Table 4.5-9: Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral – Rejects Bad Authentication Response test cases

- Test Procedure

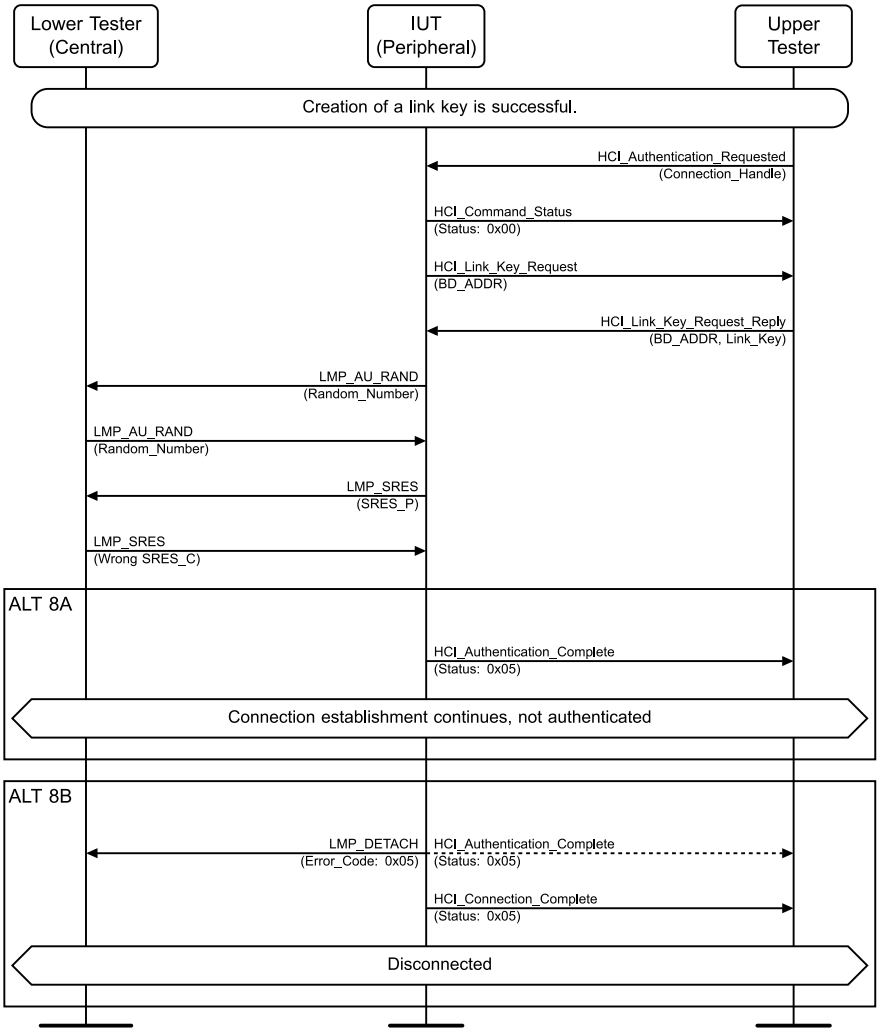


Figure 4.5-35: Secure Authentication, Responder (Lower Tester) has Link Key, Initiator (IUT) is Peripheral – Rejects Bad Authentication Response MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
6. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
7. The Lower Tester responds to the IUT with an LMP_SRES PDU with one bit at random of the SRES_C inverted.
8. Perform either alternative 8A or 8B as indicated in Table 4.5-9.
   Alternative 8A (Detach is optional):
   - 8A.1. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with the Status set to 0x05 (Authentication Failure).
   - 8A.2. The IUT and the Lower Tester continue connection establishment not authenticated.

Alternative 8B (Detach is mandatory):

8B.1. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x05 (Authentication Failure).

8B.2. Optionally, the IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

8B.3. The IUT sends an HCI_Connection_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

8B.4. The IUT and the Lower Tester disconnect.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU upon receiving the link key from the Upper Tester.

  The IUT rejects the authentication when receiving a bad LMP_SRES PDU from the Lower Tester via alternative 8A or 8B depending on whether Detach is optional or mandatory as indicated in Table 4.5-9.

LMP/AUT/BV-35-C [Secure Authentication, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral]

- Test Purpose

  Verify that the IUT properly handles the Lower Tester sending the Signed Response immediately after sending an LMP_AU_RAND PDU.

- Reference

  [1] 4.2.1.4

  [13] 5.0

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The Lower Tester is the Responder, the IUT is the Initiator and has the link key.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

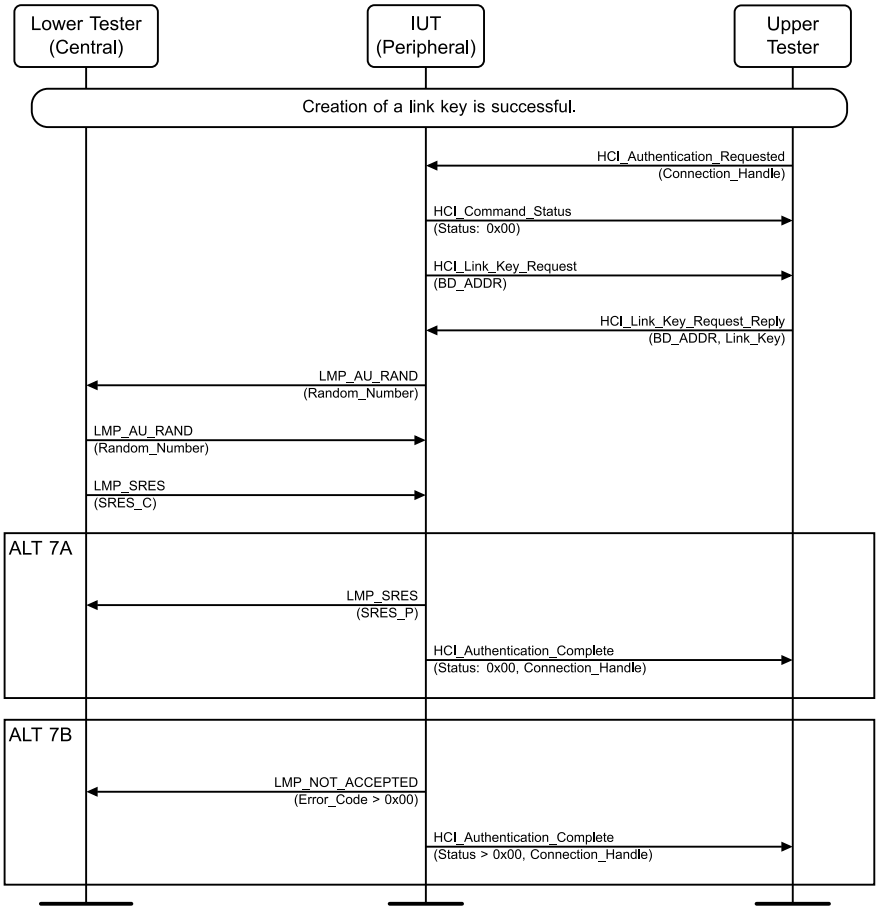


Figure 4.5-36: LMP/AUT/BV-35-C [Secure Authentication, Responder (Lower Tester) has link key, Initiator (IUT) is Peripheral] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
6. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_C.
7. Perform either alternative 7A or 7B depending on the IUT's response.

   Alternative 7A (The IUT sends an LMP_SRES PDU to the Lower Tester):

   7A.1. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES_P.

   7A.2. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester.

   Alternative 7B (The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester):

   7B.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with Error_Code set to a value greater than 0x00.

   7B.2. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status set to a value greater than 0x00.

- Expected Outcome

  Pass verdict

  In alternative 7A, the IUT accepts the LMP_SRES PDU from the Lower Tester and sends an LMP_SRES PDU to the Lower Tester and a successful HCI_Authentication_Complete event to the Upper Tester.

  In alternative 7B, the IUT rejects the LMP_SRES PDU from the Lower Tester and sends an LMP_NOT_ACCEPTED PDU with a valid error code to the Lower Tester and an HCI_Authentication_Complete event to the Upper Tester with a valid error Status value and Connection_Handle.

LMP/AUT/BI-06-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Central, Reject Role Switch]

- Test Purpose

  Verify that the IUT either rejects or disconnects the peer when receiving a role switch request during the Authentication process.

- Reference

  [14] 4.2.1.4, 4.4.2

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Peripheral and the Lower Tester is the Central.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

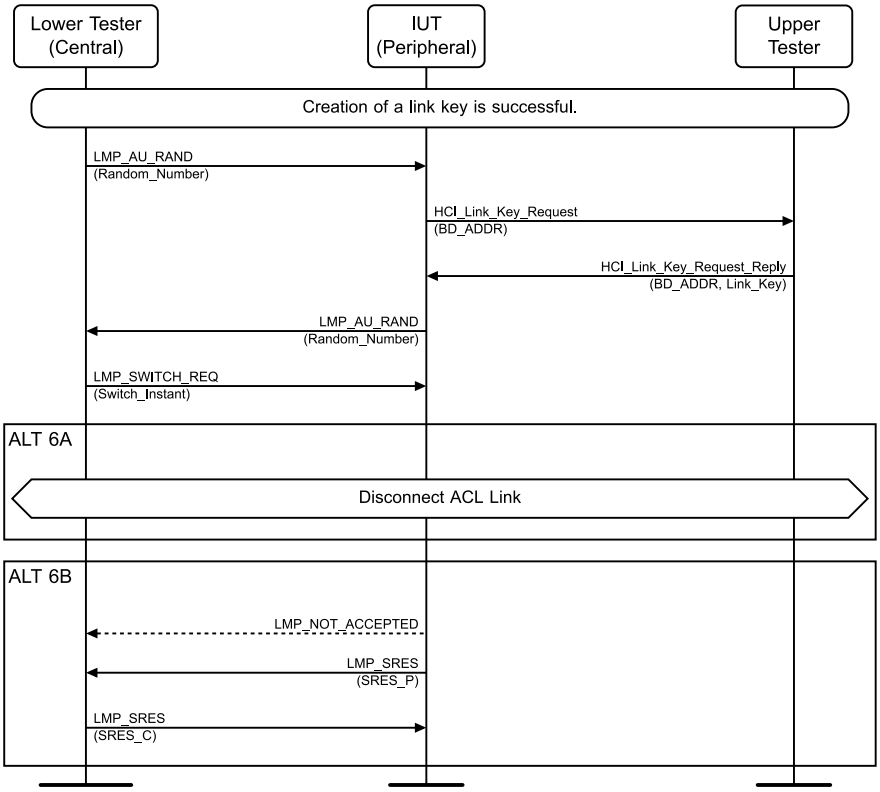


Figure 4.5-37: LMP/AUT/BI-06-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Central, Reject Role Switch] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply event with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
6. Perform either alternative 6A or 6B depending on the IUT's response.

   Alternative 6A (The IUT disconnects the ACL Link):

   6A.1. The IUT disconnects the ACL Link.

   Alternative 6B (The IUT rejects the role switch):

   6B.1. Optionally, the IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.

   6B.2. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.

   6B.3. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES_C.

- Expected Outcome

  Pass verdict

  In alternative 6A, the IUT disconnects the ACL Link.

  In alternative 6B, the IUT optionally sends the LMP_NOT_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester. The IUT sends the LMP_SRES PDU to the Lower Tester containing the correct SRES_P.

LMP/AUT/BI-07-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Peripheral, Reject Role Switch]

- Test Purpose

  Verify that the IUT either rejects or disconnects the peer when receiving a role switch request during the Authentication process.

- Reference

  [14] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and “Secure Simple Pairing P-256” default settings.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester have already created a link key.

- Test Procedure

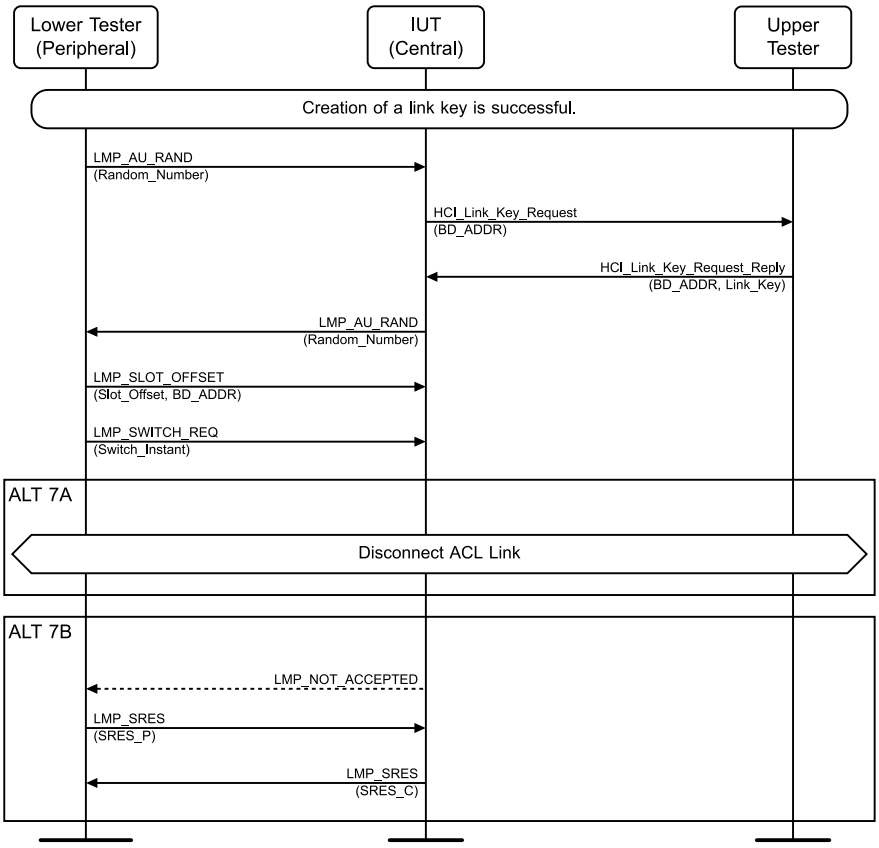


Figure 4.5-38: LMP/AUT/BI-07-C [Secure Authentication, Responder (IUT) has link Key, Initiator (Lower Tester) is Peripheral, Reject Role Switch] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply event with the BD_ADDR and Link_Key.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.

5. The Lower Tester sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and the BD_ADDR.
6. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
7. Perform either alternative 7A or 7B depending on the IUT's response.
   Alternative 7A (The IUT disconnects the ACL Link):
   7A.1. The IUT disconnects the ACL Link.
   Alternative 7B (The IUT rejects the role switch):
   7B.1. Optionally, the IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
   7B.2. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
   7B.3. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES_C.

- Expected Outcome

  Pass verdict

  In alternative 7A, the IUT disconnects the ACL Link.

  In alternative 7B, the IUT optionally sends the LMP_NOT_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester. The IUT sends the LMP_SRES PDU to the Lower Tester containing the correct SRES_P upon reception of the LMP_SRES PDU from the Lower Tester.

LMP/AUT/BV-37-C [Secure Authentication of Previously Authenticated or Stored Link Key]

- Test Purpose

  Verify that the IUT performs the Secure Authentication procedure when requested by the Host on a secure active connection when the link key has been previously authenticated or a link key is stored on the IUT.

- Reference

  [1] 4.2.1.4

  [7] 7.1.15

- Initial Condition

  - See the "Baseband assumptions" section.
  - The IUT is the Initiator and has a link key associated with the Responder. The Lower Tester is the Responder.
  - An ACL connection has been established.
  - The Lower Tester and the IUT have performed Secure Simple Pairing with Secure Connections.
  - Authentication has been previously performed, and encryption has been enabled.
  - The Lower Tester has the Secure Connections (Controller Support) LMP feature bit set. The Upper Tester sets the Secure Connections (Host Support) LMP feature bit. The Lower Tester performs a feature exchange.

- Test Procedure

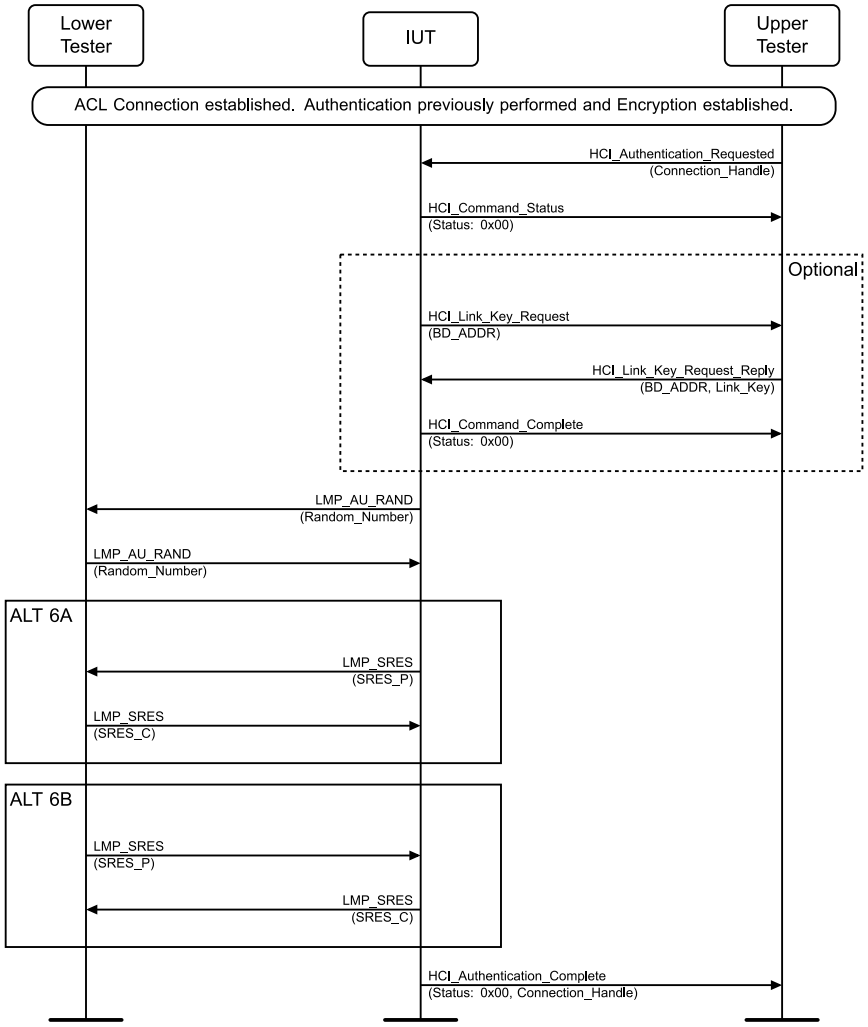


Figure 4.5-39: LMP/AUT/BV-37-C [Secure Authentication of Previously Authenticated or Stored Link Key] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. If Step 2 occurs, the Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
6. Perform either alternative 6A or 6B depending on the role of the IUT.
   Alternative 6A (The IUT is the Peripheral):
   - 6A.1. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
   - 6A.2. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES_C.

   Alternative 6B (The IUT is the Central):
   - 6B.1. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P.
   - 6B.2. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES_C.
7. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_AU_RAND PDU to the Lower Tester containing the Random_Number parameter.

  In Step 6, the IUT sends the LMP_SRES PDU to the Lower Tester containing the SRES parameter.

  In Step 7, the IUT sends a successful HCI_Authentication_Complete event to the Upper Tester after receiving an LMP_SRES PDU with the Authentication_Rsp from the Lower Tester.

4.5.6 Legacy Authentication procedures

4.5.6.1 Mutual Legacy Authentication, Initiator

- Test Purpose

  Verify that the IUT can handle mutual legacy authentication as the Initiator.

- Reference

  [1] 4.2.1.1

- Initial Condition

  - The IUT and the Lower Tester have previously established a link key and have just created a new connection with the IUT in the role specified in Table 4.5-10. The Lower Tester does not support Secure Connections.

- Test Case Configuration

Test Case	IUT Role
LMP/AUT/BV-38-C [Mutual Legacy Authentication, Initiator, Central]	Central
LMP/AUT/BV-39-C [Mutual Legacy Authentication, Initiator, Peripheral]	Peripheral

Table 4.5-10: Mutual Legacy Authentication, Initiator test cases

- Test Procedure

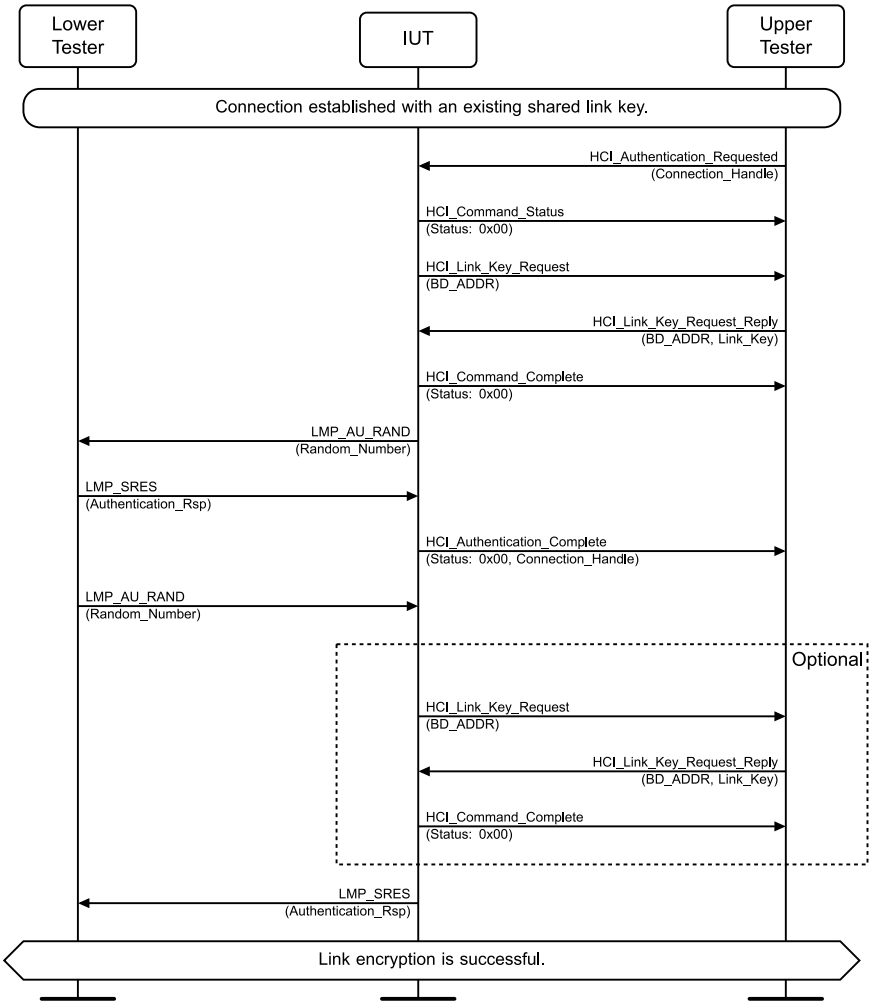


Figure 4.5-40: Mutual Legacy Authentication, Initiator MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.
6. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.
7. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
8. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
9. If Step 8 occurs, the Upper Tester responds to the IUT with an HCI_Link_Key_Request_Reply command with the BD_ADDR and Link_Key and receives a successful HCI_Command_Complete event in response.
10. The IUT sends an LMP_SRES PDU to the IUT with the Authentication_Rsp.

- Expected Outcome

  Pass verdict

  In Step 10, the IUT responds to the LMP_AU_RAND PDU with an LMP_SRES PDU containing a correct Authentication_Rsp value.

  Link encryption is successfully enabled using the shared link key.

  Fail verdict

  The IUT sends another LMP_AU_RAND PDU after responding with the LMP_SRES PDU.

LMP/AUT/BV-41-C [Delete Stored Link Key, In Connection]

- Test Purpose

  Verify that the IUT deletes the link key after the connection is disconnected.

- Reference

  [1] 4.2.1

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the Initiator. The Lower Tester is the Responder.
  - The Lower Tester does not support Secure Simple Pairing.
  - The IUT has at least one key in its key store.
  - The IUT does not contain a key for the Lower Tester’s BD_ADDR in its key store.
  - If the IUT supports authentication before connection completion, then the Upper Tester sends the HCI_Write_Authentication_Enable command with Enabled set to 0x01 before starting the connection.

- Test Procedure

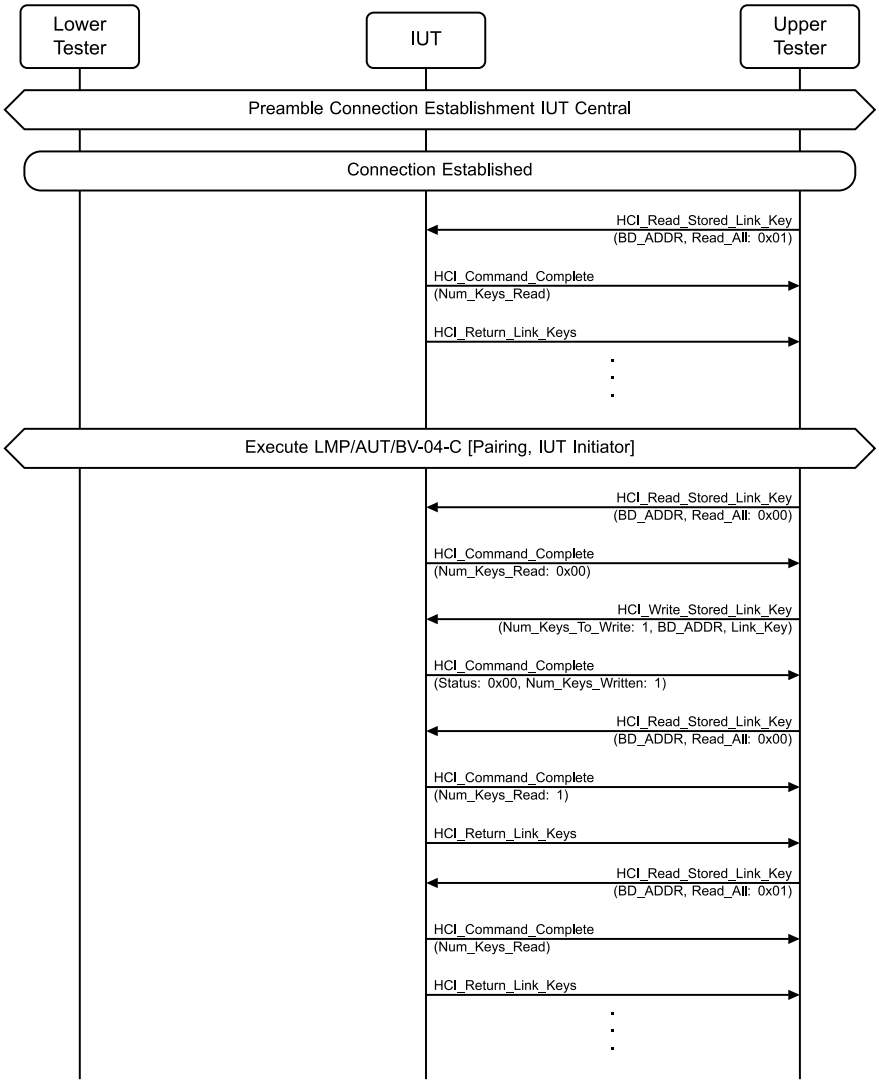


Figure 4.5-41: LMP/AUT/BV-41-C [Delete Stored Link Key, In Connection] MSC – Page 1 of 2

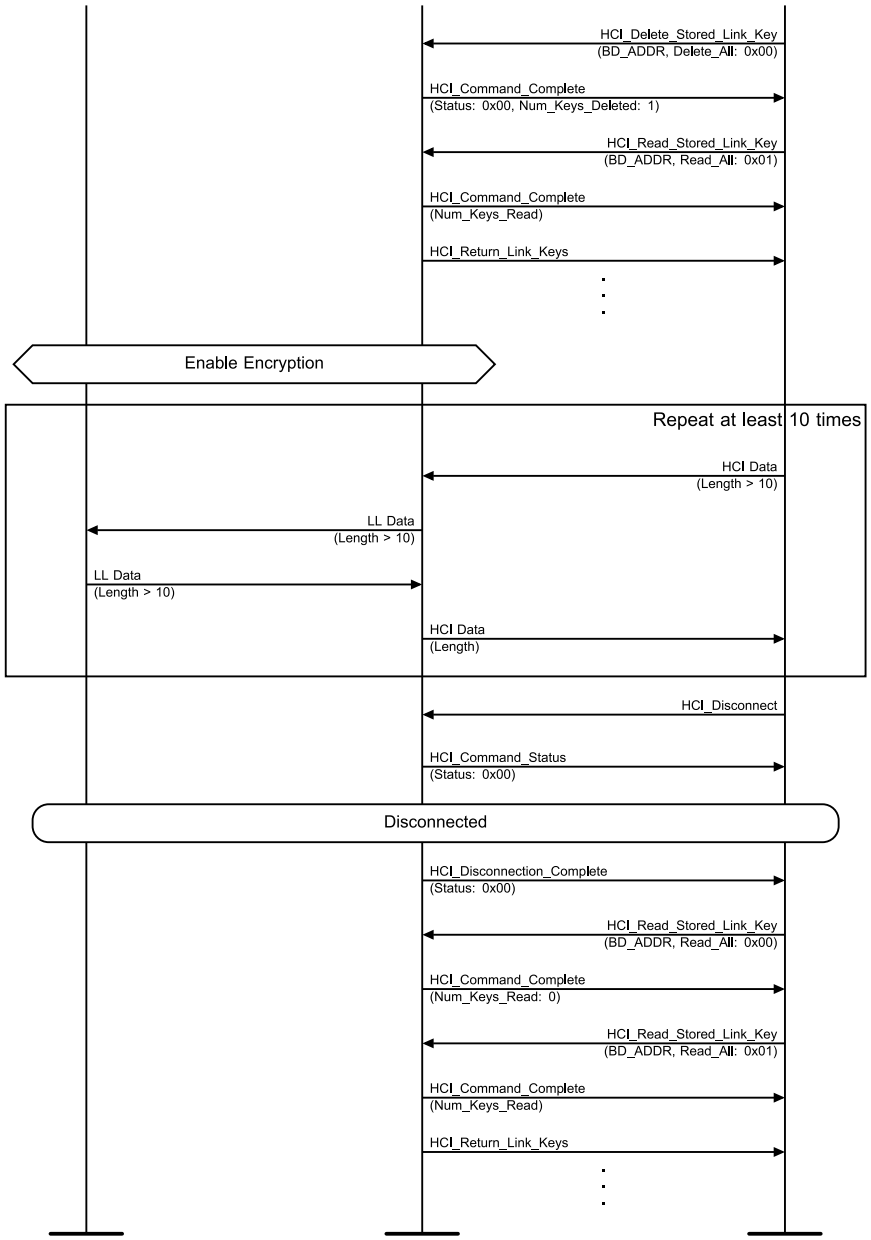


Figure 4.5-42: LMP/AUT/BV-41-C [Delete Stored Link Key, In Connection] MSC – Page 2 of 2

1. Run the “Connection Establishment IUT Central” preamble.
2. The Upper Tester sends an HCI_Read_Stored_Link_Key command to the IUT with a random BD_ADDR and Read_All set to 0x01.
3. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Read. Num_Keys_Read is stored to be used in the following steps.
4. The IUT sends one or more HCI_Return_Link_Keys events to the Upper Tester. The Lower Tester’s BD_ADDR does not appear in any of the events. The number of unique addresses in the events equals the value of Num_Keys_Read in Step 3.
5. Wait for 40 slot pairs after the last HCI_Return_Link_Keys event to ensure that no more HCI_Return_Link_Keys events are sent to the Upper Tester.
6. Execute the test procedure of Pairing, IUT Initiator.
7. The Upper Tester sends an HCI_Read_Stored_Link_Key command to the IUT with BD_ADDR set to the BD_ADDR of the Lower Tester and Read_All set to 0x00.

8. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Read set to 0.
9. Wait for 40 slot pairs after the HCI_Command_Complete event to ensure that no HCI_Return_Link_Keys events are sent to the Upper Tester.
10. The Upper Tester sends an HCI_Write_Stored_Link_Key command to the IUT with Num_Keys_To_Write set to 1, BD_ADDR[0] set to the BD_ADDR of the Lower Tester, and Link_Key[0] set to the link key created in Step 6.
11. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Written = 1.
12. The Upper Tester sends an HCI_Read_Stored_Link_Key command to the IUT with BD_ADDR set to the BD_ADDR of the Lower Tester and Read_All set to 0x00.
13. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Read = 1.
14. The IUT sends exactly one HCI_Return_Link_Keys event to the Upper Tester with Num_Keys set to 1 and BD_ADDR[0] set to the Lower Tester's BD_ADDR.
15. Wait for 40 slot pairs after the last HCI_Return_Link_Keys event to ensure that no more HCI_Return_Link_Keys events are sent.
16. The Upper Tester sends an HCI_Read_Stored_Link_Key command to the IUT with a random BD_ADDR and Read_All set to 0x01.
17. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Read set to a value one greater than the value in Step 3.
18. The IUT sends one or more HCI_Return_Link_Keys events to the Upper Tester. The Lower Tester's BD_ADDR appears in exactly one of the events. The number of unique addresses in the events equals the value of Num_Keys_Read in Step 17.
19. Wait for 40 slot pairs after the last HCI_Return_Link_Keys event to ensure that no more HCI_Return_Link_Keys events are sent.
20. The Upper Tester sends an HCI_Delete_Stored_Link_Key command to the IUT with BD_ADDR set to the BD_ADDR of the Lower Tester and Delete_All set to 0x00.
21. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Deleted = 1.
22. The Upper Tester sends an HCI_Read_Stored_Link_Key command to the IUT with a random BD_ADDR and Read_All set to 0x01.
23. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Read the same as the Num_Keys_Read from Step 3.
24. The IUT sends one or more HCI_Return_Link_Keys events to the Upper Tester. The Lower Tester's BD_ADDR does not appear in any of the events. The number of unique addresses in the events equals the value of Num_Keys_Read in Step 3.
25. Wait for 40 slot pairs after the last HCI_Return_Link_Keys event to ensure that no more HCI_Return_Link_Keys events are sent.
26. The IUT and the Lower Tester enable encryption using the link key created in Step 6.
27. The Upper Tester and the Lower Tester send each other at least 10 packets containing at least 10 octets of data each.
28. The Upper Tester sends an HCI_Disconnect command to the IUT with Connection_Handle set to the current ACL connection and receives a successful HCI_Command_Status in response.
29. The IUT and the Lower Tester are disconnected.
30. The IUT sends an HCI_Disconnection_Complete event to the Upper Tester.
31. The Upper Tester sends an HCI_Read_Stored_Link_Key command to the IUT with BD_ADDR set to the BD_ADDR of the Lower Tester and Read_All set to 0x00.
32. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Read set to 0.

33. Wait for 40 slot pairs after the HCI_Command_Complete event to ensure that no HCI_Return_Link_Keys events are sent to the Upper Tester.
34. The Upper Tester sends an HCI_Read_Stored_Link_Key command to the IUT with a random BD_ADDR and Read_All set to 0x01.
35. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Num_Keys_Read equal to the value in Step 3.
36. The IUT sends one or more HCI_Return_Link_Keys events to the Upper Tester. The Lower Tester's BD_ADDR does not appear in any of the events. The number of unique addresses in the events equals the Num_Keys_Read in Step 3.
37. Wait for 40 slot pairs after the last HCI_Return_Link_Keys event to ensure that no more HCI_Return_Link_Keys events are sent.

- Expected Outcome

  Pass verdict

  In Steps 23 and 35, the value of Num_Keys_Read is the same as in Step 3.

  In Steps 8 and 32, Num_Keys_Read is set to 0.

  In Step 13, Num_Keys_Read is set to 1.

  In Step 17, the Num_Keys_Read is one more than the value from Step 3.

  In Steps 4, 24, and 36, the number of unique addresses equals the value of Num_Keys_Read in Step 3, and the Lower Tester's BD_ADDR does not appear.

  In Step 14, the only BD_ADDR sent to the Upper Tester is that of the Lower Tester.

  In Step 18, the number of unique addresses equals the value of Num_Keys_Read in Step 17, and the Lower Tester's BD_ADDR is included in the list.

  In Step 27, the data is correctly encrypted and decrypted.

- Notes

  Possible interaction might be needed on the IUT. HCI commands might be needed for PIN, Key, etc. It is implementation dependent.

  The configuration of the IUT and the Lower Tester will decide which LMP commands to use when creating the link key.

  It must be verified that if both the Lower Tester and the IUT are configured to use a combination Key, a mutual authentication has to be carried out.

  The initiation of the pairing procedure might be taken on an already established link.

LMP/AUT/BV-42-C [Repeated Authentication Failure]

- Test Purpose

  Verify that the IUT correctly handles repeated failed authentication attempts.

- Reference

  [13] 4.2.1.2

- Initial Condition

  - See the "Baseband assumptions" section.

- Test Procedure

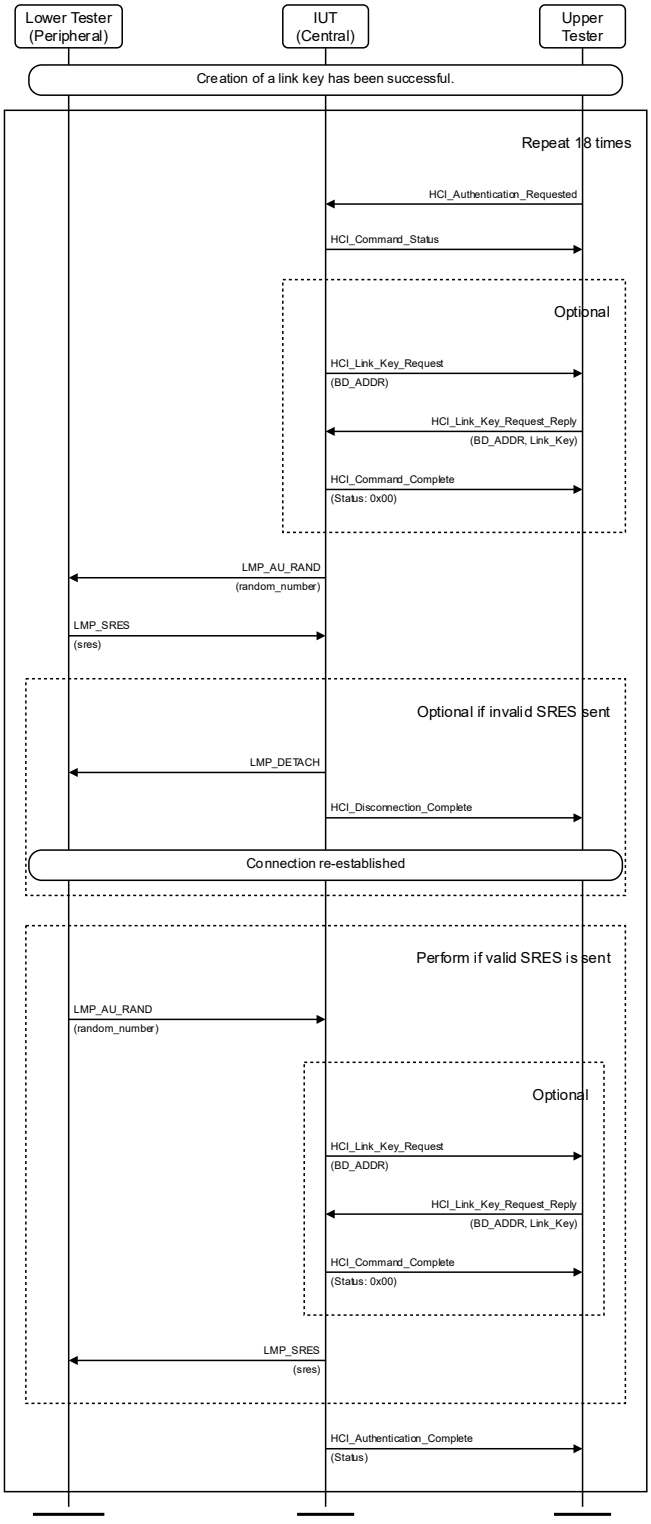


Figure 4.5-43: LMP/AUT/BV-42-C [Repeated Authentication Failure] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT and receives a successful HCI_Command_Status in return.
2. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester. If it does, then the Upper Tester responds with an HCI_Link_Key_Request_Reply containing the stored link key sent by the IUT and receives a successful HCI_Command_Complete event.
3. The IUT sends an LMP_AU_RAND PDU to the Lower Tester.
4. The Lower Tester calculates a valid rand and replies to the Lower Tester with an LMP_SRES PDU.

Perform Steps 5–7 only if the LMP_SRES PDU is valid. If the LMP_SRES PDU is invalid, then the IUT may optionally disconnect the ACL connection. If it does, then the Lower Tester reestablishes it and the next repeat begins.

5. The Lower Tester sends an LMP_AU_RAND PDU to the IUT.
6. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester. If it does, then the Upper Tester sends an HCI_Link_Key_Request_Reply containing the stored link key sent by the IUT and receives a successful HCI_Command_Complete event.
7. The IUT responds to the Lower Tester with an LMP_SRES PDU.
8. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x00 if the LMP_SRES sent in Step 4 is valid, or set to 0x05 (Authentication Failure) if it is invalid.

Repeat Steps 1–8 18 times, where every sixth repeat, one of the LMP_SRES bits in Step 4 is inverted.

The Lower Tester issues a warning if the following rules are broken:

- The minimum delay between attempts is at least 1 second.
- The exponential “stepping up” uses a multiplier of at least 2.
- The “stepping down” uses a divisor that is not greater than the multiplier.
- The maximum delay is at least 10 times the minimum delay. (This is only measurable if enough repeats are done such that a maximum is visible. If the last step up has a multiplier less than 2, it can be assumed to be the maximum.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT sends a new LMP_AU_RAND each time and does not reuse an old one.

  In Step 8, the IUT does not complete the Authentication procedure if the value for rand sent by the Lower Tester is wrong. If the value for rand is valid, then the IUT and the Lower Tester finish Authentication.

LMP/AUT/BV-43-C [Repeated Authentication Failure, Secure Authentication, Central]

- Test Purpose

  Verify that the IUT correctly handles repeated failed authentication attempts when using Secure Authentication.

- Reference

  [13] 4.2.1.4

- Initial Condition
  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.
- Test Procedure

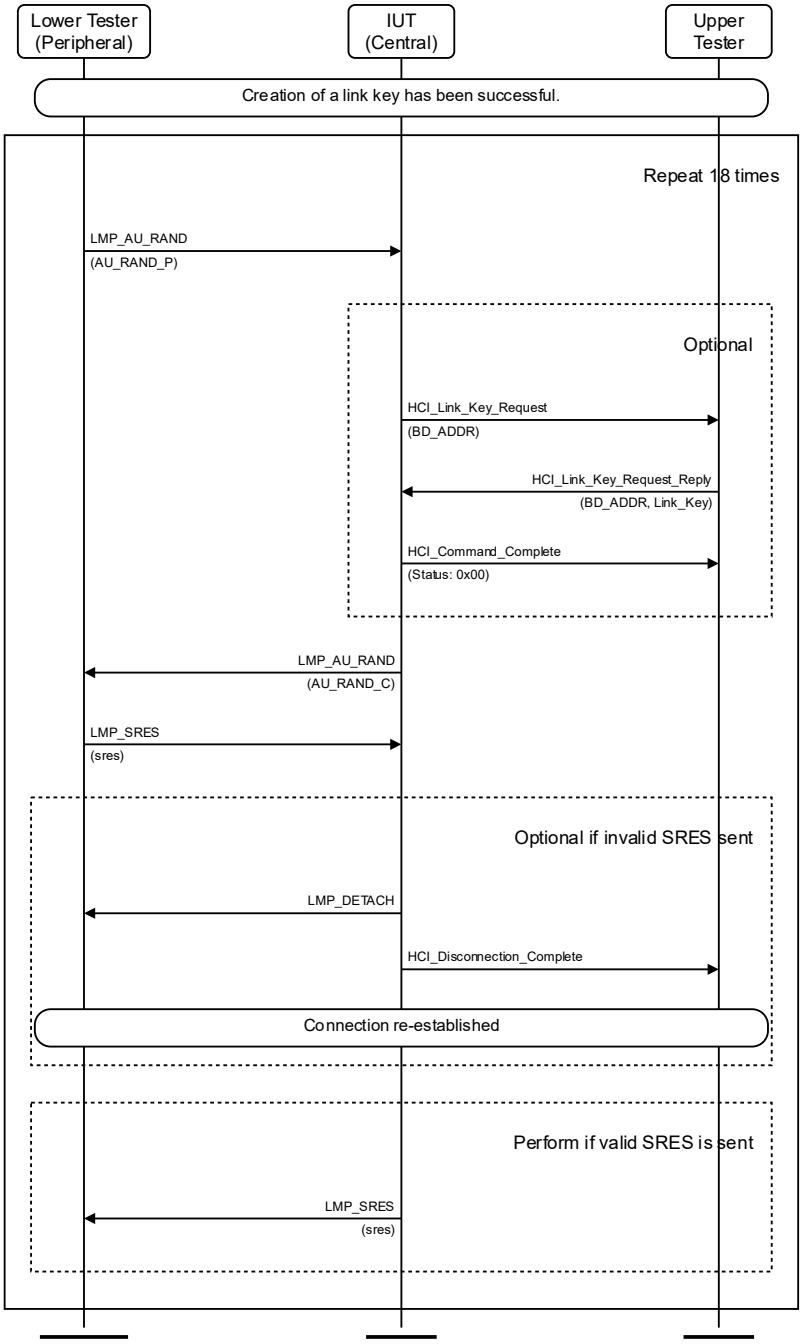


Figure 4.5-44: LMP/AUT/BV-43-C [Repeated Authentication Failure, Secure Authentication, Central] MSC

1. The Lower Tester sends an LMP_AU_RAND PDU with a new AU_RAND_P to the IUT and waits until it receives a reply.
2. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester. If it does, then the Upper Tester responds with an HCI_Link_Key_Request_Reply command containing the stored link key sent by the IUT and receives a successful HCI_Command_Complete event.
3. The IUT sends an LMP_AU_RAND PDU with a new AU_RAND_C to the Lower Tester.
4. The Lower Tester calculates a valid rand and replies to the IUT with an LMP_SRES PDU.
5. The IUT does not complete the Authentication Request if it has received an invalid rand in Step 4; otherwise, it continues with the procedure. The IUT may disconnect the connection at this point. If it does, then the Lower Tester reestablishes the connection and the next repeat begins.
6. If the rand in Step 5 is valid, the IUT sends an LMP_SRES PDU to the Lower Tester.

Repeat Steps 1–6 18 times, where every sixth repeat, one of the LMP_SRES bits in Step 4 is inverted.

The Lower Tester issues a warning if the following rules are broken:

- The minimum delay between attempts is at least 1 second.
- The exponential “stepping up” uses a multiplier of at least 2.
- The “stepping down” uses a divisor that is not greater than the multiplier.
- The maximum delay is at least 10 times the minimum delay. (This is only measurable if enough repeats are done such that a maximum is visible. If the last step up has a multiplier less than 2, it can be assumed to be the maximum.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT sends a new AU_RAND_C each time and does not reuse an old one.

  In Step 5, the IUT does not complete the Authentication procedure if the value for rand sent by the Lower Tester is wrong. If the value for rand is valid, then the IUT and the Lower Tester finish Authentication.

LMP/AUT/BV-44-C [Repeated Authentication Failure, Secure Authentication, Peripheral]

- Test Purpose

  Verify that the IUT correctly handles repeated failed authentication attempts when using Secure Authentication.

- Reference

  [13] 4.2.1.4

- Initial Condition

  - See the “Connection Establishment Lower Tester” preamble and “Secure Simple Pairing P-256” default settings.

- Test Procedure

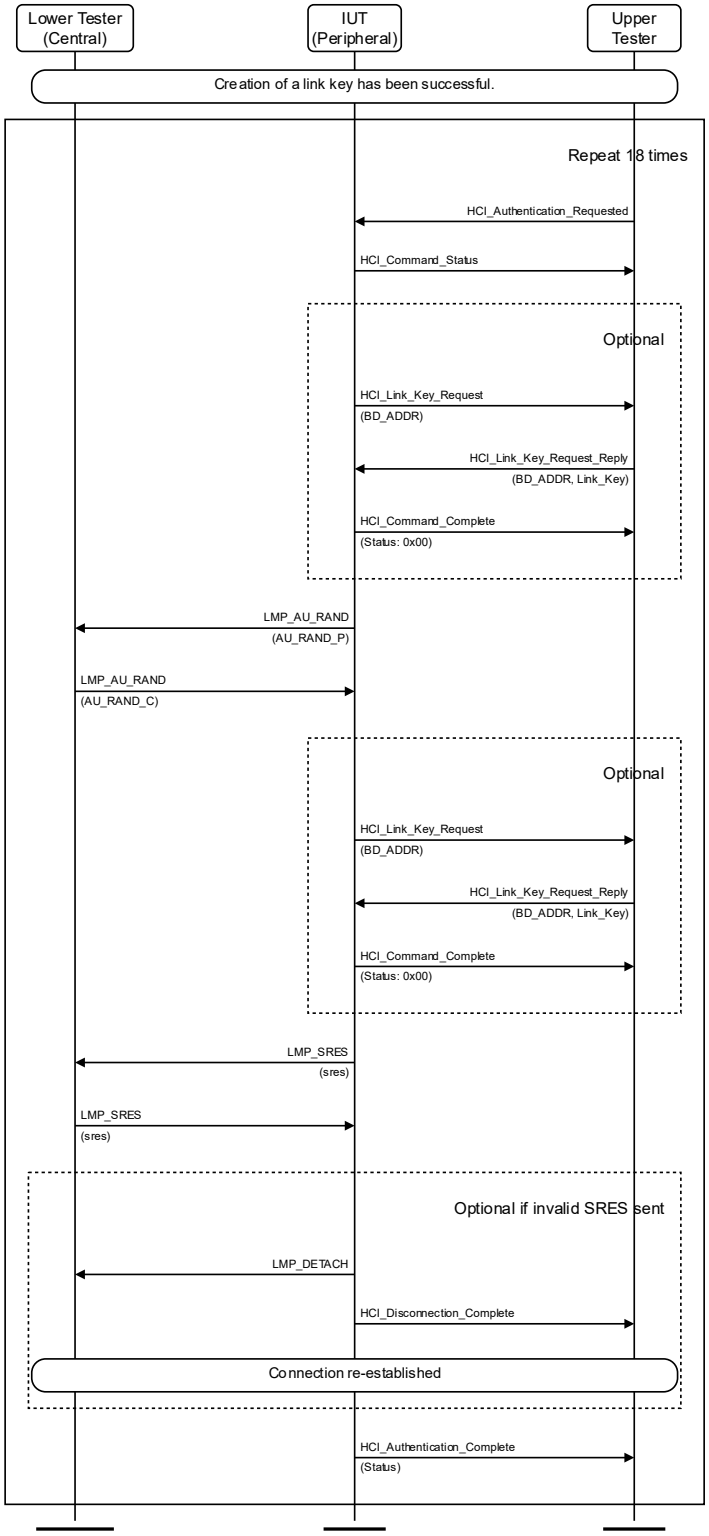


Figure 4.5-45: LMP/AUT/BV-44-C [Repeated Authentication Failure, Secure Authentication, Peripheral] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT and receives a successful HCI_Command_Status in return.
2. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester. If it does, then the Upper Tester responds with an HCI_Link_Key_Request_Reply command containing the stored link key sent by the IUT and receives a successful HCI_Command_Complete event.
3. The IUT sends an LMP_AU_RAND PDU with a new AU_RAND_P to the Lower Tester.
4. The Lower Tester sends an LMP_AU_RAND PDU with a new AU_RAND_C to the IUT and waits until it receives a reply.
5. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester. If it does, then the Upper Tester responds with an HCI_Link_Key_Request_Reply command containing the stored link key sent by the IUT and receives a successful HCI_Command_Complete event.
6. The IUT sends an LMP_SRES PDU to the Lower Tester.
7. The Lower Tester calculates a valid rand and replies to the IUT with an LMP_SRES PDU.

Steps 8 and 9 may occur in any order.

8. If the LMP_SRES sent in Step 7 is invalid, then the IUT may optionally disconnect the ACL connection. If it does, then the Lower Tester reestablishes it and the next repeat begins.
9. The IUT sends an HCI_Authentication_Complete event to the Upper Tester with Status set to 0x00 if the LMP_SRES sent in Step 6 is valid, or set to 0x05 (Authentication Failure) if it is invalid.

Repeat Steps 1–9 18 times, where every sixth repeat, one of the LMP_SRES bits in Step 7 is inverted.

The Lower Tester issues a warning if the following rules are broken:

- The minimum delay between attempts is at least 1 second.
- The exponential “stepping up” uses a multiplier of at least 2.
- The “stepping down” uses a divisor that is not greater than the multiplier.
- The maximum delay is at least 10 times the minimum delay. (This is only measurable if enough repeats are done such that a maximum is visible. If the last step up has a multiplier less than 2, it can be assumed to be the maximum.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT sends a new AU_RAND_P each time and does not reuse an old one.

  In Step 9, the IUT does not complete the Authentication procedure if the value for rand sent by the Lower Tester is wrong. If the value for rand is valid, then the IUT and the Lower Tester finish Authentication.

4.6 Encryption

Verify the correct implementation of the Encryption services.

4.6.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.6.2 Encryption – Peripheral

Verify that the Central and the Peripheral agree upon whether to use encryption or not and if encryption only applies to point-to-point packets or if encryption applies to both point-to-point packets and broadcast packets. The IUT is Peripheral.

4.6.2.1 Accept Encryption

- Test Purpose

  Verify that the IUT accepts the encryption negotiation procedure initiated by the Lower Tester and uses the encryption only for point-to-point messages.

- Reference

  [1] 4.2.5

- Initial Condition
  - An ACL connection is established.
  - Creation of a link key is successful.
  - The Lower Tester uses an acceptable key length.
  - The Lower Tester is the Central and the IUT is the Peripheral.
  - The IUT and the Lower Tester indicate support for Secure Connections in their LMP Features based on Table 4.6-1.
- Test Case Configuration

Test Case	Secure Connections	Encryption
LMP/ENC/BV-01-C [Accept Encryption]	Neither support	E0
LMP/ENC/BV-26-C [Accept AES-CCM Encryption Request]	IUT and Lower Tester – Controller and Host	AES-CCM
LMP/ENC/BV-33-C [Accept AES-CCM Encryption Request – Legacy Host]	Lower Tester only – Controller and Host	E0

Table 4.6-1: Accept Encryption test cases

- Test Procedure

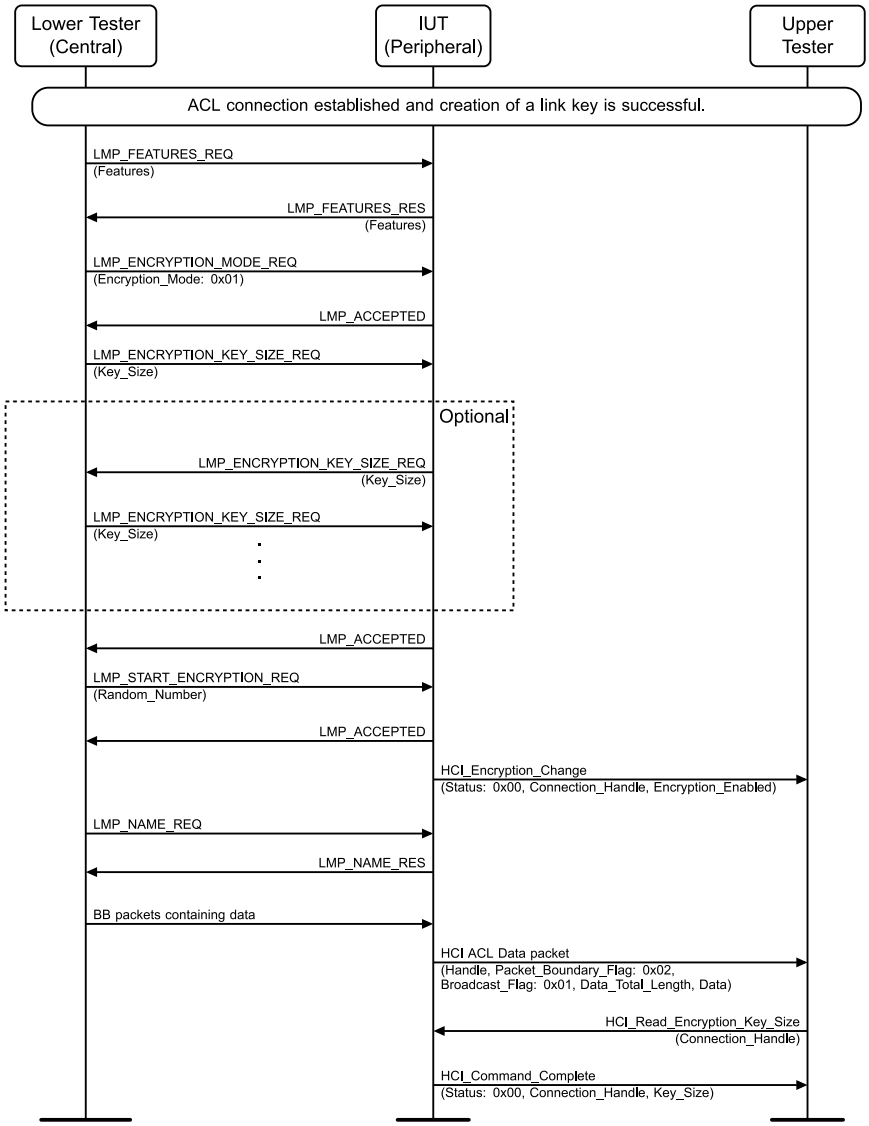


Figure 4.6-1: Accept Encryption MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
5. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size.
6. If the IUT responds to the Lower Tester with an LMP_ENCRYPTION_KEY_SIZE_REQ PDU with a new Key_Size, the Lower Tester and the IUT continue key size negotiation until Step 7 occurs.
7. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

8. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
10. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01, if using E0 encryption, or 0x02, if using AES-CCM encryption as indicated in Table 4.6-1.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.
13. The Lower Tester sends BB packets containing data to the IUT.
14. The IUT sends an HCI ACL Data packet to the Upper Tester.
15. The Upper Tester sends an HCI_Read_Encryption_Key_Size command to the IUT with the Connection_Handle and receives a successful HCI_Command_Complete event with the Connection_Handle and Key_Size.

- Expected Outcome

  Pass verdict

  The IUT accepts the encryption negotiation and uses the encryption afterwards.

  In Step 10, the Encryption_Enabled parameter of the HCI_Encryption_Change event is set to 0x01 if using E0 encryption or 0x02 if using AES-CCM encryption as indicated in Table 4.6-1.

  In Step 12, the IUT sends the LMP_NAME_RES PDU and proves that encryption is used.

  In Step 14, the IUT sends an HCI ACL Data packet with non-encrypted payload to the Upper Tester.

  In Step 15, the IUT sends a successful HCI_Command_Complete event with the Key_Size parameter matching the Key_Size in the LMP_ENCRYPTION_KEY_SIZE_REQ PDU accepted by the IUT.

LMP/ENC/BV-02-C [Accept Broadcast Encryption]

- Test Purpose

  Verify that the IUT accepts the broadcast encryption negotiation procedure and uses the encryption both for point-to-point messages and for broadcast messages.

- Reference

  [1] 4.2.5

- Initial Condition

  - An ACL connection is established.
  - Creation of a link key is successful.
  - No encryption is being used.
  - The Lower Tester is the Central and the IUT is the Peripheral.

- Test Procedure

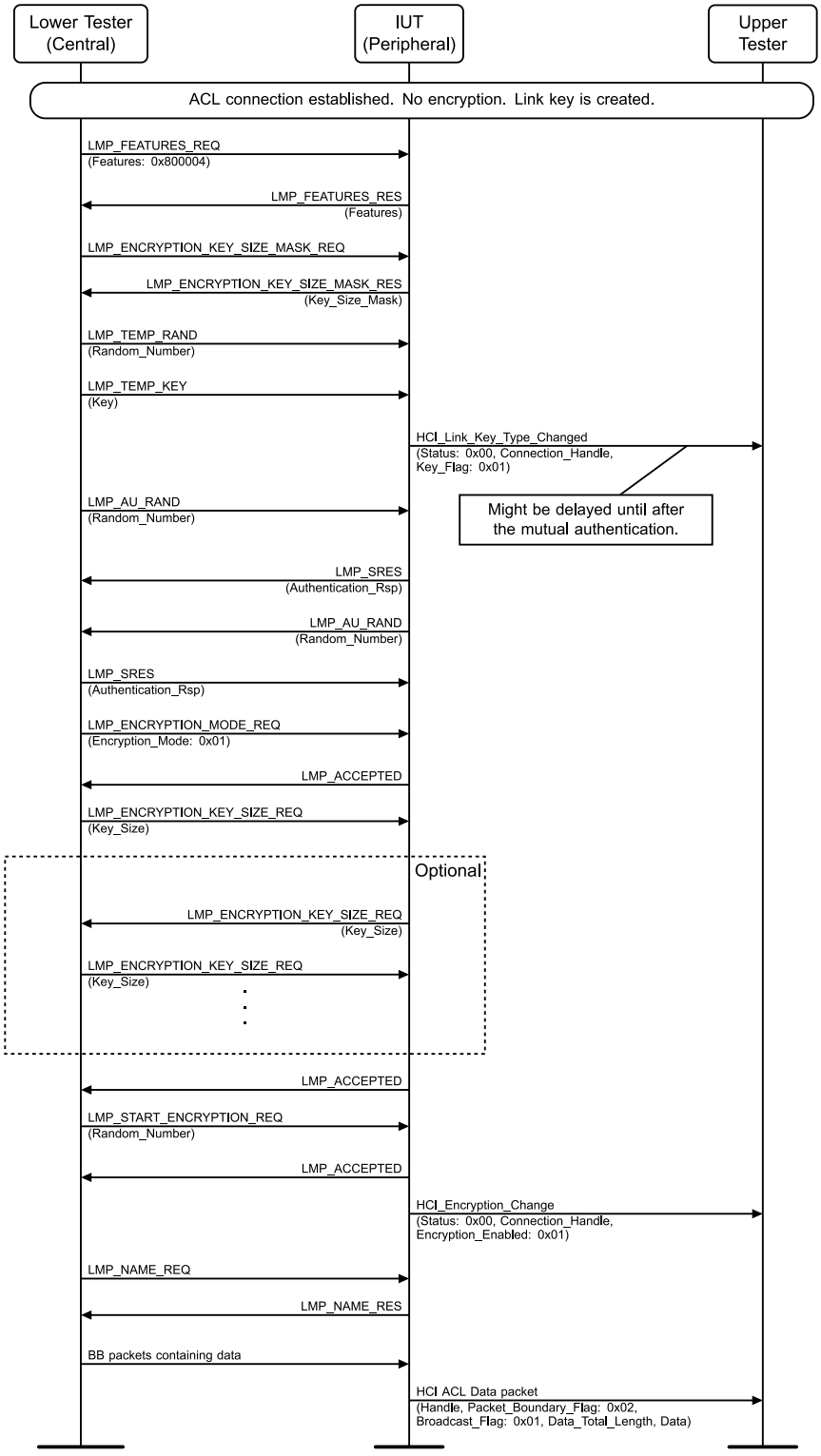


Figure 4.6-2: LMP/ENC/BV-02-C [Accept Broadcast Encryption] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter set to 0x800004 indicating Encryption and Broadcast Encryption support.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

3. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU to the IUT.
4. The IUT responds with an LMP_ENCRYPTION_KEY_SIZE_MASK_RES PDU with the Key_Size_Mask.
5. The Lower Tester sends an LMP_TEMP_RAND PDU with a Random_Number and an LMP_TEMP_KEY PDU with a Key to the IUT.
6. The IUT sends a successful HCI_Link_Key_Type_Changed event to the Upper Tester with the Connection_Handle and Key_Flag set to 0x01 (Using Temporary Link Key).
7. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
8. The IUT sends an LMP_SRES PDU to the Lower Tester with the Authentication_Rsp.
9. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
10. The Lower Tester sends an LMP_SRES PDU to the IUT with the Authentication_Rsp.
11. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
12. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
13. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size.
14. If the IUT responds to the Lower Tester with an LMP_ENCRYPTION_KEY_SIZE_REQ PDU with a new Key_Size, the Lower Tester and the IUT continue key size negotiation until Step 15 occurs.
15. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
16. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the Lower tester with a Random_Number.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
18. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.
19. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
20. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.
21. The Lower Tester sends BB packets containing data to the IUT.
22. The IUT sends an HCI ACL Data packet to the Upper Tester.

- Expected Outcome

  Pass verdict

  The IUT accepts the encryption negotiation and uses the encryption on both broadcast and point-to-point messages.

  In Step 22, an HCI ACL Data packet is sent to the Upper Tester.

LMP/ENC/BV-66-C [Accept Encryption]

- Test Purpose

  Verify that the IUT accepts the encryption negotiation procedure and uses the encryption only for point-to-point messages. The Lower Tester is Central and the IUT is Peripheral.

- Reference

  [1] 4.2.5

- Initial Condition
  - An ACL connection is established.
  - Creation of a link key is successful.
  - The Lower Tester uses an acceptable Key length.
- Test Procedure

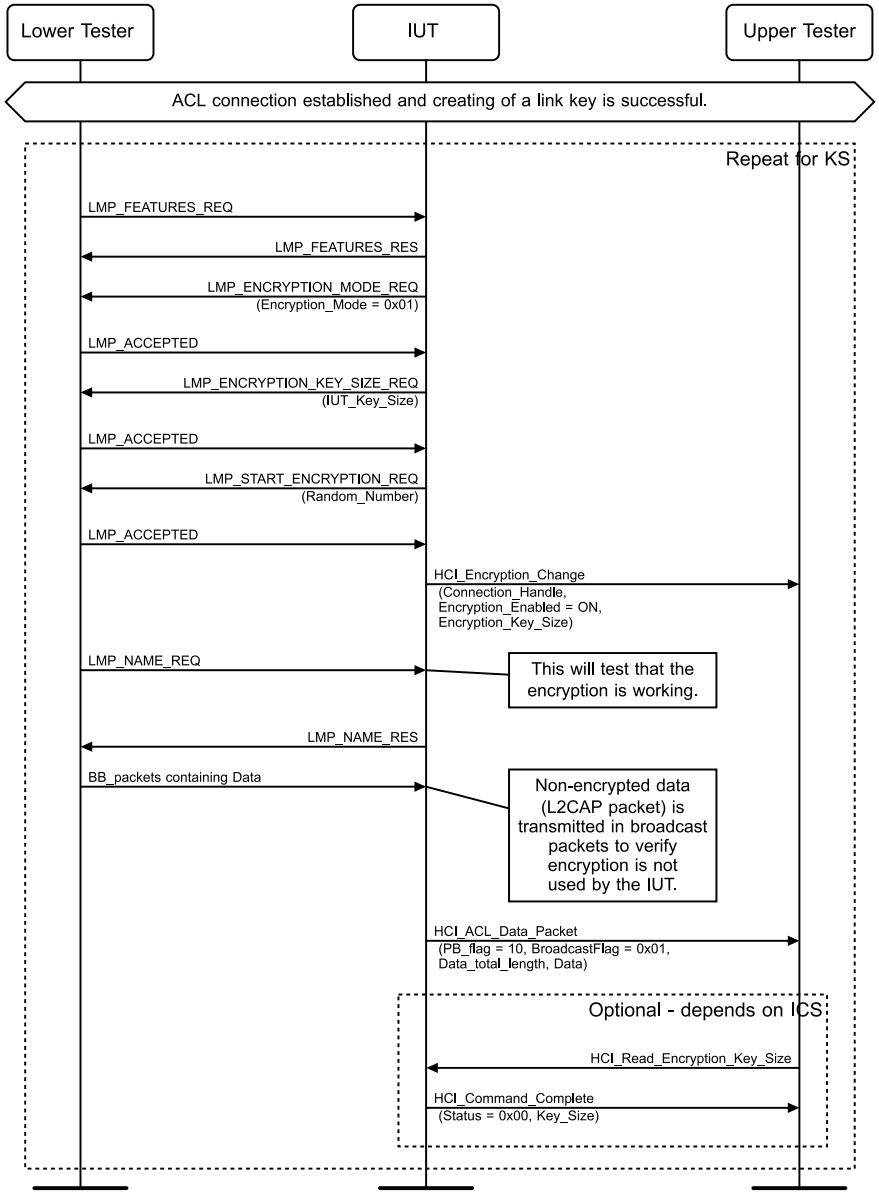


Figure 4.6-3: LMP/ENC/BV-66-C [Accept Encryption] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT.
2. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester.
3. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
4. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.

5. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with an IUT_Key_Size. If the IUT sends the LMP PDU when KS < 7, the test fails.
6. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
7. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
8. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
9. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled ON and an Encryption_Key_Size.
10. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
11. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
12. The Lower Tester sends BB packets containing data.
13. The IUT sends an HCI ACL Data Packet to the Upper Tester with PB_flag = 10, BroadcastFlag = 0x01, Data_total_length, and Data.
14. Depending on the ICS, the IUT may execute Step 14.
    14A. The Upper Tester sends an HCI_Read_Encryption_Key_Size command to the IUT.
    14B. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with Key_Size.

- Expected Outcome

  Pass verdict

  The IUT accepts the encryption negotiation and uses the encryption afterwards.

  The IUT must respond correctly to the PDU LMP_NAME_REQ to prove that encryption is used.

  The IUT sends HCI ACL Data with non-encrypted payload to the Upper Tester.

  If Read Encryption Key_Size is supported, the IUT returns the Key_Size parameter from the LMP_ENCRYPTION_KEY_SIZE_REQ PDU in the Command Complete event following the HCI Read Encryption Key_Size command.

  Fail verdict

  In Step 5, the IUT sends the LMP_ENCRYPTION_KEY_SIZE_REQ PDU when KS < 7.

- Notes

  If the IUT starts to negotiate for encryption Key_Size the Lower Tester must negotiate.

LMP/ENC/BV-67-C [Accept Broadcast Encryption]

- Test Purpose

  Verify that the IUT accepts the broadcast encryption negotiation procedure and uses the encryption both for point-to-point messages as well as broadcast messages. The Lower Tester is Central and the IUT is Peripheral.

- Reference

  [1] 4.2.5

- Initial Condition

  - See Figure 4.6-4.

- Test Procedure

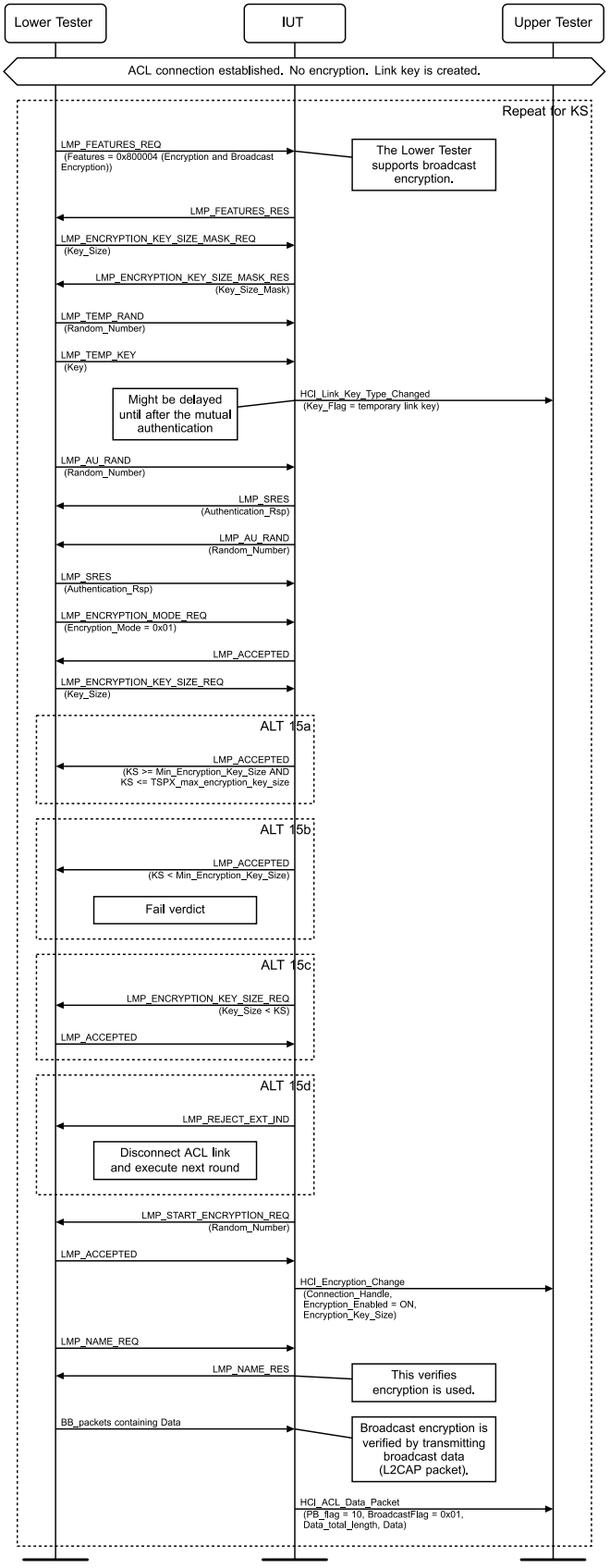


Figure 4.6-4: LMP/ENC/BV-67-C [Accept Broadcast Encryption] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with Features set to 0x800004 (Encryption and Broadcast Encryption).
2. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester.
3. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU to the IUT with a Key_Size set to KS.
4. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_MASK_RES PDU to the Lower Tester with Key_Size_Mask.
5. The Lower Tester sends an LMP_TEMP_RAND PDU to the IUT with a Random_Number.
6. The Lower Tester sends an LMP_TEMP_KEY PDU to the IUT with a Key.
7. The IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with Key_Flag set to the temp link key.
8. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
9. The IUT sends an LMP_SRES PDU to the Lower Tester with Authentication_Rsp.
10. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
11. The Lower Tester sends an LMP_SRES PDU to the IUT with Authentication_Rsp.
12. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
13. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
14. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with Key_Size set to KS.
15. The IUT executes either 15a, 15b, 15c, or 15d:
    a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
    b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
    c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
    d. Reject the suggested Key_Size, in which case skip to Step 23.
16. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with Random_Number.
17. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
18. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled ON and an Encryption_Key_Size.
19. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
20. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
21. The Lower Tester sends BB packet containing data to the IUT.
22. The IUT sends HCI ACL Data Packet events to the Upper Tester with PB_Flag set to 10, Broadcast_Flag set to 0x01, Data_total_length, and Data.
23. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT accepts the encryption negotiation and uses the encryption on both broadcast and point-to-point messages. HCI ACL Data Packet is sent to the Upper Tester.

Fail verdict

In Step 15b, the IUT accepts a key size < Min_Encryption_Key_Size.

In Step 15c, the IUT suggests a key size < 7.

- Notes

  If the IUT starts to negotiate for encryption Key_Size, the Lower Tester must negotiate.

LMP/ENC/BV-71-C [Accept AES-CCM Encryption Request]

- Test Purpose

  Verify that the IUT accepts the encryption negotiation procedure initiated by the Lower Tester and uses AES-CCM encryption only for point-to-point messages when the remote controller's LMP feature bits indicate support for Secure Connections both in the Controller and the Host.

- Reference

  [1] 4.2.5

- Initial Condition

  - The Lower Tester is Central and the IUT is Peripheral.
  - An ACL connection has been established between the IUT and the Lower Tester, and the creation of a link key between the IUT and the Lower Tester has been successful.
  - The Lower Tester uses an acceptable key length.
  - The Lower Tester LMP feature bits have support for Secure Connections both in the Controller and the Host.

- Test Procedure

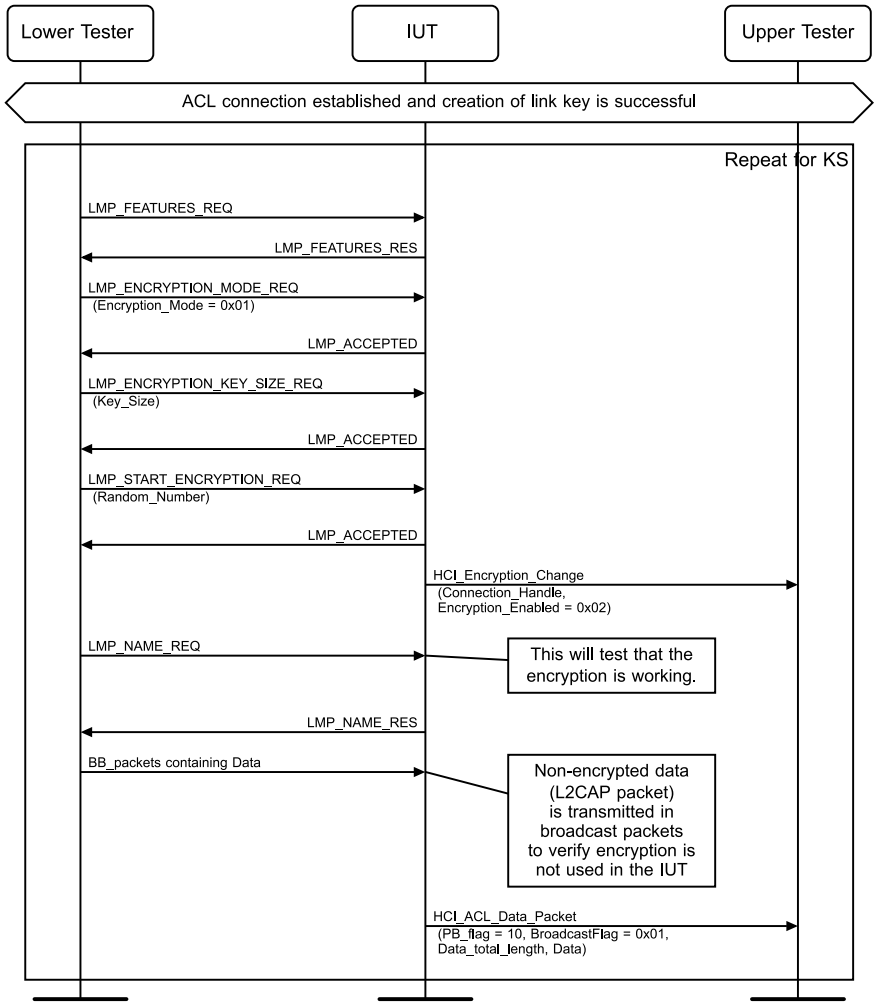


Figure 4.6-5: LMP/ENC/BV-71-C [Accept AES-CCM Encryption Request] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with Features set for Secure Connections for both the Controller and the Host.
2. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester.
3. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
4. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
5. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size set to KS.
6. The IUT executes either 6a, 6b, 6c, or 6d.
   a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
   b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
   c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
   d. Reject the suggested Key_Size, in which case skip to Step 14.

7. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with Random_Number.
8. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
9. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x02.
10. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
11. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
12. The Lower Tester sends BB packet containing data to the IUT.
13. The IUT sends HCI ACL Data Packet events to the Upper Tester with PB_Flag set to 10, Broadcast_Flag set to 0x01, Data_total_length, and Data.
14. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT accepts the encryption negotiation and uses the encryption afterwards.

  The IUT responds correctly to the PDU LMP_NAME_REQ (this proves that encryption is used).

  The IUT sends HCI ACL Data with non-encrypted payload to the Upper Tester.

  If Read Encryption Key_Size is supported, the IUT returns the Key_Size parameter from the LMP_ENCRYPTION_KEY_SIZE_REQ PDU in the Command Complete event following the HCI Read Encryption Key_Size command.

  The Encryption_Enabled Parameter in the Encryption Change event reports AES-CCM encryption has been enabled.

  Fail verdict

  In Step 6b, the IUT accepts a key size < Min_Encryption_Key_Size.

  In Step 6c, the IUT suggests a key size < 7.

- Notes

  If the IUT starts to negotiate for encryption Key_Size, the Lower Tester must negotiate.

  This test case is similar to LMP/ENC/BV-66-C [Accept Encryption].

LMP/ENC/BV-72-C [Accept AES-CCM Encryption Request – Legacy Host]

- Test Purpose

  Verify that the IUT accepts the encryption negotiation procedure initiated by the Lower Tester and uses E0 encryption only for point-to-point messages when the remote controller's LMP feature bits indicate support for Secure Connections both in the Controller and the Host but the local Host does not indicate support for Secure Connections and reports the correct Encryption_Enabled to a legacy Host.

- Reference

  [1] 4.2.5

- Initial Condition

  - The Lower Tester is Central and the IUT is Peripheral.
  - The Upper Tester does not set the Secure Connections Host Support to enabled.

- An ACL connection has been established between the IUT and the Lower Tester, and the creation of a link key between the IUT and the Lower Tester has been successful.
- The Lower Tester uses an acceptable Key length.

- Test Procedure

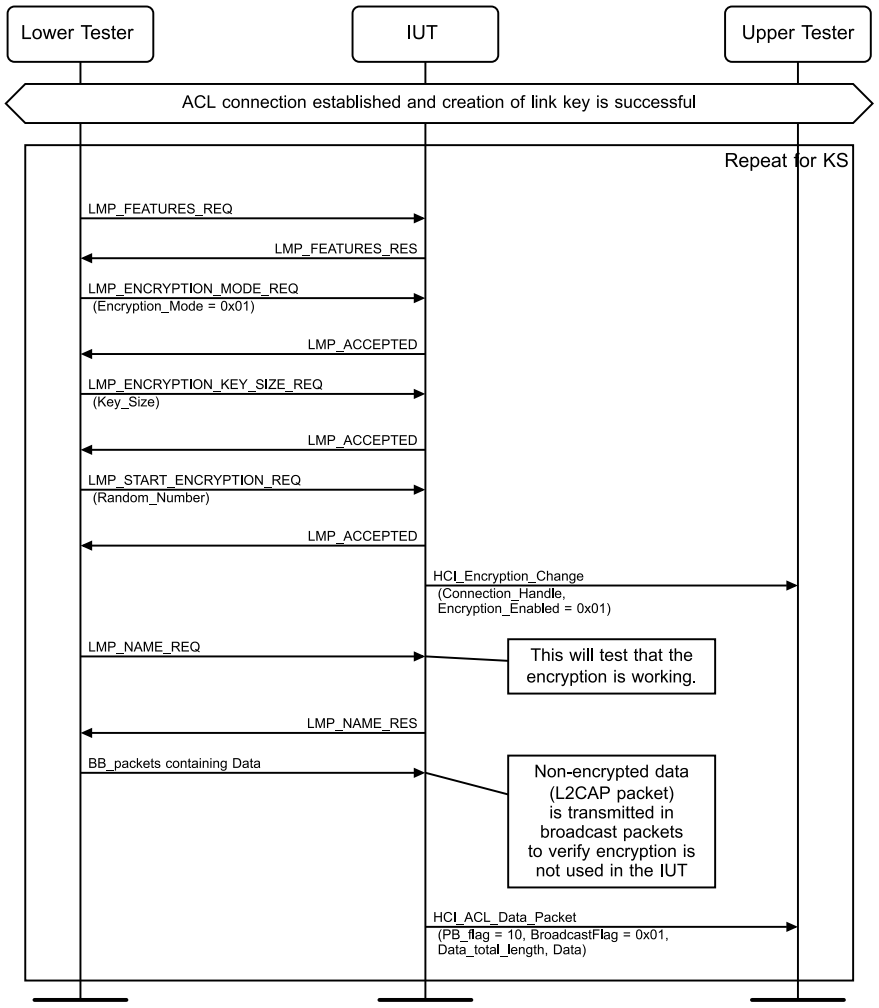


Figure 4.6-6: LMP/ENC/BV-72-C [Accept AES-CCM Encryption Request – Legacy Host] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT.
2. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester.
3. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
4. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
5. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size set to KS.
6. The IUT executes either 6a, 6b, 6c, or 6d.
   a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
   b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.

c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
d. Reject the suggested Key_Size, in which case skip to Step 14.

7. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with Random_Number.
8. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
9. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x01.
10. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
11. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
12. The Lower Tester sends BB packet containing data to the IUT.
13. The IUT sends HCI ACL Data Packet events to the Upper Tester with PB_Flag set to 10, Broadcast_Flag set to 0x01, Data_total_length, and Data.
14. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT accepts the encryption negotiation and uses encryption afterwards.

  The IUT responds correctly to the PDU LMP_NAME_REQ (this proves that encryption is used).

  The IUT sends HCI ACL Data with non-encrypted payload to the Upper Tester.

  If Read Encryption Key_Size is supported, the IUT returns the Key_Size parameter from the LMP_ENCRYPTION_KEY_SIZE_REQ PDU in the Command Complete event following the HCI Read Encryption Key_Size command.

  The Encryption_Enabled Parameter in the Encryption Change event reports “Link Level Encryption is ON with E0”.

  Fail verdict

  In Step 6b, the IUT accepts a key size < Min_Encryption_Key_Size.

  In Step 6c, the IUT suggests a key size < 7.

- Notes

  If the IUT starts to negotiate for encryption Key_Size, the Lower Tester must negotiate.

4.6.2.2 Stop Encryption

- Test Purpose

  Verify that the IUT stops using encryption after a request from the Lower Tester or requests to stop using encryption upon receiving the appropriate HCI command.

- Reference

  [1] 4.2.5.4

- Initial Condition

  - An Encrypted Point-to-Point connection has been established.
  - The Lower Tester is the Central and the IUT is the Peripheral.
  - The IUT has defined its supported LMP Features.

- Test Case Configuration

Test Case	Stop Requester
LMP/ENC/BV-04-C [Stop Encryption, Central Command]	Central
LMP/ENC/BV-09-C [Stop Encryption, Host Command]	Host

Table 4.6-2: Stop Encryption test cases

- Test Procedure

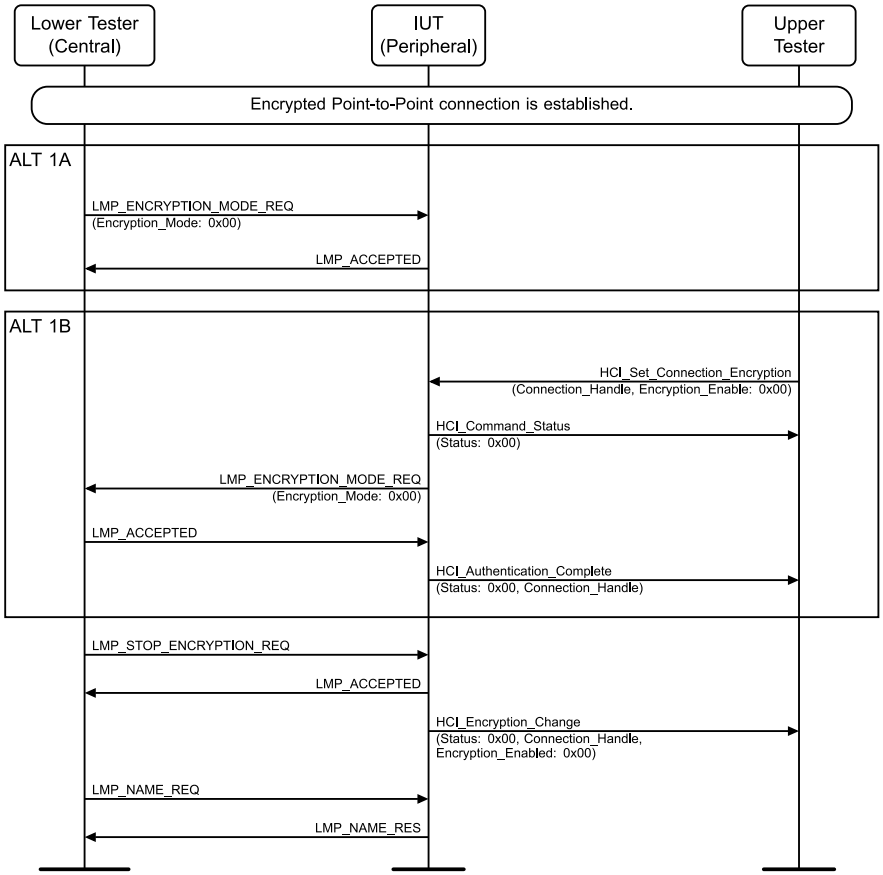


Figure 4.6-7: Stop Encryption MSC

1. Perform either alternative 1A or 1B depending on who is the stop requester as indicated in Table 4.6-2:

   Alternative 1A (The Central is the stop requester):

   1A.1. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ to the IUT with Encryption_Mode set to 0x00 (No Encryption).

   1A.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.

   Alternative 1B (The Host is the stop requester):

   1B.1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x00 and receives a successful HCI_Command_Status event in return.

   1B.2. The IUT sends an LMP_ENCRYPTION_MODE_REQ to the Lower Tester with Encryption_Mode set to 0x00 (No Encryption).

   1B.3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.

2. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.
3. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enable set to 0x00.
6. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
7. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.

- Expected Outcome

  Pass verdict

  In alternative 1B, the IUT sends the LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester.

  In Step 4, the IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_STOP_ENCRYPTION_REQ PDU and stops using encryption.

  In Step 7, the IUT sends the LMP_NAME_RES PDU to the Lower Tester and proves that encryption is not used.

LMP/ENC/BV-11-C [Semi-permanent Link Key]

- Test Purpose

  Verify that the IUT accepts that the semi-permanent link key becomes the current link key upon notice from the Lower Tester and that the encryption is stopped.

- Reference

  [1] 4.2.4.2

- Initial Condition

  - Broadcast encryption is used.
  - The Lower Tester is the Central. The IUT is the Peripheral.

- Test Procedure

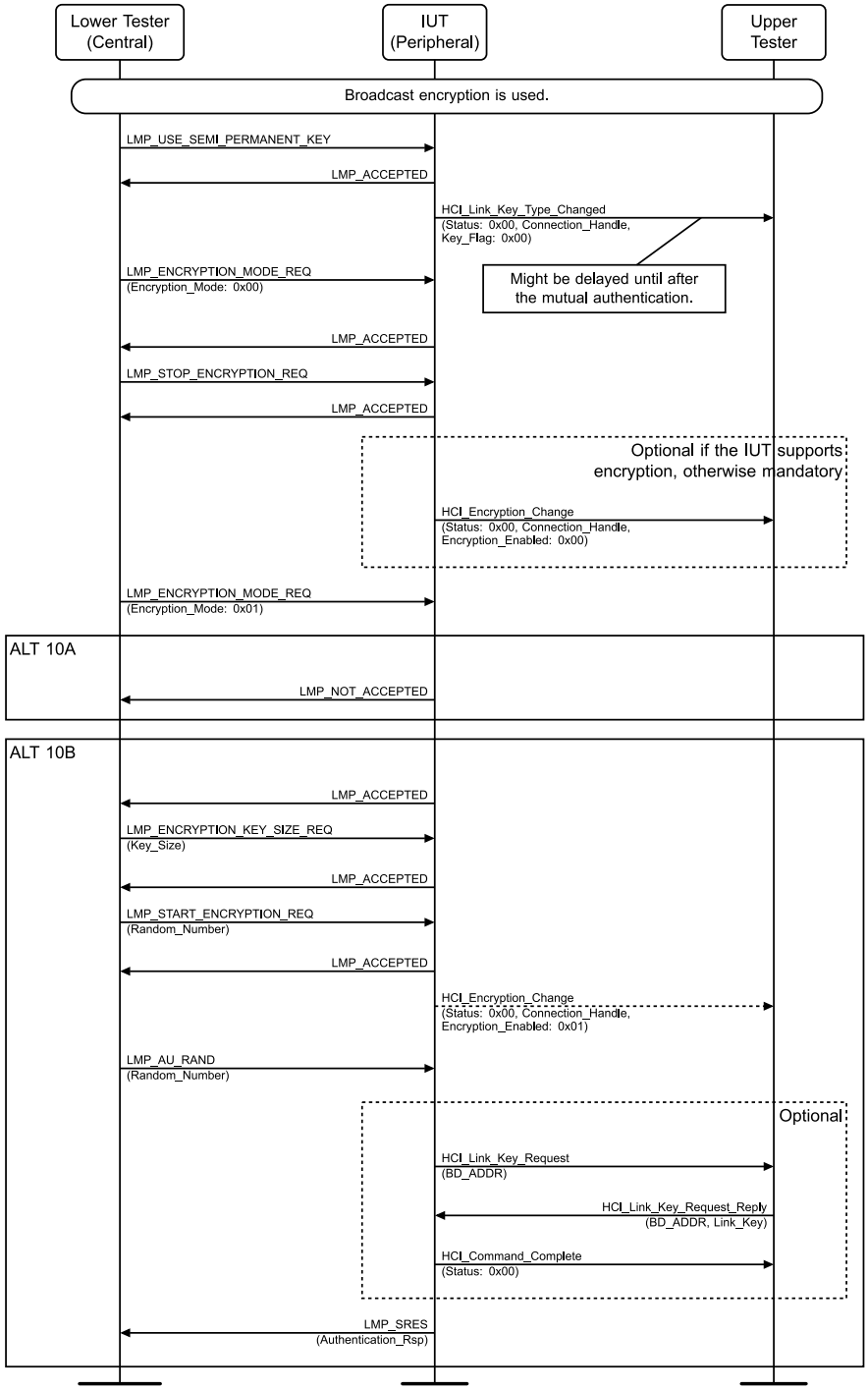


Figure 4.6-8: LMP/ENC/BV-11-C [Semi-permanent Link Key] MSC

1. The Lower Tester sends an LMP_USE_SEMI_PERMANENT_KEY PDU to the IUT.
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_USE_SEMI_PERMANENT_KEY PDU Opcode.
3. The IUT sends a successful HCI_Link_Key_Type_Changed event to the Upper Tester with the Connection_Handle and Key_Flag set to 0x00 (Using Semi-permanent Link Key).
4. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x00.

5. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
6. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
8. If the IUT does not support encryption or, optionally, if it does, then the IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x00.
9. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
10. Perform either alternative 10A or 10B depending on whether the IUT supports encryption.
    Alternative 10A (The IUT does not support encryption):

    10A.1. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.

    Alternative 10B (The IUT supports encryption):

    10B.1. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
    10B.2. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size.
    10B.3. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    10B.4. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
    10B.5. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    10B.6. Optionally, the IUT sends an HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.
    10B.7. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
    10B.8. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
    10B.9. If Step 10B.8 occurs, the Upper Tester sends an HCI_Link_Key_Request_Reply command to the IUT with the BD_ADDR and Link_Key and receives a successful HCI_Command_Complete event in response.
    10B10. The IUT sends an LMP_SRES PDU to the Lower Tester with the Authentication_Rsp.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_USE_SEMI_PERMANENT_KEY, LMP_ENCRYPTION_MODE_REQ and LMP_STOP_ENCRYPTION_REQ PDUs.

  The link key is a Semi-Permanent Key, and encryption is stopped.

LMP/ENC/BV-68-C [Semi-permanent Link Key]

- Test Purpose

  Verify that the IUT accepts that the semi-permanent link key becomes the current link key upon notice from the Lower Tester. Verify that the encryption is stopped. The Lower Tester is Central. The IUT is Peripheral.

- Reference

  [1] 4.2.4.2

- Initial Condition

  - See Figure 4.6-9.

- Test Procedure

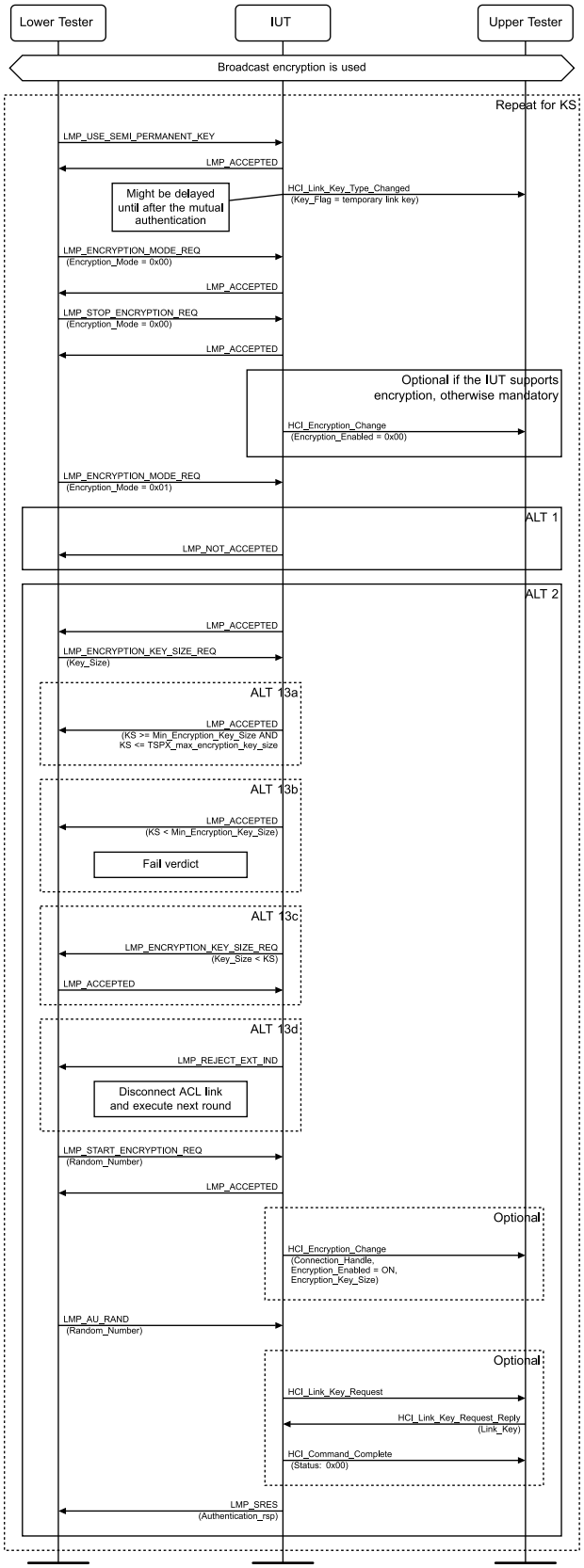


Figure 4.6-9: LMP/ENC/BV-68-C [Semi-permanent Link Key] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Lower Tester sends an LMP_USE_SEMI_PERMANENT_KEY PDU to the IUT.
2. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
3. The IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with a Key_Flag.
4. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x00.
5. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
6. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT with Encryption_Mode set to 0x00.
7. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
8. If the IUT supports encryption, the IUT may execute Step 8.
   8A. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x00.
9. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
10. If the IUT does not support encryption, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester and continues to the next KS round.
11. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
12. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size set to KS.
13. The IUT executes either 13a, 13b, 13c, or 13d.
    a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
    b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
    c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
    d. Reject the suggested Key_Size, in which case skip to Step 20.
14. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
15. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
16. The IUT may send an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled ON and an Encryption_Key_Size.
17. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
18. The IUT may execute Step 18.
    18A. The IUT sends an HCI_Link_Key_Request event to the Upper Tester.
    18B. The Upper Tester sends an HCI_Link_Key_Request_Reply command to the IUT with Link_Key.
    18C. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
19. The IUT sends an LMP_SRES to the Lower Tester with Authentication_rsp.
20. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT transmits PDU LMP_ACCEPTED upon reception of LMP_USE_SEMI_PERMANENT_KEY, LMP_ENCRYPTION_MODE_REQ and LMP_STOP_ENCRYPTION_REQ. The link key must be the semi-permanent key and encryption must be stopped.

Fail verdict

In Step 13b, the IUT accepts a key size < Min_Encryption_Key_Size.

In Step 13c, the IUT suggests a key size < 7.

LMP/ENC/BV-12-C [Reject Broadcast Encryption]

- Test Purpose

  Verify that the IUT does not accept the broadcast encryption negotiation procedure.

- Reference

  [1] 4.2.5

- Initial Condition

  - An ACL connection is established.
  - Creation of a link key is successful.
  - The Lower Tester is the Central and the IUT is the Peripheral.

- Test Procedure

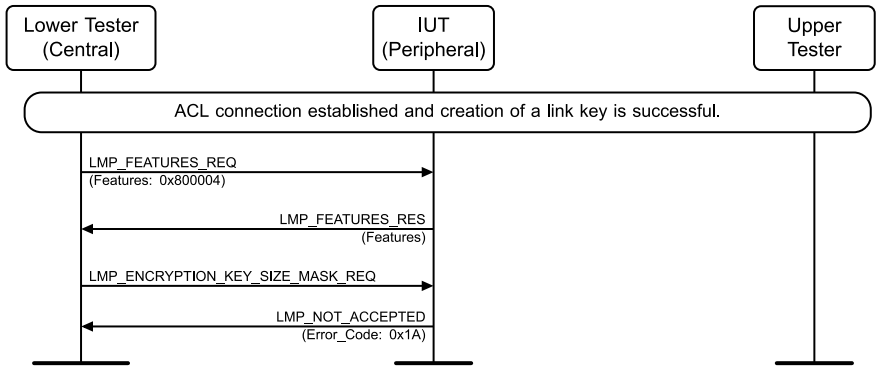


Figure 4.6-10: LMP/ENC/BV-12-C [Reject Broadcast Encryption] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter set to 0x800004 indicating Encryption and Broadcast Encryption support.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU to the IUT.
4. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU Opcode and Error_Code set to 0x1A (Unsupported LMP Feature).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1A (Unsupported LMP Feature) upon reception of the LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU.

4.6.2.3 Pausing and Resuming Encryption without Disabling Encryption_Mode – Peripheral Initiated

- Test Purpose

  Verify that the IUT as the Peripheral can pause and resume encryption without disabling the Encryption_Mode.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition

  - The IUT is the Peripheral and the Lower Tester is the Central.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-3.
  - Both devices are sending data to the other device.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-15-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Peripheral Initiated]	E0
LMP/ENC/BV-38-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Peripheral Initiated as a result of change connection link Key]	AES-CCM

Table 4.6-3: Pausing and Resuming Encryption without Disabling Encryption_Mode – Peripheral Initiated test cases

- Test Procedure

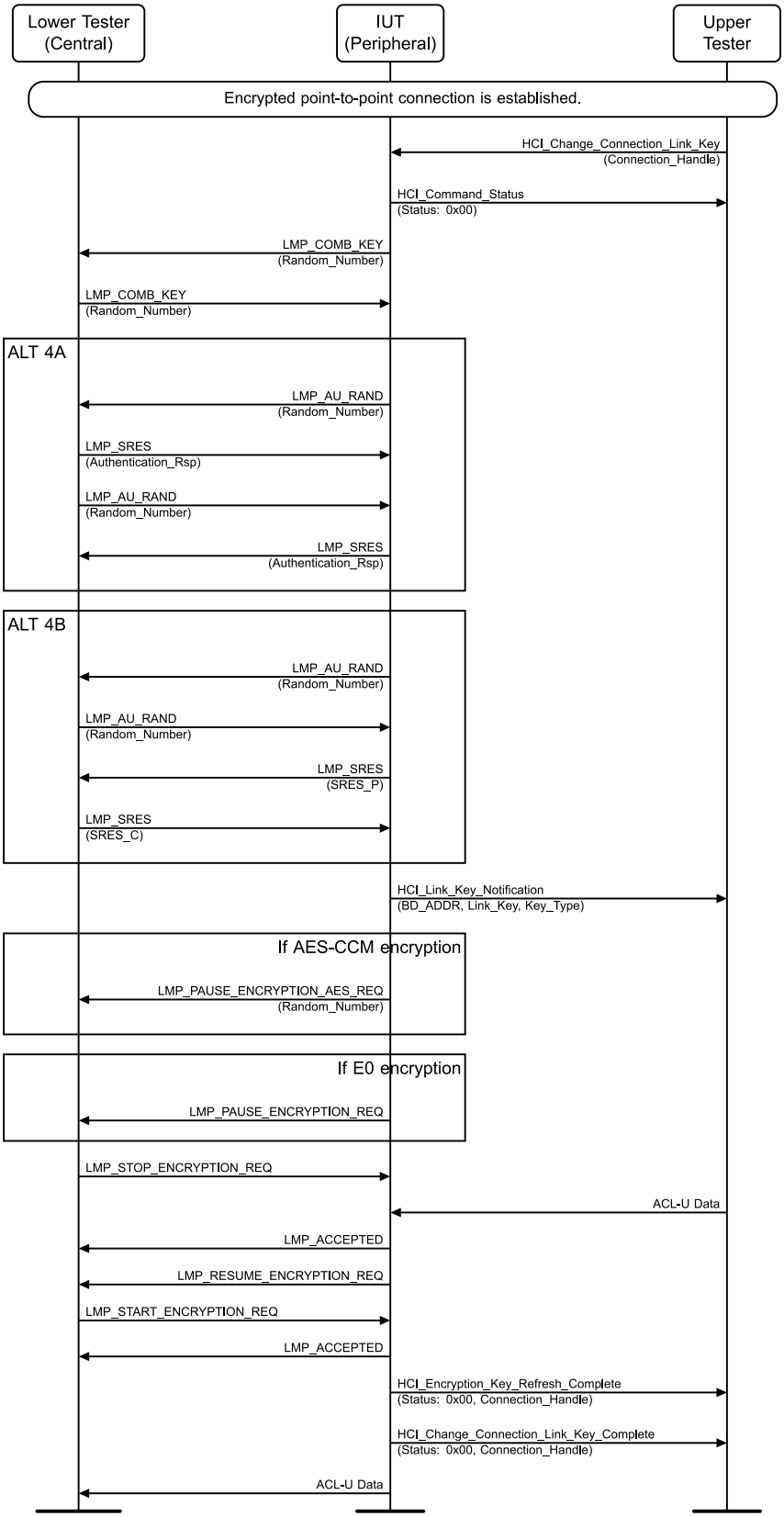


Figure 4.6-11: Pausing and Resuming Encryption without Disabling Encryption_Mode – Peripheral Initiated MSC

1. The Upper Tester sends an HCI_Change_Connection_Link_Key command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
4. Perform either alternative 4A or 4B depending on the encryption indicated in Table 4.6-3.
   Alternative 4A (The connection uses E0 encryption):
   - 4A.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
   - 4A.2. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.
   - 4A.3. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
   - 4A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the Authentication_Rsp.

   Alternative 4B (The connection uses AES-CCM encryption):
   - 4B.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
   - 4B.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
   - 4B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
   - 4B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES_C.
5. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with the BD_ADDR, Link_Key, and Key_Type.
6. If AES-CCM encryption is being used as indicated in Table 4.6-3, the IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number; otherwise, the IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
7. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
8. The Upper Tester sends ACL-U Data to the IUT.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
10. The IUT sends an LMP_RESUME_ENCRYPTION_REQ PDU to the Lower Tester.
11. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT.
12. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
13. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event with the Connection_Handle and has sent a successful HCI_Change_Connection_Link_Key_Complete event with the Connection_Handle to the Upper Tester.
14. The IUT continues sending ACL-U Data to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT pauses encryption to the Lower Tester using the LMP_PAUSE_ENCRYPTION_REQ PDU, if using E0 encryption, or the LMP_PAUSE_ENCRYPTION_AES_REQ PDU, if using AES-CCM encryption.

  The IUT sends no data packets to the Lower Tester while encryption is paused.

  The IUT resumes sending data packets to the Lower Tester after encryption is resumed.

- Notes

  The HCI_Change_Connection_Link_Key_Complete event may be sent any time after the HCI_Link_Key_Notification event.

4.6.2.4 Pausing and Resuming Encryption without Disabling Encryption_Mode – Peripheral Initiated with Role Switch

- Test Procedure

  Verify that the IUT as Peripheral can pause and resume encryption without disabling the Encryption_Mode as part of the role switch procedure.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition

  - The IUT is the Peripheral and the Lower Tester is the Central.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-4.
  - Both devices are sending data to the other device.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-17-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Peripheral Initiated with Role Switch]	E0
LMP/ENC/BV-42-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Peripheral Initiated with role switch]	AES-CCM

Table 4.6-4: Pausing and Resuming Encryption without Disabling Encryption_Mode – Peripheral Initiated with Role Switch test cases

- Test Procedure

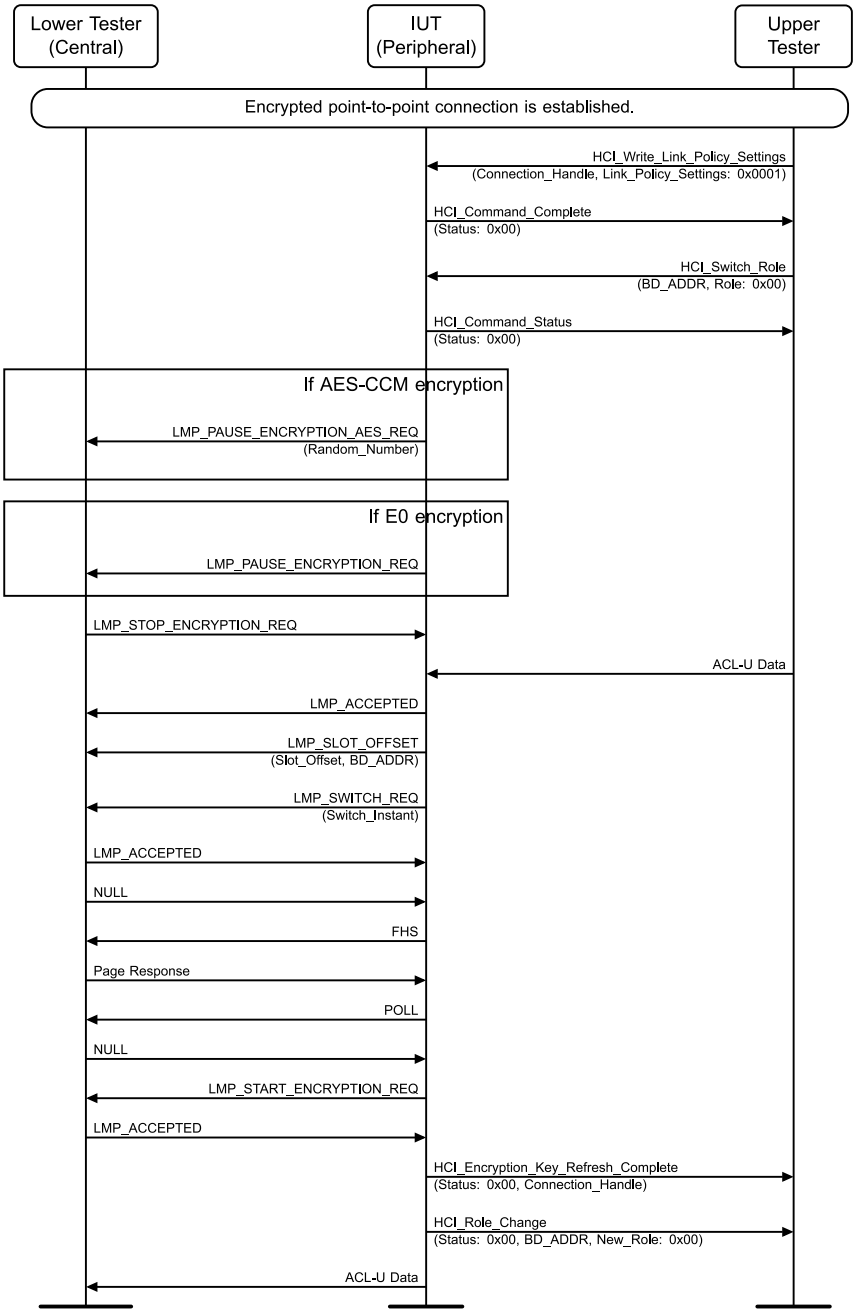


Figure 4.6-12: Pausing and Resuming Encryption without Disabling Encryption_Mode – Peripheral Initiated with Role Switch MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0001 (Enable Role Switch) and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
3. If AES-CCM encryption is being used as indicated in Table 4.6-4, the IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number; otherwise, the IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
4. The Lower Tester responds to the IUT with an LMP_STOP_ENCRYPTION_REQ PDU.

5. The Upper Tester sends ACL-U Data to the IUT.
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
7. The IUT sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the Lower Tester.
8. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
9. The Lower Tester sends a NULL packet to the IUT.
10. The IUT sends an FHS packet to the Lower Tester.
11. The Lower Tester sends a Page Response packet to the IUT.
12. The IUT sends a POLL packet to the Lower Tester.
13. The Lower Tester sends a NULL packet to the IUT.
14. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
15. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
16. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event with the Connection_Handle and a successful HCI_Role_Change event with the BD_ADDR and New_Role set to 0x00 (Central) to the Upper Tester.
17. The IUT continues sending ACL-U Data to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT pauses encryption to the Lower Tester using the LMP_PAUSE_ENCRYPTION_REQ PDU, if using E0 encryption, or the LMP_PAUSE_ENCRYPTION_AES_REQ PDU, if using AES-CCM encryption.

  The IUT sends no data packets while encryption is paused.

  The IUT sends data packets after encryption is resumed.

  The role switch succeeds.

- Notes

  The HCI_Role_Change event may be received any time after role switch.

4.6.2.5 Initiate Encryption

- Test Purpose

  Verify that the IUT initiates the encryption procedure and uses the encryption only for point-to-point messages.

- Reference

  [1] 4.2.5

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection is established.
  - Creation of a link key is successful.

- The IUT is the Peripheral and the Lower Tester is the Central.
- If AES-CCM encryption is to be used, the IUT and Lower Tester both indicate support for Secure Connections both in the Controller and the Host in their LMP Features.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-22-C [Initiate Encryption]	E0
LMP/ENC/BV-25-C [Initiate AES-CCM Encryption]	AES-CCM

Table 4.6-5: Initiate Encryption test cases

- Test Procedure

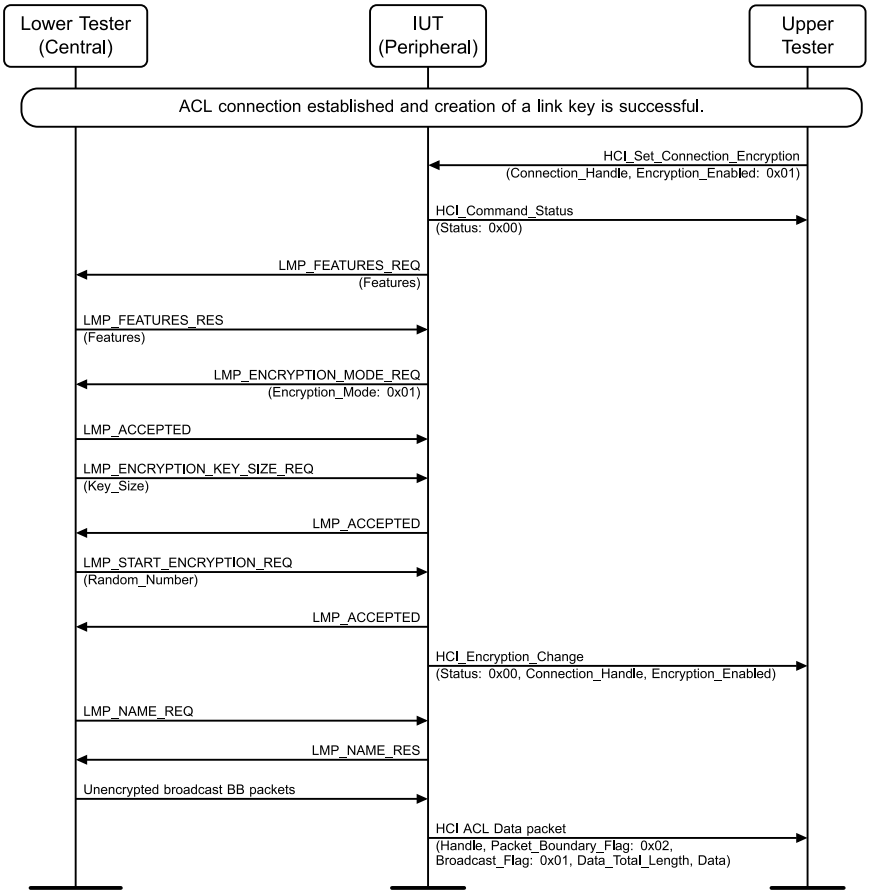


Figure 4.6-13: Initiate Encryption MSC

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status event in return.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
3. The Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with the Features parameter.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.

5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
6. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
8. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
10. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01, if using E0 encryption, or 0x02, if using AES-CCM encryption, as indicated in Table 4.6-5.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.
13. The Lower Tester sends unencrypted broadcast BB packets to the IUT.
14. The IUT sends an HCI ACL Data packet to the Upper Tester.

- Expected Outcome

  Pass verdict

  The LMP_FEATURES_REQ PDU is sent at least once by the IUT before starting the encryption.

  The IUT initiates the encryption negotiation and uses the encryption afterwards.

  In Step 10, the Encryption_Enabled parameter of the HCI_Encryption_Change event is set to 0x01 if using E0 encryption or 0x02 if using AES-CCM encryption.

  In Step 12, the IUT sends the LMP_NAME_RES PDU and proves that encryption is used.

  In Step 14, the IUT passes on broadcast ACL traffic with non-encrypted payloads from the Lower Tester to the Upper Tester.

LMP/ENC/BV-69-C [Initiate Encryption]

- Test Purpose

  Verify that the IUT initiates the encryption procedure and uses encryption only for point-to-point messages. The IUT is Peripheral and the Lower Tester is Central.

- Reference

  [1] 4.2.5

- Initial Condition

  - See the “Default settings” section.

- Test Procedure

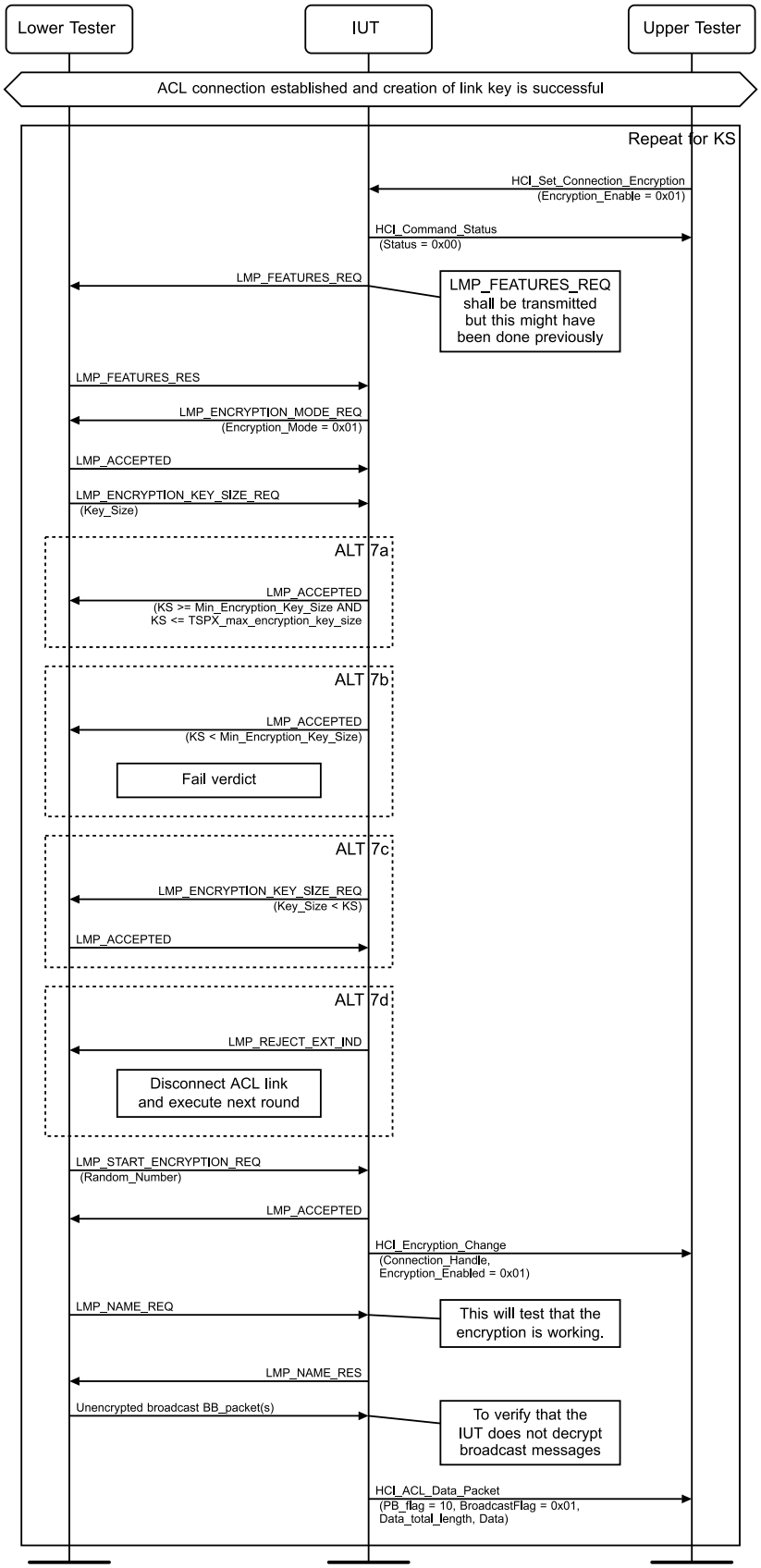


Figure 4.6-14: LMP/ENC/BV-69-C [Initiate Encryption] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester.
3. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
5. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
6. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with Key_Size set to KS.
7. The IUT executes either 7a, 7b, 7c, or 7d.
   a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
   b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
   c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
   d. Reject the suggested Key_Size, in which case skip to Step 15.
8. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
9. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
10. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x01.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
13. The Lower Tester starts sending Unencrypted broadcast BB packets to the IUT.
14. The IUT sends an HCI_ACL_Data_Packet event to the Upper Tester with PB_Flag set to 10, BroadcastFlag set to 0x01, Data_total_length, and Data.
15. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The PDU LMP_FEATURES_REQ has to be sent at least once by the IUT before starting the encryption.

  The IUT initiates the encryption negotiation and uses the encryption afterwards.

  The IUT responds correctly to the PDU LMP_NAME_REQ to prove that encryption is used.

  The IUT passes on broadcast ACL traffic with non-encrypted payloads from the Lower Tester to the Upper Tester.

  The Encryption_Enabled Parameter in the Encryption Change event reports encryption has been enabled.

  Fail verdict

  In Step 7b, the IUT accepts a key size < Min_Encryption_Key_Size.

  In Step 7c, the IUT suggests a key size < 7.

- Notes

  The suggested Key_Size must be within the Lower Tester's Key_Size range.

LMP/ENC/BV-70-C [Initiate AES-CCM Encryption]

- Test Purpose

  Verify that the IUT initiates the encryption procedure and uses AES-CCM encryption only for point-to-point messages when the remote controller's LMP feature bits indicate support for Secure Connections both in the Controller and Host.

- Reference

  [1] 4.2.5

- Initial Condition

  - The Lower Tester is Central and the IUT is Peripheral.
  - An ACL connection has been established between the IUT and the Lower Tester, and the creation of a link key between the IUT and the Lower Tester has been successful.
  - The Lower Tester LMP feature bits have support for Secure Connections both in the Controller and the Host.

- Test Procedure

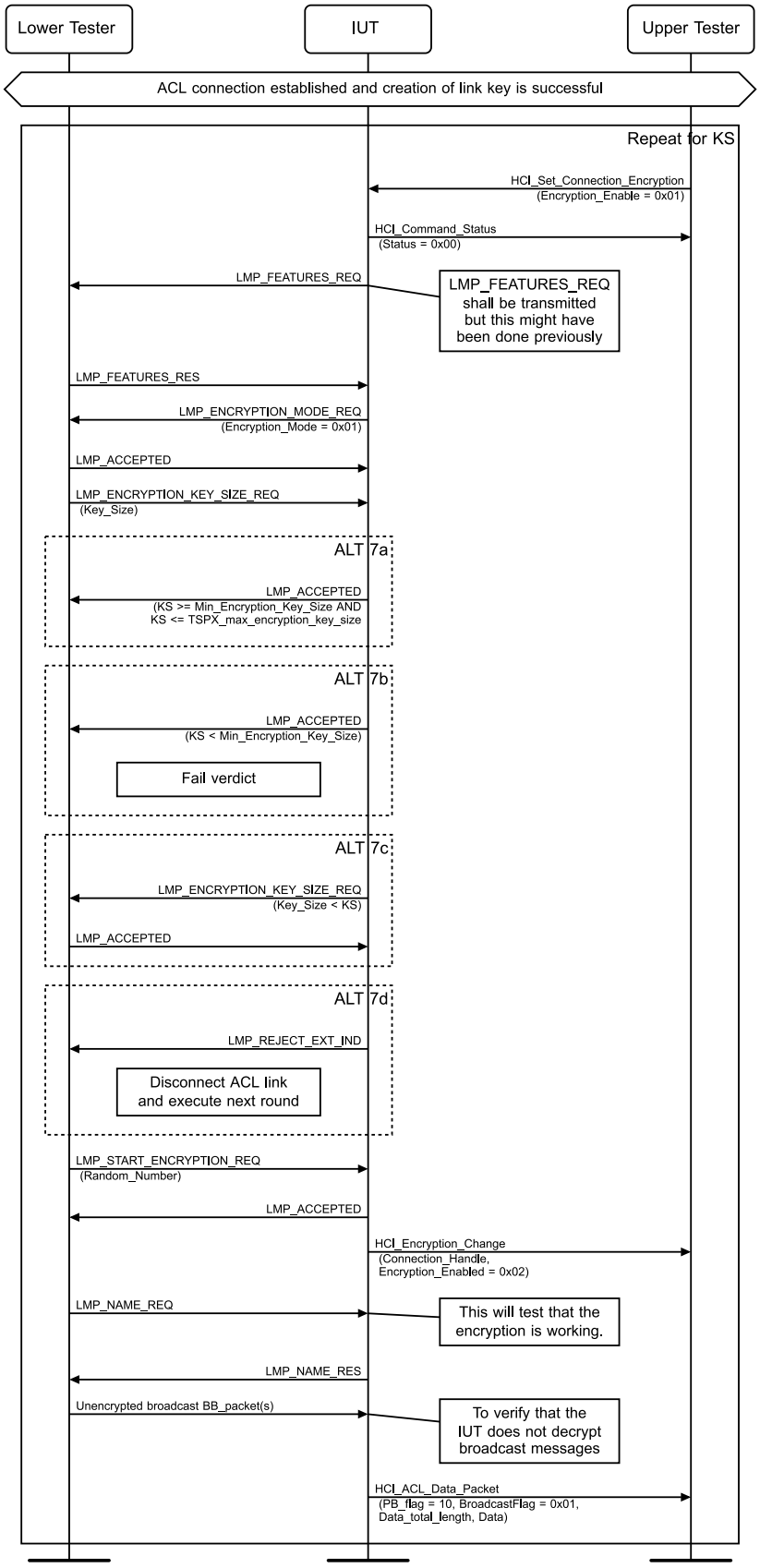


Figure 4.6-15: LMP/ENC/BV-70-C [Initiate AES-CCM Encryption] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with feature bits set for Secure Connections for both the Controller and the Host.
3. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
5. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
6. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with Key_Size set to KS.
7. The IUT executes either 7a, 7b, 7c, or 7d.
    a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
    b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
    c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
    d. Reject the suggested Key_Size, in which case skip to Step 15.
8. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
9. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
10. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x01.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
13. The Lower Tester starts sending Unencrypted broadcast BB packets to the IUT.
14. The IUT sends an HCI_ACL_Data_Packet event to the Upper Tester with PB_Flag set to 10, BroadcastFlag set to 0x01, Data_total_length, and Data.
15. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT initiates the encryption negotiation and uses the encryption afterwards.

  The IUT responds correctly to the PDU LMP_NAME_REQ (this proves that encryption is used).

  The IUT sends HCI ACL Data with non-encrypted payload to the Upper Tester.

  If Read Encryption Key_Size is supported, the IUT returns the Key_Size parameter from the LMP_ENCRYPTION_KEY_SIZE_REQ PDU in the Command Complete event following the HCI Read Encryption Key_Size command.

  The Encryption_Enabled Parameter in the Encryption Change event reports AES-CCM encryption has been enabled.

  Fail verdict

  In Step 7b, the IUT accepts a key size < Min_Encryption_Key_Size.

  In Step 7c, the IUT suggests a key size < 7.

- Notes

  If the IUT starts to negotiate for encryption Key_Size the Lower Tester must negotiate.

  The suggested Key_Size must be within the Lower Tester's Key_Size range.

4.6.2.6 Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated as a result of change connection link key

- Test Purpose

  Verify that the IUT as the Peripheral can respond to a Central-initiated pause and resume of encryption without disabling the Encryption_Mode as part of the change connection link key procedure.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5

- Initial Condition

  - The Lower Tester is the Central and the IUT is the Peripheral.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-6.
  - Both devices are sending data to each other.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-23-C [Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated as a result of change connection link Key]	E0
LMP/ENC/BV-40-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Central Initiated as a result of change connection link Key]	AES-CCM

Table 4.6-6: Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated as a result of change connection link key test cases

- Test Procedure

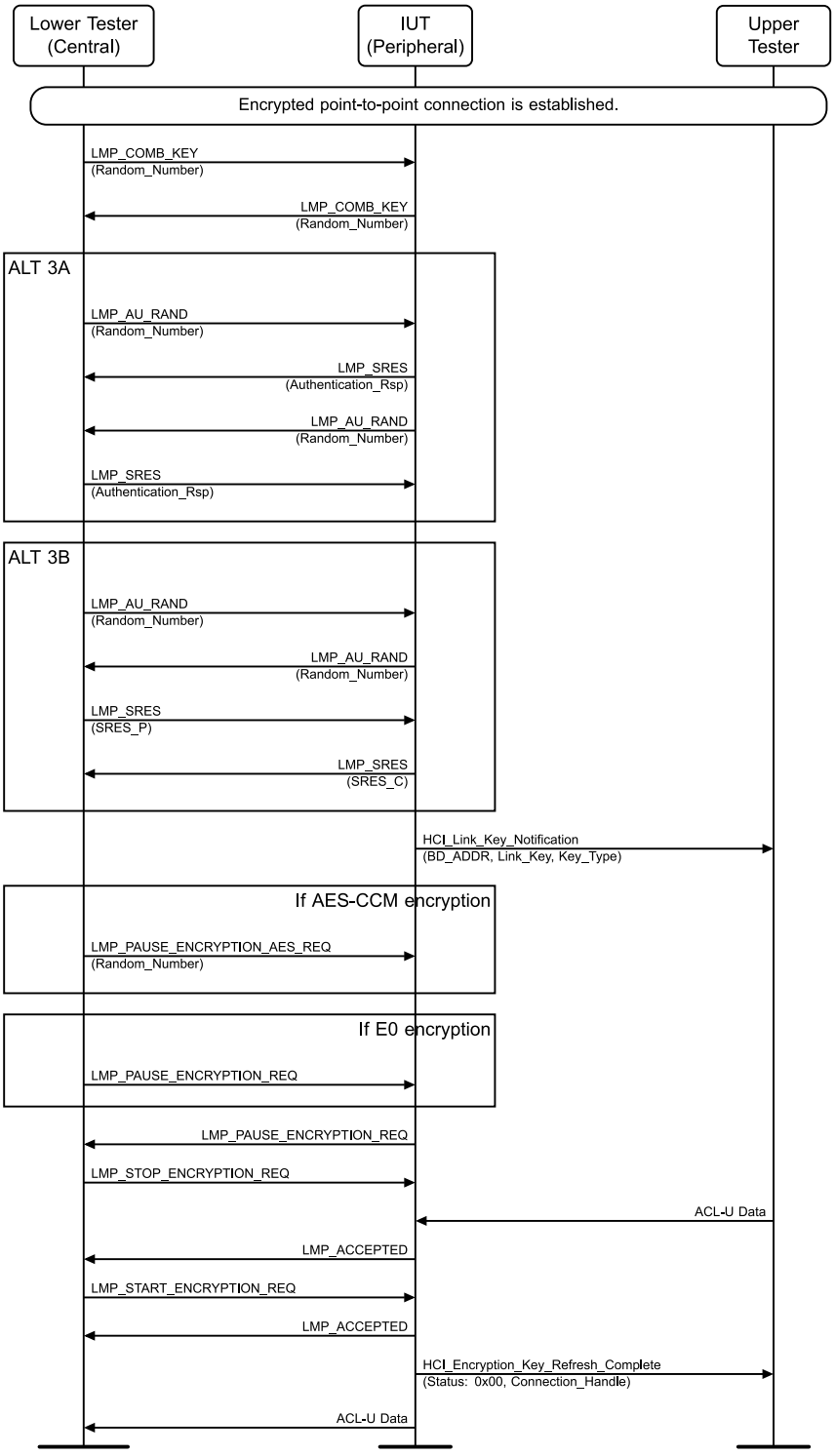


Figure 4.6-16: Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated as a result of change connection link key MSC

1. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
2. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.

3. Perform either alternative 3A or 3B depending on the encryption indicated in Table 4.6-6.
   Alternative 3A (The connection uses E0 encryption):
   - 3A.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
   - 3A.2. The IUT responds to the Lower Tester with an LMP_SRES PDU with the Authentication_Rsp.
   - 3A.3. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
   - 3A.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.

   Alternative 3B (The connection uses AES-CCM encryption):
   - 3B.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
   - 3B.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.
   - 3B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_P.
   - 3B.4. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_C.
4. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type set to 0x00 (Combination Key), if E0 encryption is being used, or 0x06, if AES-CCM encryption is being used.
5. If AES-CCM encryption is being used as indicated in Table 4.6-6, the IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number; otherwise, the IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
6. The IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
7. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
8. The Upper Tester sends ACL-U Data to the IUT.
9. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
10. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
11. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
12. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.
13. The IUT sends the ACL-U Data to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 6, the IUT sends the LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester in response to the LMP_PAUSE_ENCRYPTION_REQ PDU or the LMP_PAUSE_ENCRYPTION_AES_REQ PDU.

  The IUT sends no data packets while encryption is paused.

  The IUT sends data packets after encryption is resumed.

4.6.2.7 Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated with role switch

- Test Purpose

  Verify that the IUT as the Peripheral can respond to a Central-initiated pause and resume of encryption without disabling the Encryption_Mode as part of the role switch procedure.

- Reference

  [1] 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition

  - The Lower Tester is the Central and the IUT is the Peripheral.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-7.
  - Both devices are sending data to each other.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-24-C [Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated with role switch]	E0
LMP/ENC/BV-44-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Peripheral, Central Initiated with role switch]	AES-CCM

Table 4.6-7: Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated with role switch test cases

- Test Procedure

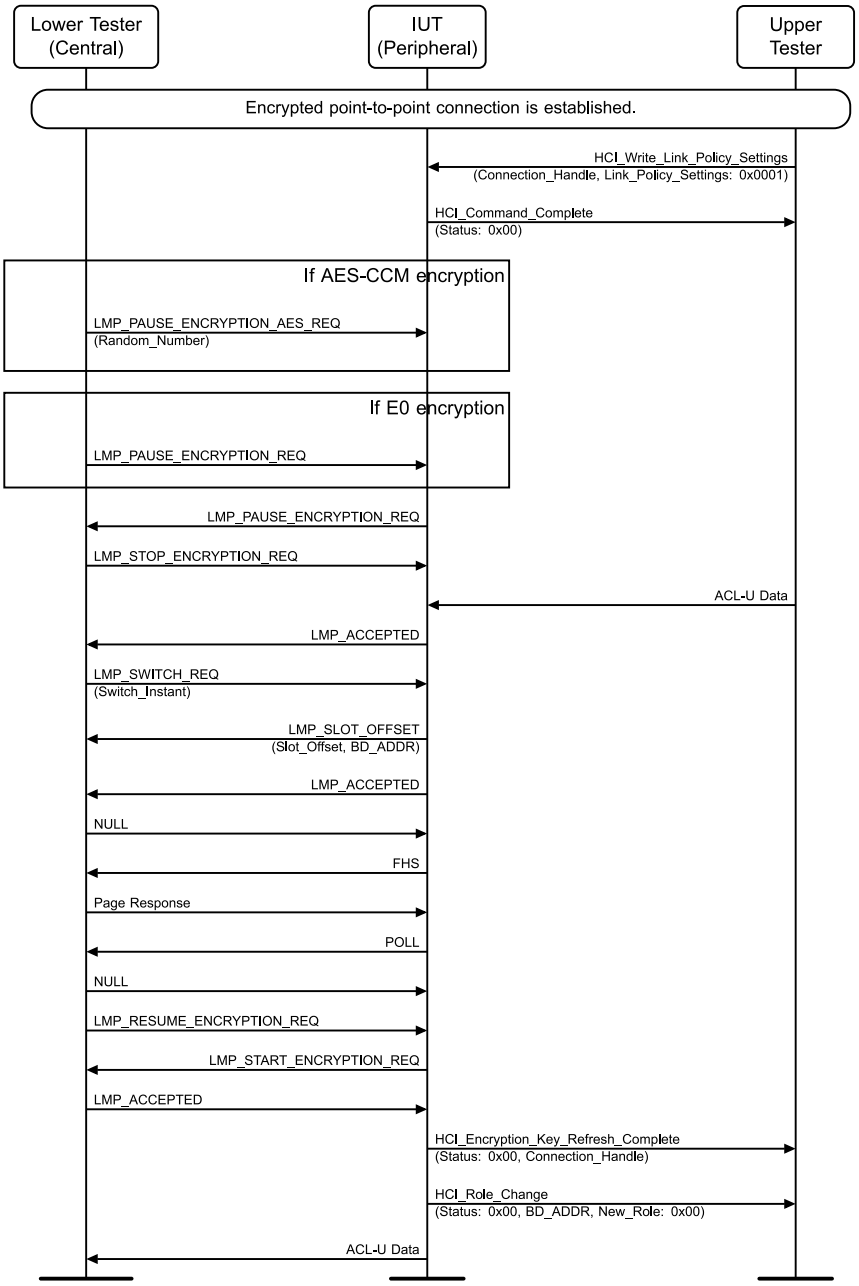


Figure 4.6-17: Pausing and Resuming without Disabling Encryption, IUT Peripheral, Central Initiated with role switch MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command with the Connection Handle and Link Policy Settings set to 0x00001 (Enable Role Switch) and receives a successful HCI_Command_Complete event.
2. If AES-CCM encryption is being used as indicated in Table 4.6-7, the Lower Tester sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the IUT with a Random_Number; otherwise, the Lower Tester sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the IUT.
3. The IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
4. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
5. The Upper Tester sends ACL-U Data to the IUT.

6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
7. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
8. The IUT sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and the BD_ADDR.
9. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
10. The Lower Tester sends a NULL packet to the IUT.
11. The IUT sends an FHS packet to the Lower Tester.
12. The Lower Tester sends a Page Response to the IUT.
13. The IUT sends a POLL packet to the Lower Tester.
14. The Lower Tester sends a NULL packet to the IUT.
15. The Lower Tester sends an LMP_RESUME_ENCRYPTION_REQ PDU to the IUT.
16. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
17. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
18. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event with the Connection_Handle and a successful HCI_Role_Change event with the BD_ADDR and New_Role set to 0x00 (Central) to the Upper Tester.
19. The IUT sends ACL-U Data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT sends the LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester in response to the LMP_PAUSE_ENCRYPTION_REQ PDU or the LMP_PAUSE_ENCRYPTION_AES_REQ PDU.

  The IUT transmits no data packets while encryption is paused.

  The IUT transmits data packets after encryption is resumed.

  The role switch succeeds.

4.6.2.8 Key_Size Negotiation as Peripheral

- Test Purpose

  Verify that the IUT in the Peripheral role correctly reports the negotiated encryption Key_Size.

- Reference

  [1] 4.2.5

- Initial Condition

  - The TSPX_min_supported_encryption_key_size IXIT statement gives the value for the minimum encryption Key_Size.
  - The TSPX_max_supported_encryption_key_size IXIT statement gives the value for the maximum encryption Key_Size.
  - The Lower Tester has a minimum encryption Key_Size set to 1.
  - See Initial Conditions in Table 4.6-8.

- Test Case Configuration

Test Case	Initial Condition	HCI Set Min Encryption Key Size Support	IUT is Initiator	Encryption Type	Encryption _Enabled (Step 9)
LMP/ENC/BV-51-C [Key_Size Negotiation as Peripheral - E0, Acceptor]	Section 4.2.3 Encryption	Yes	No	E0	0x01
LMP/ENC/BV-52-C [Key_Size Negotiation as Peripheral - AES, Acceptor]	Section 4.2.7 AES-CCM Encryption	Yes	No	AES	0x02
LMP/ENC/BV-55-C [Key_Size Negotiation as Peripheral - E0, Initiator]	Section 4.2.3 Encryption	Yes	Yes	E0	0x01
LMP/ENC/BV-56-C [Key_Size Negotiation as Peripheral - AES, Initiator]	Section 4.2.7 AES-CCM Encryption	Yes	Yes	AES	0x02
LMP/ENC/BV-59-C [Key_Size Negotiation as Peripheral - E0, Acceptor]	Section 4.2.3 Encryption	No	No	E0	0x01
LMP/ENC/BV-60-C [Key_Size Negotiation as Peripheral - AES, Acceptor]	Section 4.2.7 AES-CCM Encryption	No	No	AES	0x02
LMP/ENC/BV-61-C [Key_Size Negotiation as Peripheral - E0, Initiator]	Section 4.2.3 Encryption	No	Yes	E0	0x01
LMP/ENC/BV-62-C [Key_Size Negotiation as Peripheral - AES, Initiator]	Section 4.2.7 AES-CCM Encryption	No	Yes	AES	0x02

Table 4.6-8: Key_Size Negotiation as Peripheral test cases

- Test Procedure

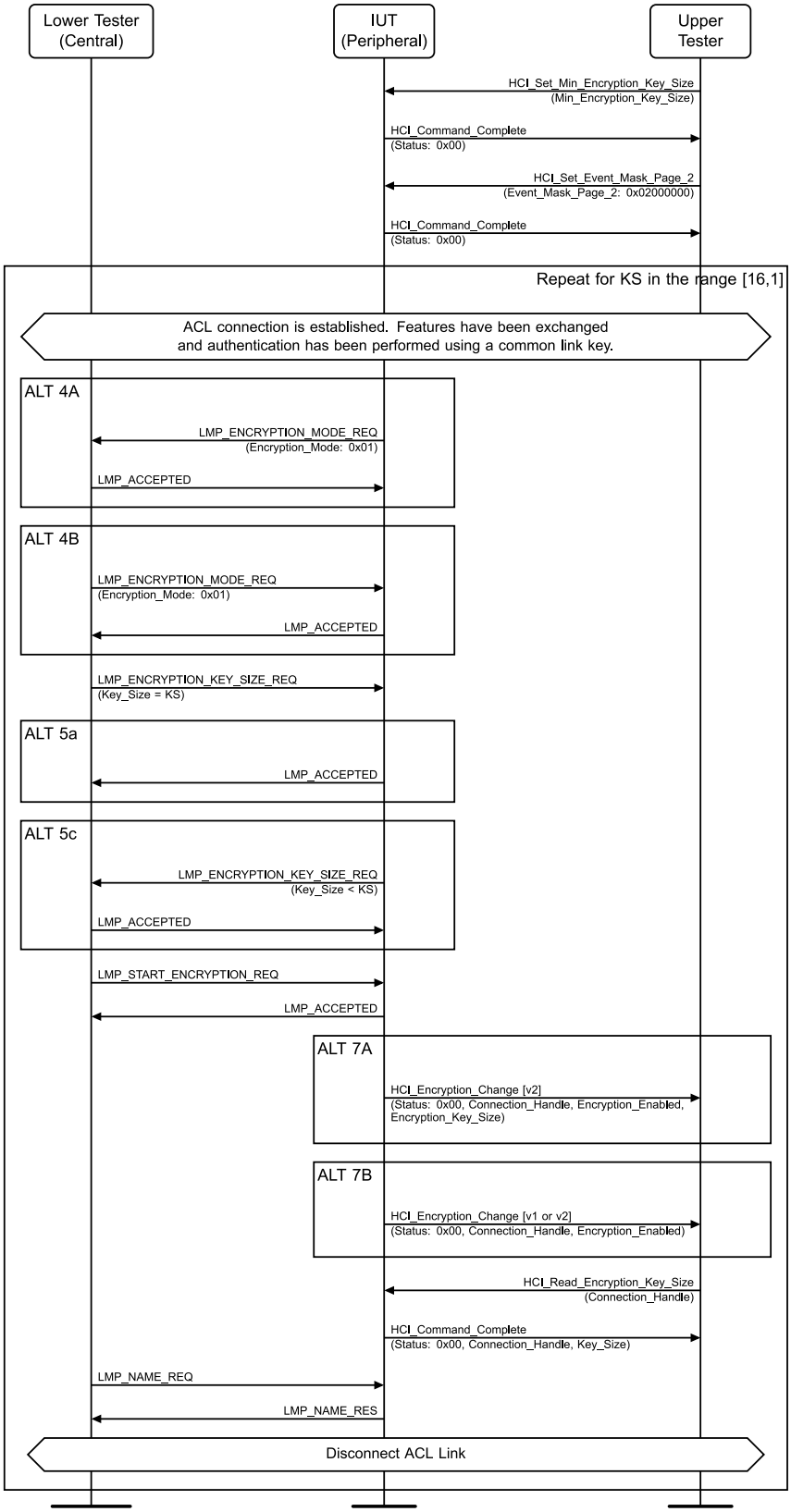


Figure 4.6-18: Key_Size Negotiation as Peripheral MSC

Repeat Steps 2–10 for each encryption Key_Size value KS in the interval [16, 1].

1. Perform either alternative 1A or 1B depending on the IUT's support for HCI_Set_Min_Encryption_Key_Size as specified in Table 4.6-8.
   Alternative 1A (The IUT supports HCI_Set_Min_Encryption_Key_Size):
   - 1A.1. The Upper Tester sends the HCI_Set_Min_Encryption_Key_Size command with the Min_Encryption_Key_Size set to INT((TSPX_min_supported_encryption_key_size + TSPX_max_supported_encryption_key_size) / 2).
   - 1A.2. The Upper Tester sends an HCI_Set_Event_Mask_Page_2 command to the IUT with the Event_Mask_Page_2 set to 0x02000000 and receives a successful HCI_Command_Complete in return.

   Alternative 1B (The IUT does not support HCI_Set_Min_Encryption_Key_Size):
   - 1B.1. The Min_Encryption_Key_Size is considered the same as TSPX_min_supported_encryption_key_size for the rest of the test case below.
2. Establish an ACL connection between the IUT and the Lower Tester.
3. The Lower Tester initiates the exchange of all supported Features (LMP_FEATURES_REQ PDU and, if relevant, LMP_FEATURES_REQ_EXT PDU). The Lower Tester indicates Secure Connections support for both Host and controller only when AES encryption is indicated in Table 4.6-8.
4. Perform either alternative 4A or 4B depending on the IUT role in Table 4.6-8.
   Alternative 4A (IUT is Initiator):
   - 4A.1. The Upper Tester initiates authentication using a random link key known to the IUT and the Upper Tester.
   - 4A.2. The Upper Tester orders the IUT to enable link encryption, and the IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
   - 4A.3. The Lower Tester replies to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.

   Alternative 4B (IUT is Responder):
   - 4B.1. The Lower Tester initiates authentication using a random link key known to the IUT and the Upper Tester.
   - 4B.2. The Lower Tester begins the link encryption procedure by sending an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
   - 4B.3. The IUT replies to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
5. The Lower Tester sends the LMP_ENCRYPTION_KEY_SIZE_REQ PDU with Key_Size set to KS. The IUT does one of the following:
   a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
   b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
   c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
   d. Reject the suggested Key_Size, in which case skip to Step 10.
6. The Lower Tester continues the link encryption procedure by sending an LMP_START_ENCRYPTION_REQ PDU, and the IUT replies with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

7. Perform either alternative 7A or 7B depending on the IUT's support for HCI_Set_Min_Encryption_Key_Size as specified in Table 4.6-8.
   Alternative 7A (The IUT supports HCI_Set_Min_Encryption_Key_Size):
   7A.1. The IUT sends an HCI_Encryption_Change [v2] event to the Upper Tester with the Status field set to 0x00, the Encryption_Key_Size set to the value negotiated in Step 5, and the Encryption_Enabled field set to the value indicated in Table 4.6-8.
   Alternative 7B (The IUT does not support HCI_Set_Min_Encryption_Key_Size):
   7B.1. The IUT sends either an HCI_Encryption_Change [v1] or an HCI_Encryption_Change [v2] event to the Upper Tester with the Status field set to 0x00 and the Encryption_Enabled field set to the value indicated in Table 4.6-8.
8. The Upper Tester sends an HCI_Read_Encryption_Key_Size command to the IUT and receives a successful HCI_Command_Complete event in response with the Key_Size parameter equal to the negotiated Key_Size in Step 5.
9. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT, and the IUT replies with an LMP_NAME_RES PDU, verifying that the encryption uses the negotiated Key_Size.
10. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  In Step 8, the IUT correctly reports the negotiated encryption Key_Size for each accepted Key_Size value, if the HCI_Read_Encryption_Key_Size command is supported.

  At least one Key_Size value is accepted by the IUT.

  If the HCI_Set_Min_Encryption_Key_Size command is supported, then each accepted Key_Size value in Step 7A.1 >= the Min_Encryption_Key_Size from Step 1.

LMP/ENC/BI-01-C [Encryption, Peripheral, Reject Role Switch]

- Test Purpose

  Verify that the IUT rejects a role switch request during the Encryption process.

- Reference

  [1] 4.2.5, 4.4.2

- Initial Condition

  - See the "Connection Establishment Lower Tester" preamble.
  - The Lower Tester supports role switch.
  - The Lower Tester initiates authentication and optionally expects a mutual authentication.
  - The Lower Tester uses an acceptable Key length.
  - The Lower Tester is the Central and the IUT is the Peripheral.

- Test Procedure

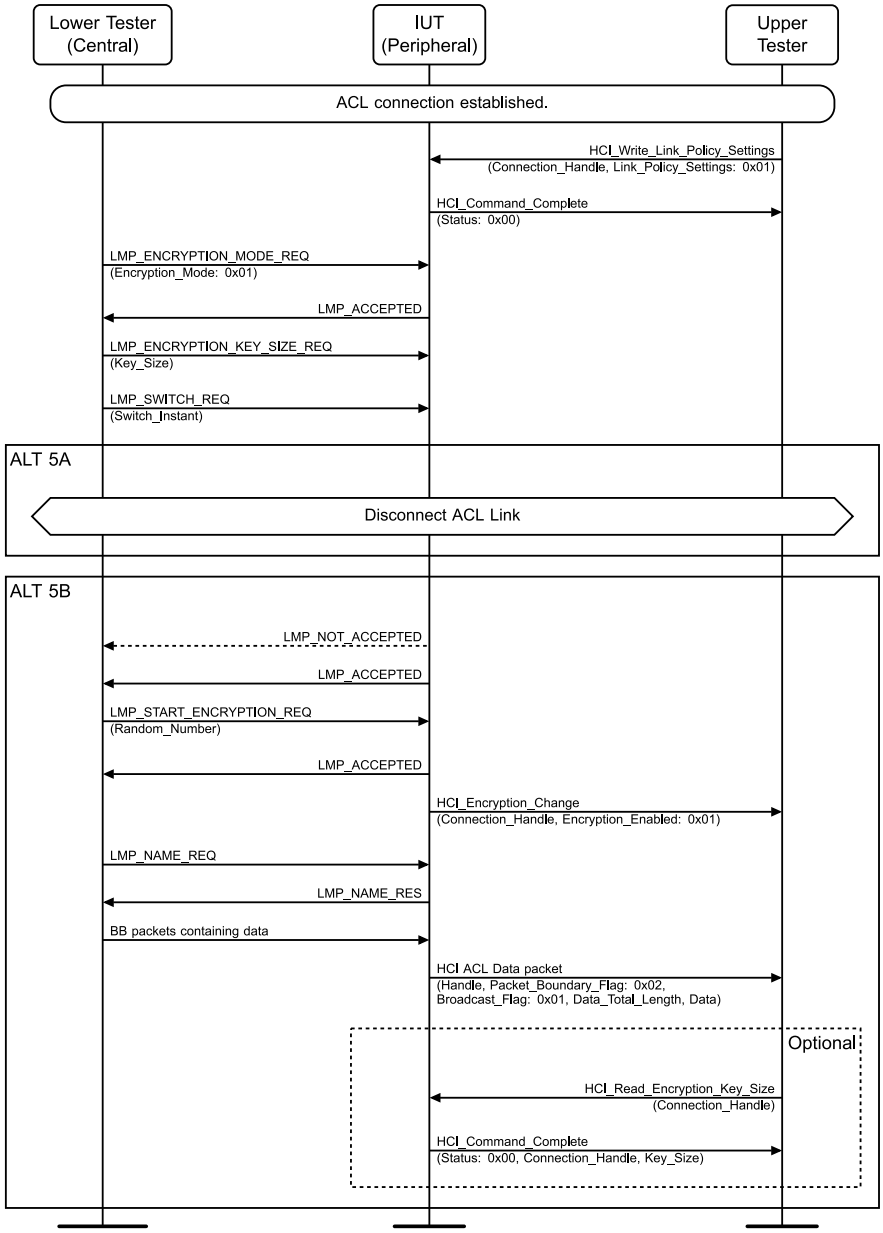


Figure 4.6-19: LMP/ENC/BI-01-C [Encryption, Peripheral, Reject Role Switch] MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x01 and receives a successful HCI_Command_Complete event in response.
2. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01.
3. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
4. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU with the Key_Size and an LMP_SWITCH_REQ PDU with the Switch_Instant.

5. Perform either alternative 5A or 5B depending on the IUT's response.
   Alternative 5A (The IUT disconnects the ACL Link):
   5A.1. The IUT disconnects the ACL Link.
   Alternative 5B (The IUT continues with the encryption procedure):
   5B.1. Optionally, the IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
   5B.2. If the IUT responds to the Lower Tester with an LMP_ENCRYPTION_KEY_SIZE_REQ PDU with a new Key_Size, the Lower Tester and the IUT continue key size negotiation until Step 5B.3 occurs.
   5B.3. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
   5B.4. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
   5B.5. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
   5B.6. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.
   5B.7. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
   5B.8. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.
   5B.9. The Lower Tester sends BB packets containing data to the IUT.
   5B.10. The IUT sends an HCI ACL Data packet to the Upper Tester.
   5B.11. If supported by the IUT, the Upper Tester sends an HCI_Read_Encryption_Key_Size command to the IUT with the Connection_Handle and receives a successful HCI_Command_Complete event with the Connection_Handle and Key_Size.

- Expected Outcome

  Pass verdict

  In alternative 5A, the IUT disconnects the ACL Link.

  In alternative 5B, the IUT optionally sends an LMP_NOT_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester. The IUT sends an LMP_ACCEPTED PDU to the Start Encryption Request.

LMP/ENC/BI-03-C [Reject Start Encryption Request, Encrypted Connection, Peripheral]

- Test Purpose

  Verify that the IUT as the Peripheral rejects a Start Encryption Request when the IUT and the Lower Tester have an encrypted connection.

- Reference

  [1] 4.2.5.3

- Initial Condition

  - The Lower Tester is the Central and the IUT is the Peripheral.
  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester.

- Test Procedure

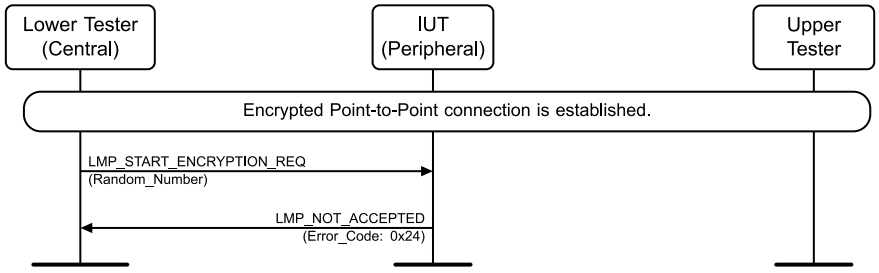


Figure 4.6-20: LMP/ENC/BI-03-C [Reject Start Encryption Request, Encrypted Connection, Peripheral] MSC

1. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
2. The IUT responds with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x24 (LMP PDU Not Allowed).

- Expected Outcome

  Pass verdict

  The IUT responds to the LMP_START_ENCRYPTION_REQ PDU with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x24 (LMP PDU Not Allowed).

LMP/ENC/BV-19-C [Stopping and Restarting Encryption with Legacy Device - Peripheral Initiated]

- Test Procedure

  Verify that the IUT as the Peripheral can stop and restart encryption with a device that does not support Pause Encryption.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition

  - The IUT is the Peripheral and the Lower Tester is the Central.
  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester.
  - The Lower Tester is a device that does not support Encryption Pause Resume.

- Test Procedure

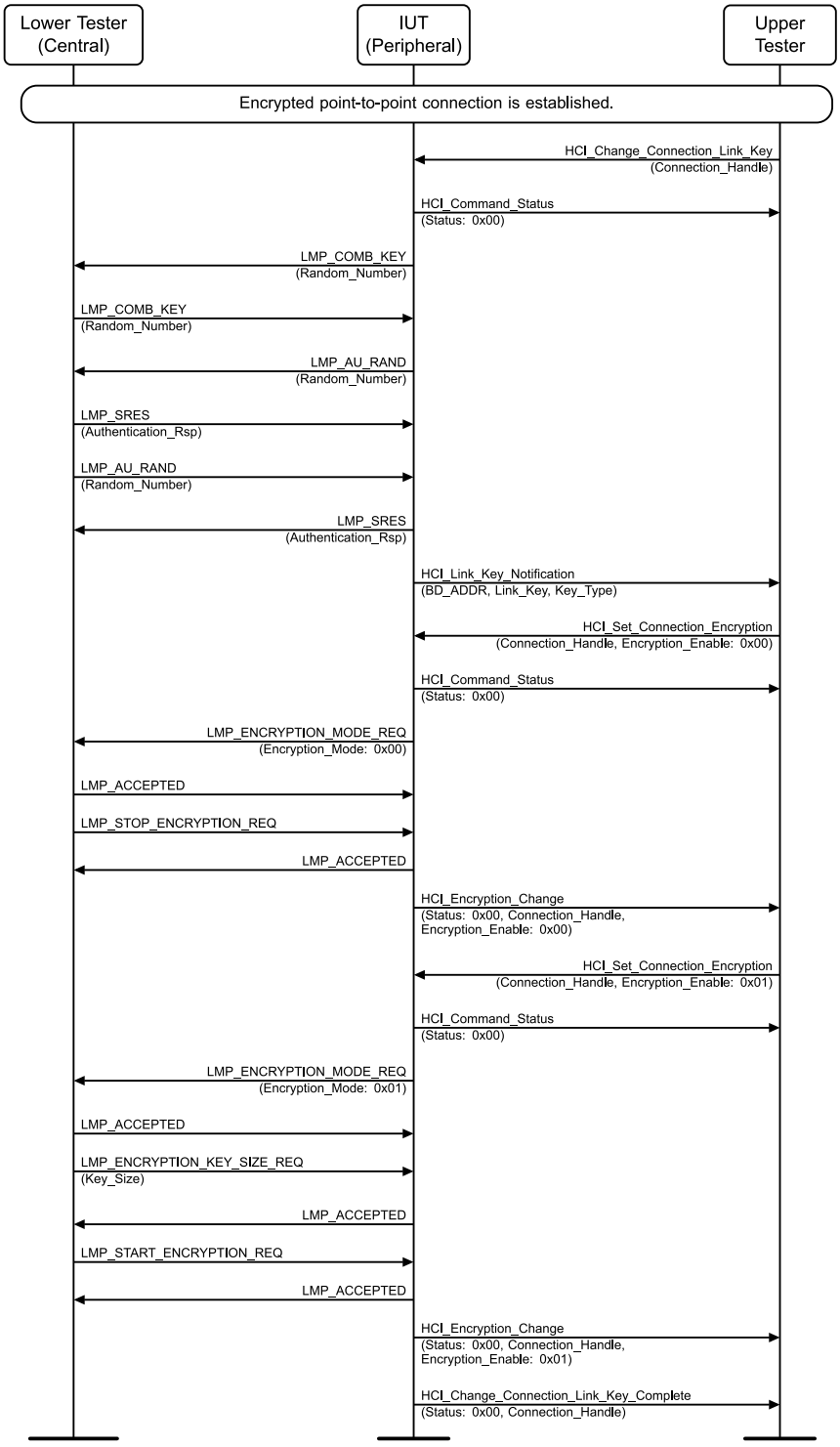


Figure 4.6-21: LMP/ENC/BV-19-C [Stopping and Restarting Encryption with Legacy Device - Peripheral Initiated] MSC

1. The Upper Tester sends an HCI_Change_Connection_Link_Key command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.

4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.
6. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
7. The IUT responds to the Lower Tester with an LMP_SRES PDU with the Authentication_Rsp.
8. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with the BD_ADDR, Link_Key, and Key_Type.
9. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with Encryption_Enable set to 0x00 and receives a successful HCI_Command_Status event in response.
10. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x00.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
12. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
13. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
14. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enable set to 0x00.
15. The Upper Tester sends an HCI_Set_Connection_Encryption PDU to the IUT with Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status event in response.
16. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
17. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
18. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.
19. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
20. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
22. The IUT sends a successful HCI_Encryption_Change event with the Connection_Handle and Encryption_Enable set to 0x01 and has sent a successful HCI_Change_Connection_Link_Key event with the Connection_Handle to the Upper Tester.

- Expected Outcome

  Pass verdict

  Encryption is stopped and then restarted.

- Notes

  The HCI_Change_Connection_Link_Key_Complete event may be received any time after the HCI_Link_Key_Notification event.

4.6.2.9 Key_Size Negotiation as Peripheral

- Test Purpose

  Verify that the IUT in the Peripheral role correctly reports the negotiated encryption Key_Size.

- Reference

  [1] 4.2.5

- Initial Condition
  - An IXIT statement, TSPX_min_supported_encryption_key_size, gives the value for the minimum encryption Key_Size.
  - An IXIT statement, TSPX_max_supported_encryption_key_size, gives the value for the maximum encryption Key_Size.
  - The Lower Tester has a minimum encryption Key_Size set to 1.
  - See Initial Conditions in Table 4.6-9.
- Test Case Configuration

Test Case	Initial Condition	HCI Set Min Encryption Key Size Support	IUT is Initiator	Encryption Type	Encryption _Enabled (Step 9)
LMP/ENC/BV-73-C [Key_Size Negotiation as Peripheral - E0, Acceptor]	See the "Default Settings: Encryption" section.	Yes	No	E0	0x01
LMP/ENC/BV-74-C [Key_Size Negotiation as Peripheral - AES, Acceptor]	See the "Default Settings: AES-CCM Encryption" section.	Yes	No	AES	0x02
LMP/ENC/BV-75-C [Key_Size Negotiation as Peripheral - E0, Initiator]	See the "Default Settings: Encryption" section.	Yes	Yes	E0	0x01
LMP/ENC/BV-76-C [Key_Size Negotiation as Peripheral - AES, Initiator]	See the "Default Settings: AES-CCM Encryption" section.	Yes	Yes	AES	0x02
LMP/ENC/BV-77-C [Key_Size Negotiation as Peripheral - E0, Acceptor]	See the "Default Settings: Encryption" section.	No	No	E0	0x01
LMP/ENC/BV-78-C [Key_Size Negotiation as Peripheral - AES, Acceptor]	See the "Default Settings: AES-CCM Encryption" section.	No	No	AES	0x02
LMP/ENC/BV-79-C [Key_Size Negotiation as Peripheral - E0, Initiator]	See the "Default Settings: Encryption" section.	No	Yes	E0	0x01

Test Case	Initial Condition	HCI Set Min Encryption Key Size Support	IUT is Initiator	Encryption Type	Encryption _Enabled (Step 9)
LMP/ENC/BV-80-C [Key_Size Negotiation as Peripheral - AES, Initiator]	See the "Default Settings: AES-CCM Encryption" section.	No	Yes	AES	0x02

Table 4.6-9: Key_Size Negotiation as Peripheral test cases

- Test Procedure

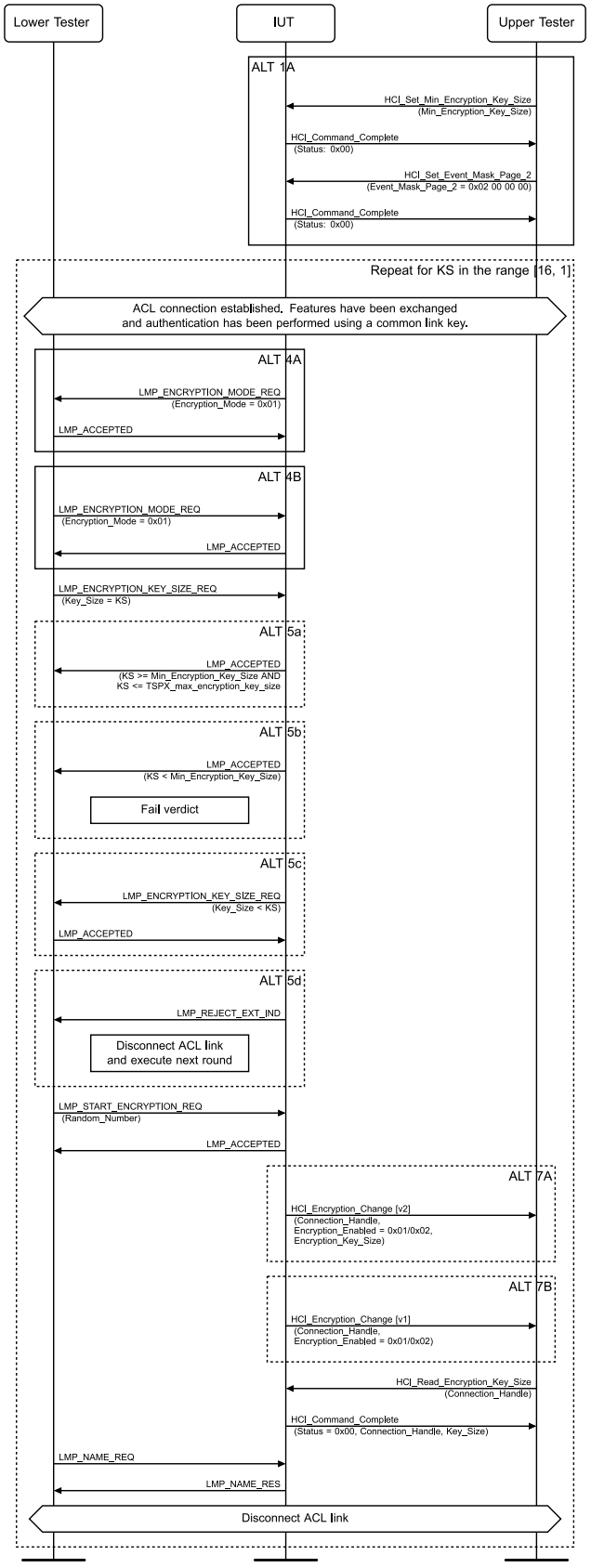


Figure 4.6-22: Key_Size Negotiation as Peripheral MSC

Repeat Steps 2–10 for each encryption Key_Size value KS in the interval [16, 1]:

1. Perform either alternative 1A or 1B depending on the IUT support for HCI Set Min Encryption Key Size as specified in Table 4.6-9.
   Alternative 1A (The IUT supports HCI Set Min Encryption Key Size):
   - 1A.1. The Upper Tester sends the HCI_Set_Min_Encryption_Key_Size command with the Min_Encryption_Key_Size set to INT((TSPX_min_supported_encryption_key_size + TSPX_max_supported_encryption_key_size) / 2).
   - 1A.2. The Upper Tester sends an HCI_Set_Event_Mask_Page_2 command to the IUT with the Event_Mask_Page_2 set to 0x02000000 and receives a successful HCI_Command_Complete in return.

   Alternative 1B (The IUT does not support HCI Set Min Encryption Key Size):
   - 1B.1. The Min_Encryption_Key_Size is considered the same as TSPX_min_supported_encryption_key_size for the rest of the test case below.
2. Establish an ACL connection between the IUT and the Lower Tester.
3. The Lower Tester initiates the exchange of all supported Features (LMP_FEATURES_REQ PDU and, if relevant, LMP_FEATURES_REQ_EXT). The Lower Tester indicates Secure Connections support for both Host and controller only when AES encryption is indicated in Table 4.6-9.
4. Perform either alternative 4A or 4B depending on the IUT role in Table 4.6-9.
   Alternative 4A (IUT is Initiator):
   - 4A.1. The Upper Tester initiates authentication using a random link key known to the IUT and the Upper Tester.
   - 4A.2. The Upper Tester orders the IUT to enable link encryption, and the IUT sends to the Lower Tester an LMP_ENCRYPTION_MODE_REQ PDU with the encryption_mode field set to 0x01, and the Lower Tester replies with an LMP_ACCEPTED PDU.

   Alternative 4B (IUT is Responder):
   - 4B.1. The Lower Tester initiates authentication using a random link key known to the IUT and the Upper Tester.
   - 4B.2. The Lower Tester begins the link encryption procedure by sending an LMP_ENCRYPTION_MODE_REQ PDU with the encryption_mode field set to 0x01, and the IUT replies with an LMP_ACCEPTED PDU.
5. The Lower Tester sends the LMP_ENCRYPTION_KEY_SIZE_REQ PDU suggesting a Key_Size equal to KS. The IUT may:
   a. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
   b. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
   c. Suggest a lower Key_Size equal to TSPX_max_encryption_key_size, which the Lower Tester will accept that is greater than or equal to 7. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
   d. Reject the suggested Key_Size, in which case skip to Step 10.
6. The Lower Tester continues the link encryption procedure by sending an LMP_START_ENCRYPTION_REQ PDU and the IUT replies with an LMP_ACCEPTED PDU.
7. Perform either alternative 7A or 7B depending on the IUT support for HCI Set Min Encryption Key Size as specified in Table 4.6-9.
   Alternative 7A (The IUT supports HCI Set Min Encryption Key Size):
   - 7A.1. The IUT sends to the Upper Tester an HCI_Encryption_Change [v2] event with the Status field set to 0x00, the Encryption_Key_Size set to the value negotiated in Step 5, and the Encryption_Enabled field set to the value indicated in Table 4.6-9 for this test.

Alternative 7B (The IUT does not support HCI Set Min Encryption Key Size):

7B.1. The IUT sends to the Upper Tester either an HCI_Encryption_Change [v1] or an HCI_Encryption_Change [v2] event with the Status field set to 0x00 and the Encryption_Enabled field set to the value indicated in Table 4.6-9 for this test.

8. The Upper Tester issues the HCI Read Encryption Key_Size command to the IUT, and the IUT responds with an HCI Command Complete event with the Key_Size parameter equal to the negotiated Key_Size in Step 5.
9. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT, and the IUT replies with an LMP_NAME_RES PDU, verifying that the encryption uses the negotiated Key_Size.
10. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT correctly reports the negotiated encryption Key_Size for each accepted Key_Size value, if HCI Read Encryption Key_Size Command is supported.

  At least one Key_Size value is accepted by the IUT.

  If the HCI_Set_Min_Encryption_Key_Size command is supported, then each accepted Key_Size value in Step 7A.1 >= the Min_Encryption_Key_Size from Step 1.

  Step 5d is executed for each KS < 7.

LMP/ENC/BV-81-C [Verify Key Size Mask]

- Test Purpose

  Verify that the IUT in the Peripheral role correctly reports the supported Key Size.

- Reference

  [1] 4.2.5

- Initial Condition

  - The Lower Tester has a minimum encryption Key_Size set to 1.
  - See Initial Conditions in Table 4.6-9.

- Test Procedure

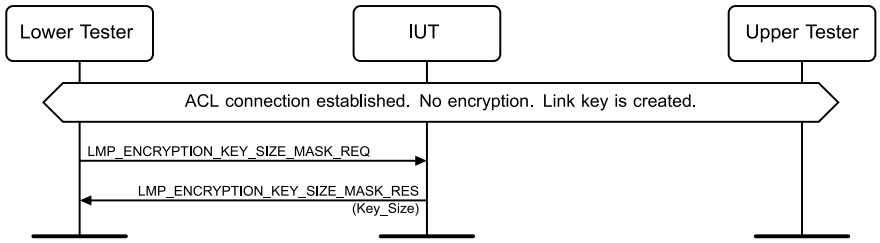


Figure 4.6-23: LMP/ENC/BV-81-C [Verify Key Size Mask] MSC

1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU to the IUT.
2. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_MASK_RES PDU with Key_Size with at least one bit 7–15 set and no bit 0–6 set.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ENCRYPTION_KEY_SIZE_MASK_RES PDU with a Key Size that contains at least one bit 7–15 set and bits 0–6 are not set.

  Fail verdict

  In Step 2, at least one bit 0–6 is set.

4.6.3 Encryption – Central

Verify that the Central and the Peripheral agree upon whether to use encryption or not and if encryption only applies to point-to-point packets or if encryption applies to both point-to-point packets and broadcast packets. The IUT is Central.

4.6.3.1 Initiate Encryption

- Test Purpose

  Verify that the IUT initiates the encryption procedure and uses encryption on point-to-point messages only.

- Reference

  [1] 4.2.5

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT and the Lower Tester indicate support for Secure Connections in their LMP Features based on Table 4.6-10.

- Test Case Configuration

Test Case	Secure Connections	Encryption
LMP/ENC/BV-05-C [Initiate Encryption]	Neither support	E0
LMP/ENC/BV-34-C [Initiate AES-CCM Encryption]	IUT and Lower Tester – Controller and Host	AES-CCM
LMP/ENC/BV-50-C [Initiate AES-CCM Encryption – Legacy Host]	Lower Tester only – Controller and Host	E0

Table 4.6-10: Initiate Encryption test cases

- Test Procedure

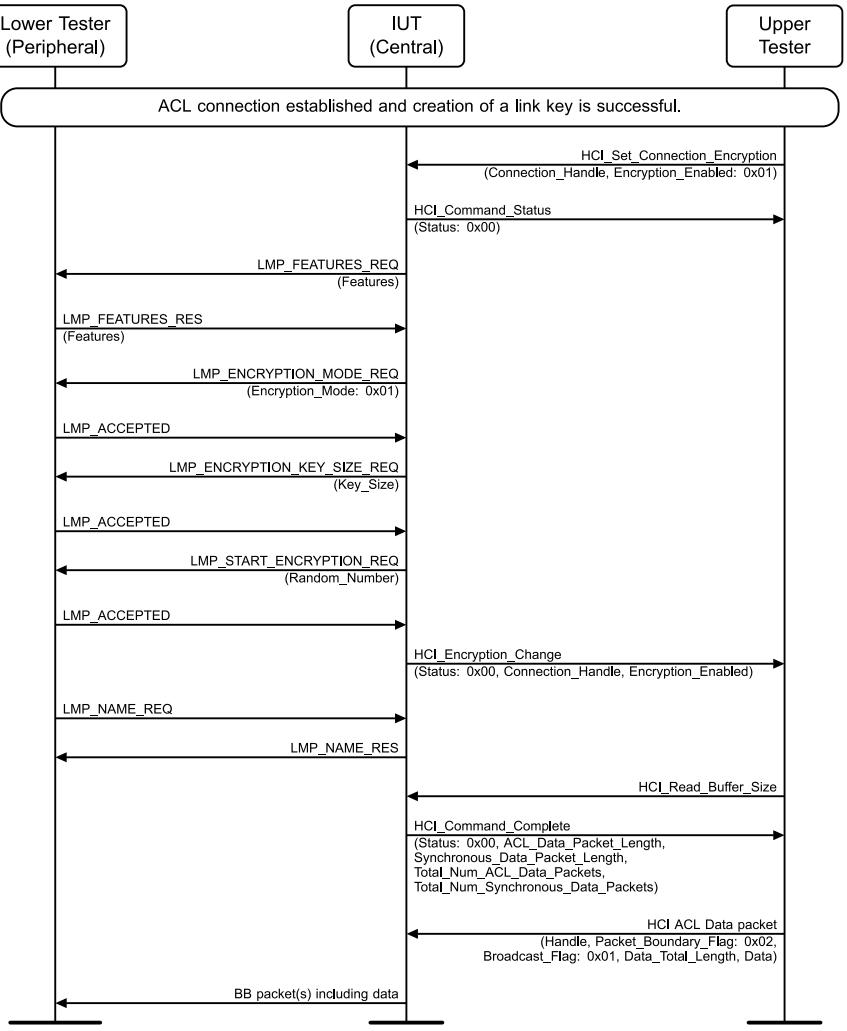


Figure 4.6-24: Initiate Encryption MSC

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
3. The Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with the Features parameter.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
6. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with a Key_Size that is within the Lower Tester's Key_Size range.
7. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
8. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

9. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
10. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enable set to 0x01, if using E0 encryption, or 0x02, if using AES-CCM encryption, as indicated in Table 4.6-10.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.
13. The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event with ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, and Total_Num_Synchronous_Data_Packets.
14. The Upper Tester sends HCI ACL Data packets to the IUT.
15. The IUT sends BB data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_FEATURES_REQ PDU at least once before starting the encryption procedure.

  The IUT initiates the encryption negotiation and uses the encryption afterwards.

  In Step 12, the IUT sends the LMP_NAME_RES PDU to the Lower Tester and proves that encryption is used.

  In Step 15, the IUT sends broadcast ACL packets to the Lower Tester with non-encrypted payload.

LMP/ENC/BV-82-C [Initiate Encryption]

- Test Purpose

  Verify that the IUT initiates the encryption procedure and uses encryption on point-to-point messages only. The IUT is Central and the Lower Tester is Peripheral.

- Reference

  [1] 4.2.5

- Initial Condition

  - See the “Default settings” section.

- Test Procedure

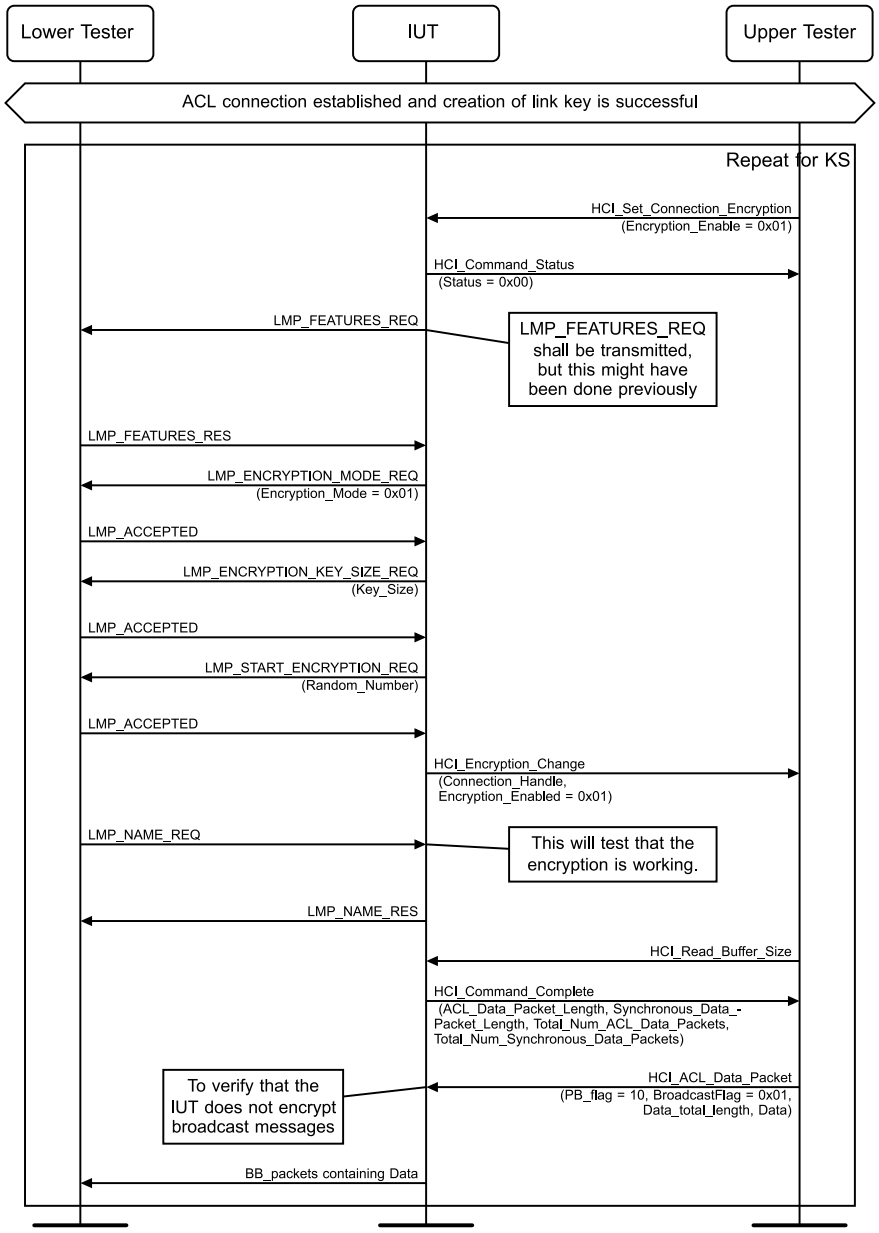


Figure 4.6-25: LMP/ENC/BV-82-C [Initiate Encryption] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester.
3. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
5. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
6. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with a Key_Size set to KS. If the IUT sends the LMP PDU when KS < 7, the test fails.
7. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with Random_Number.

8. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with Random_Number.
9. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
10. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x01.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
13. The Upper Tester sends an HCI ACL Data Packet to the IUT with PB_Flag set to 10, Broadcast_Flag set to 0x01, Data_total_length, and Data.
14. The IUT sends BB packet containing data to the Lower Tester.
15. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The PDU LMP_FEATURES_REQ has to be sent at least once by the IUT before starting the encryption.

  The IUT initiates the encryption negotiation and uses the encryption afterwards.

  The IUT responds correctly to the PDU LMP_NAME_REQ to prove that encryption is used.

  The IUT transmits broadcast ACL packets with non-encrypted payload to the Lower Tester.

  Fail verdict

  In Step 6, the IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU when KS < 7.

- Notes

  The suggested Key_Size must be within the Lower Tester's Key_Size range.

LMP/ENC/BV-85-C [Initiate AES-CCM Encryption]

- Test Purpose

  Verify that the IUT initiates the encryption procedure and uses AES-CCM encryption only for point-to-point messages when the remote controller's LMP feature bits indicate support for Secure Connections both in the Controller and the Host.

- Reference

  [1] 4.2.5

- Initial Condition

  - The Lower Tester is Peripheral and the IUT is Central.
  - An ACL connection has been established between the IUT and the Lower Tester, and the creation of a link key between the IUT and the Lower Tester has been successful.

- Test Procedure

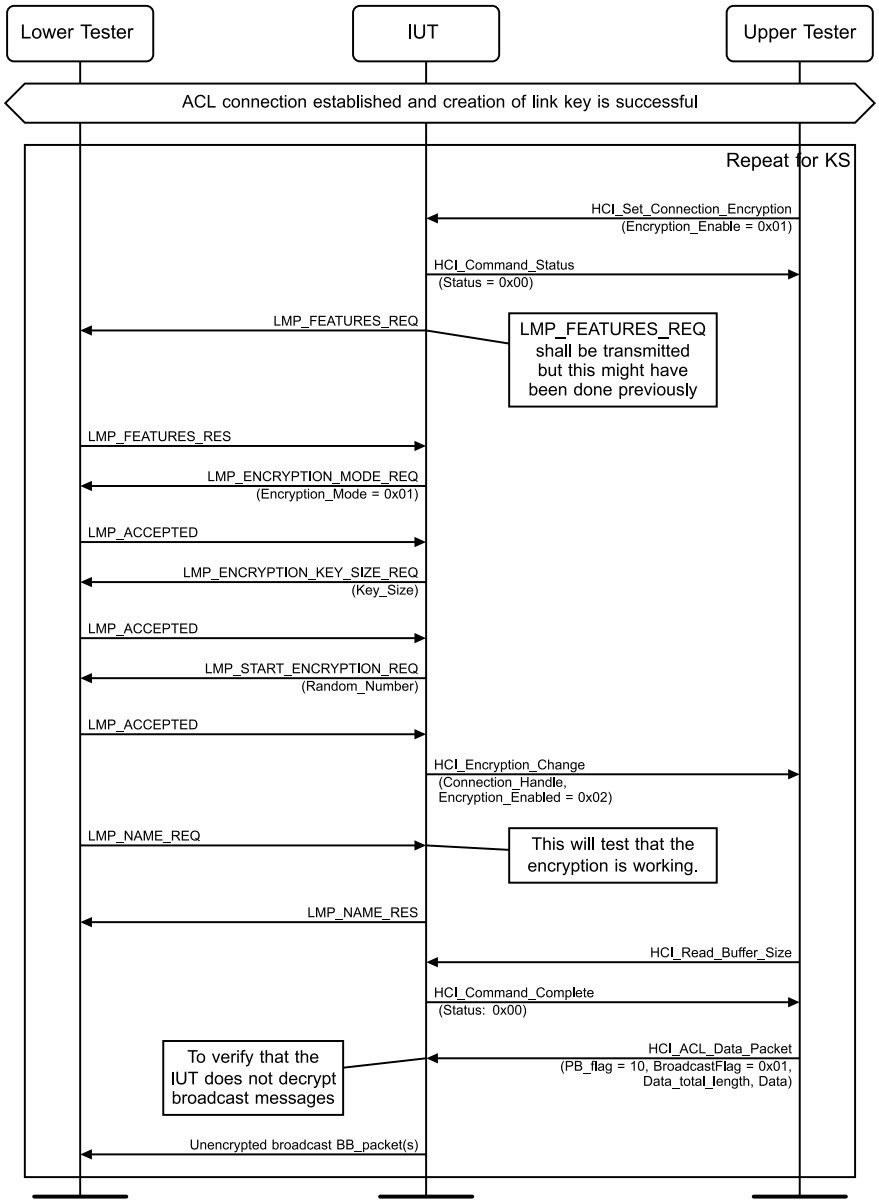


Figure 4.6-26: LMP/ENC/BV-85-C [Initiate AES-CCM Encryption] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with feature bits set for Secure Connections for both the Controller and the Host.
3. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT with feature bits set for Secure Connections both in the Controller and the Host.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
5. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
6. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with Key_Size set to KS. If the IUT sends the LMP PDU when KS < 7, the test fails.

7. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
8. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
9. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
10. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x02.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
13. The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event in response.
14. The Upper Tester sends an HCI_ACL_Data_Packet to the Upper Tester with PB_Flag set to 10, BroadcastFlag set to 0x01, Data_total_length, and Data.
15. The Lower Tester starts sending Unencrypted broadcast BB packets to the IUT.
16. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT initiates the encryption negotiation and uses the encryption afterwards.

  The IUT responds correctly to the PDU LMP_NAME_REQ (this proves that encryption is used).

  The IUT transmits broadcast ACL packets with non-encrypted payload to the Lower Tester.

  If Read Encryption Key_Size is supported, the IUT returns the Key_Size parameter from the LMP_ENCRYPTION_KEY_SIZE_REQ PDU in the Command Complete event following the HCI Read Encryption Key_Size command.

  The Encryption_Enabled Parameter in the Encryption Change event reports AES-CCM encryption has been enabled.

  Fail verdict

  In Step 6, the IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU when KS < 7.

- Notes

  The suggested Key_Size must be within the Lower Tester's Key_Size range.

  If the IUT starts to negotiate for encryption Key_Size the Lower Tester must negotiate.

  This test case is similar to LMP/ENC/BV-82-C [Initiate Encryption].

LMP/ENC/BV-86-C [Initiate AES-CCM Encryption – Legacy Host]

- Test Purpose

  Verify that the IUT initiates the encryption procedure and uses E0 encryption only for point-to-point messages when the remote controller's LMP feature bits indicate support for Secure Connections both in the Controller and the Host but the local Host does not indicate support for Secure Connections and reports the correct Encryption_Enabled to a legacy Host.

- Reference

  [1] 4.2.5

- Initial Condition
  - The Lower Tester is Peripheral and the IUT is Central.
  - The Upper Tester does not set the Secure Connections Host Support to enabled.
  - ACL connection has been established between the IUT and the Lower Tester and the creation of a link key between the IUT and Lower Tester has been successful.
- Test Procedure

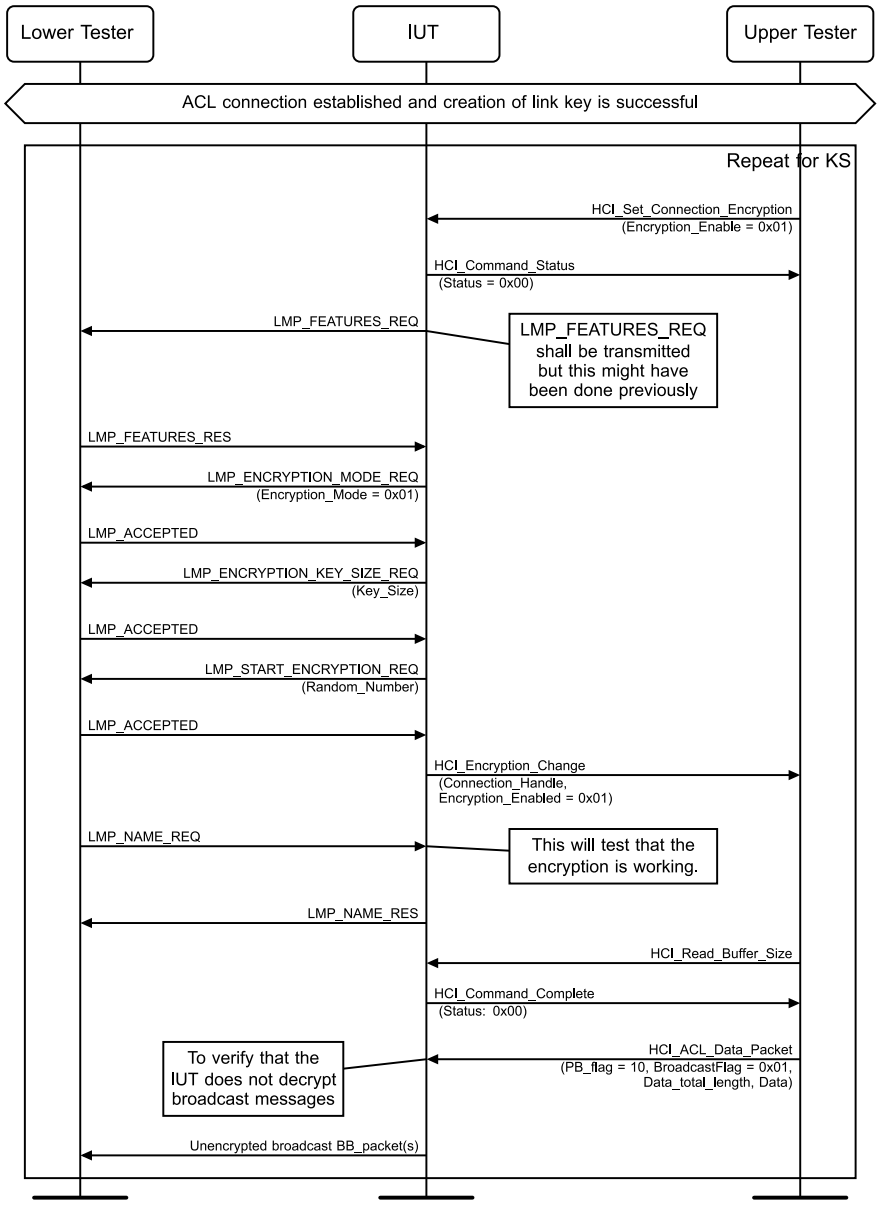


Figure 4.6-27: LMP/ENC/BV-86-C [Initiate AES-CCM Encryption – Legacy Host] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with feature bits set for Secure Connections for both the Controller and the Host.

3. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT with feature bits set for Secure Connections both in the Controller and the Host.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
5. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
6. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with Key_Size set to KS. If the IUT sends the LMP PDU when KS < 7, the test fails.
7. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
8. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
9. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
10. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x01.
11. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
12. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
13. The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event in response.
14. The Upper Tester sends an HCI_ACL_Data_Packet to the Upper Tester with PB_Flag set to 10, BroadcastFlag set to 0x01, Data_total_length, and Data.
15. The Lower Tester starts sending Unencrypted broadcast BB packets to the IUT.
16. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT initiates the encryption negotiation and uses the encryption afterwards.

  The IUT responds correctly to the PDU LMP_NAME_REQ (this proves that encryption is used).

  The IUT transmits broadcast ACL packets with non-encrypted payload to the Lower Tester.

  If Read Encryption Key_Size is supported, the IUT returns the Key_Size parameter from the LMP_ENCRYPTION_KEY_SIZE_REQ PDU in the Command Complete event following the HCI Read Encryption Key_Size command.

  The Encryption_Enabled Parameter in the Encryption Change event or the Connection Complete event reports “Link Level Encryption is ON with E0”.

  Fail verdict

  In Step 6, the IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU when KS < 7.

- Notes

  The suggested Key_Size must be within the Lower Tester’s Key_Size range.

LMP/ENC/BV-06-C [Peripheral Declines Encryption]

- Test Purpose

  Verify that the IUT accepts that the Lower Tester declines the Encryption_Mode.

- Reference

  [1] 4.2.5

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established between the IUT and the Lower Tester, and the creation of a link key between the IUT and the Lower Tester has been successful.
  - The Lower Tester and the IUT have both sent LMP_SETUP_COMPLETE PDUs.
  - The IUT is the Central and the Lower Tester is the Peripheral.
- Test Procedure

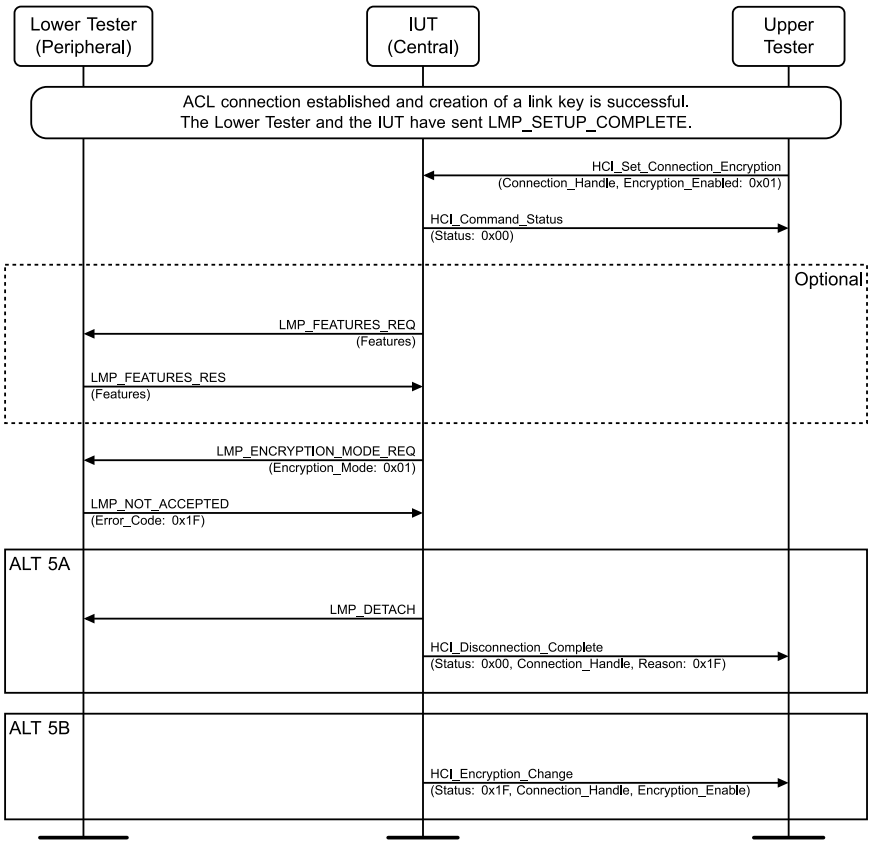


Figure 4.6-28: LMP/ENC/BV-06-C [Peripheral Declines Encryption] MSC

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status event in response.
2. If the IUT and the Lower Tester have not exchanged LMP Features, the IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter and receives an LMP_FEATURES_RES PDU with the Features parameter in response.
3. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
4. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode and Error_Code set to 0x1F (Unspecified Error).
5. Perform either alternative 5A or 5B depending on the IUT’s response.
   Alternative 5A (The IUT sends the LMP_DETACH PDU):
   - 5A.1. The IUT sends an LMP_DETACH PDU to the Lower Tester.
   - 5A.2. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x1F (Unspecified Error).

Alternative 5B (The IUT does not send the LMP_DETACH PDU):

5B.1. The IUT sends an HCI_Encryption_Change event to the Upper Tester with the Status set to 0x1F (Unspecified Error), Connection_Handle, and Encryption_Enable.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT sends the LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester.

  After receiving the LMP_NOT_ACCEPTED PDU from the Lower Tester in Step 4, the IUT does not continue with the encryption negotiation.

4.6.3.2 Encryption Stop

- Test Purpose

  Verify that the IUT initiates stop of encryption or accepts encryption stop upon request from the Lower Tester.

- Reference

  [1] 4.2.5.4

- Initial Condition

  - See Section 4.2.3, Encryption.
  - The IUT is the Central and the Lower Tester is the Peripheral.

- Test Case Configuration

Test Case	Stop Requester
LMP/ENC/BV-07-C [Initiate Encryption Stop]	Host
LMP/ENC/BV-08-C [Stop Encryption, Peripheral Request]	Peripheral

Table 4.6-11: Encryption Stop test cases

- Test Procedure

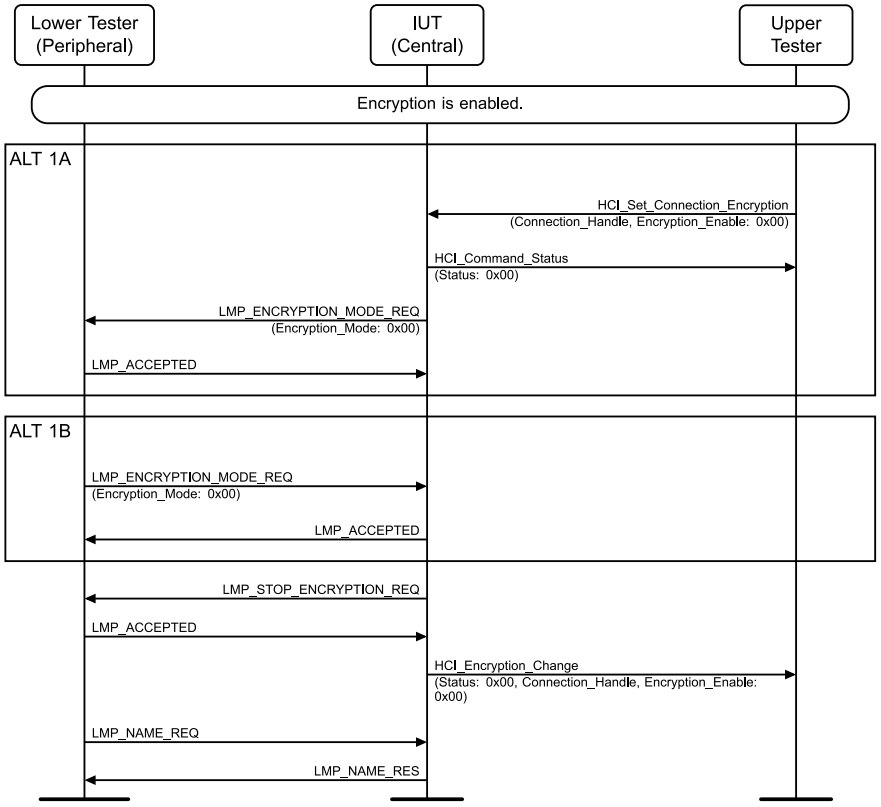


Figure 4.6-29: Initiate Encryption Stop MSC

1. Perform either alternative 1A or 1B depending on who is the stop requester as indicated in Table 4.6-11.
   Alternative 1A (The Host is the stop requester):
   - 1A.1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x00 and receives a successful HCI_Command_Status event in return.
   - 1A.2. The IUT sends an LMP_ENCRYPTION_MODE_REQ to the Lower Tester with Encryption_Mode set to 0x00 (No Encryption).
   - 1A.3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.

   Alternative 1B (The Peripheral is the stop requester):
   - 1B.1. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ to the IUT with Encryption_Mode set to 0x00 (No Encryption).
   - 1B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
2. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
4. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enable set to 0x00.
5. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
6. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester and stops using encryption.

  In Step 6, the IUT sends the LMP_NAME_RES PDU to the Lower Tester and proves that encryption is not being used.

LMP/ENC/BV-10-C [Initiate Broadcast Encryption]

- Test Purpose

  Verify that the IUT initiates the broadcast encryption negotiation procedure and uses encryption on point-to-point and broadcast messages.

- Reference

  [1] 4.2.5

- Initial Condition
  - An ACL connection is established.
  - No encryption is being used.
  - The Lower Tester is the Peripheral and the IUT is the Central.
  - HCI Buffers have been read.
  - The Lower Tester does not initiate the exchange of supported Features.
  - Broadcast and Unicast use different HCI connection handles.

- Test Procedure

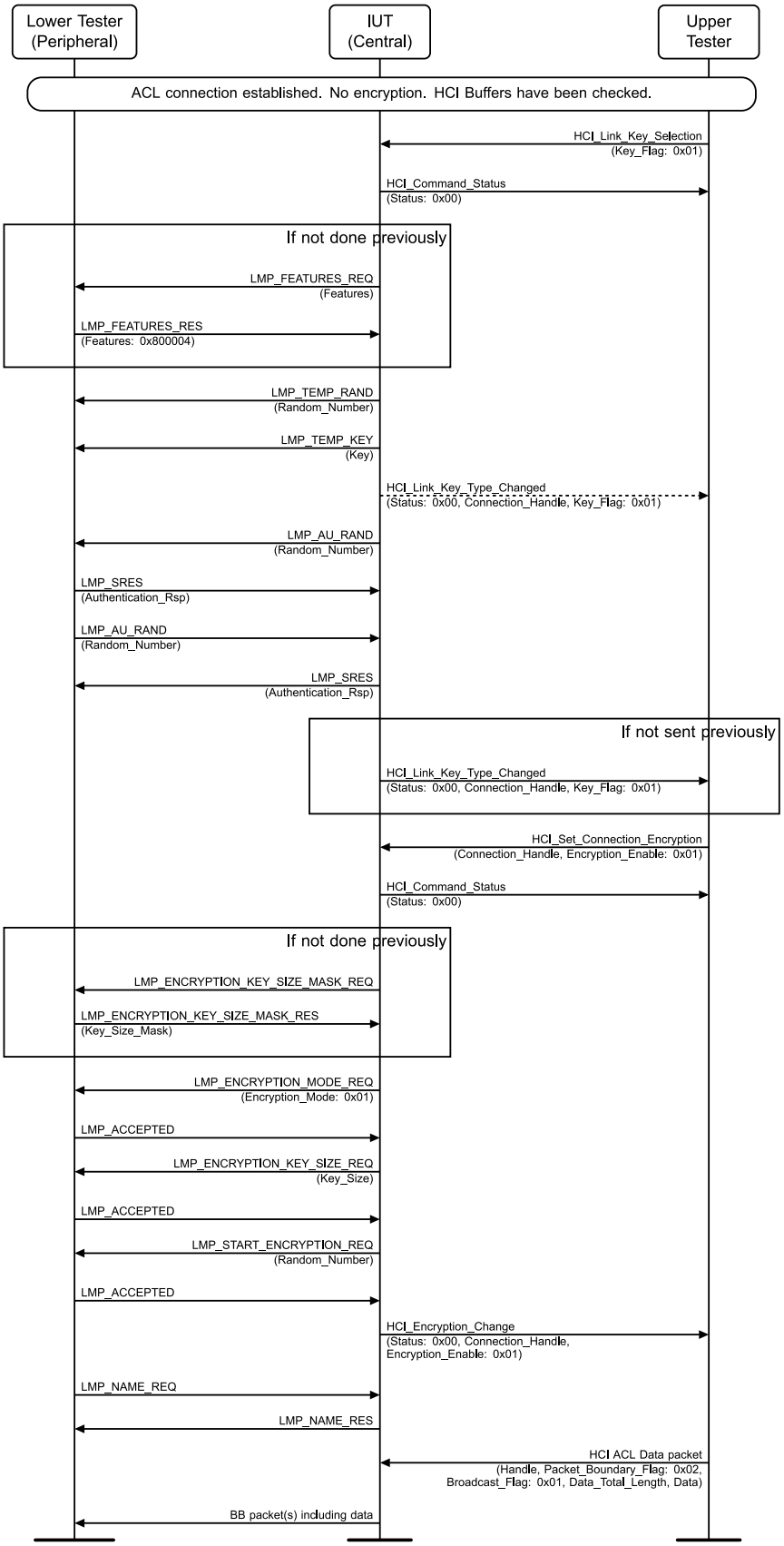


Figure 4.6-30: LMP/ENC/BV-10-C [Initiate Broadcast Encryption] MSC

1. The Upper Tester sends an HCI_Link_Key_Selection command to the IUT with Key_Flag set to 0x01 (Use Temporary Link Key) and receives a successful HCI_Command_Status event in response.
2. If LMP Features have not been exchanged, the IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter and receives an LMP_FEATURES_RES PDU in response with Features set to 0x800004 indicating Encryption and Broadcast Encryption support.
3. The IUT sends an LMP_TEMP_RAND PDU with a Random_Number and an LMP_TEMP_KEY PDU with a Key to the Lower Tester.
4. Optionally, the IUT sends a successful HCI_Link_Key_Type_Changed event to the Upper Tester with the Connection_Handle and Key_Flag set to 0x01 (Using Temporary Link Key).
5. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
6. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.
7. The Lower Tester sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
8. The IUT responds to the Lower Tester with an LMP_SRES PDU with the Authentication_Rsp.
9. If not sent in Step 4, the IUT sends a successful HCI_Link_Key_Type_Changed event to the Upper Tester with the Connection_Handle and Key_Flag set to 0x01 (Using Temporary Link Key).
10. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status event in return.
11. If the Lower Tester's supported encryption key sizes have not already been requested, the IUT sends an LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU to the Lower Tester and receives an LMP_ENCRYPTION_KEY_SIZE_MASK_RES with a Key_Size_Mask from the Lower Tester.
12. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
14. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
15. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
16. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
17. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
18. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enable set to 0x01.
19. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
20. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU.
21. The Upper Tester sends HCI ACL Data packets to the IUT.
22. The IUT sends BB data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT initiates the broadcast encryption negotiation and uses point-to-point and broadcast encryption afterwards.

  The LMP_FEATURES_REQ PDU is sent at least once by the IUT before starting the encryption.

LMP/ENC/BV-83-C [Initiate Broadcast Encryption]

- Test Purpose

  Verify that the IUT initiates the broadcast encryption negotiation procedure and uses encryption on point-to-point and broadcast messages.

- Reference

  [1] 4.2.5

- Initial Condition

  - See Figure 4.6-31.

- Test Procedure

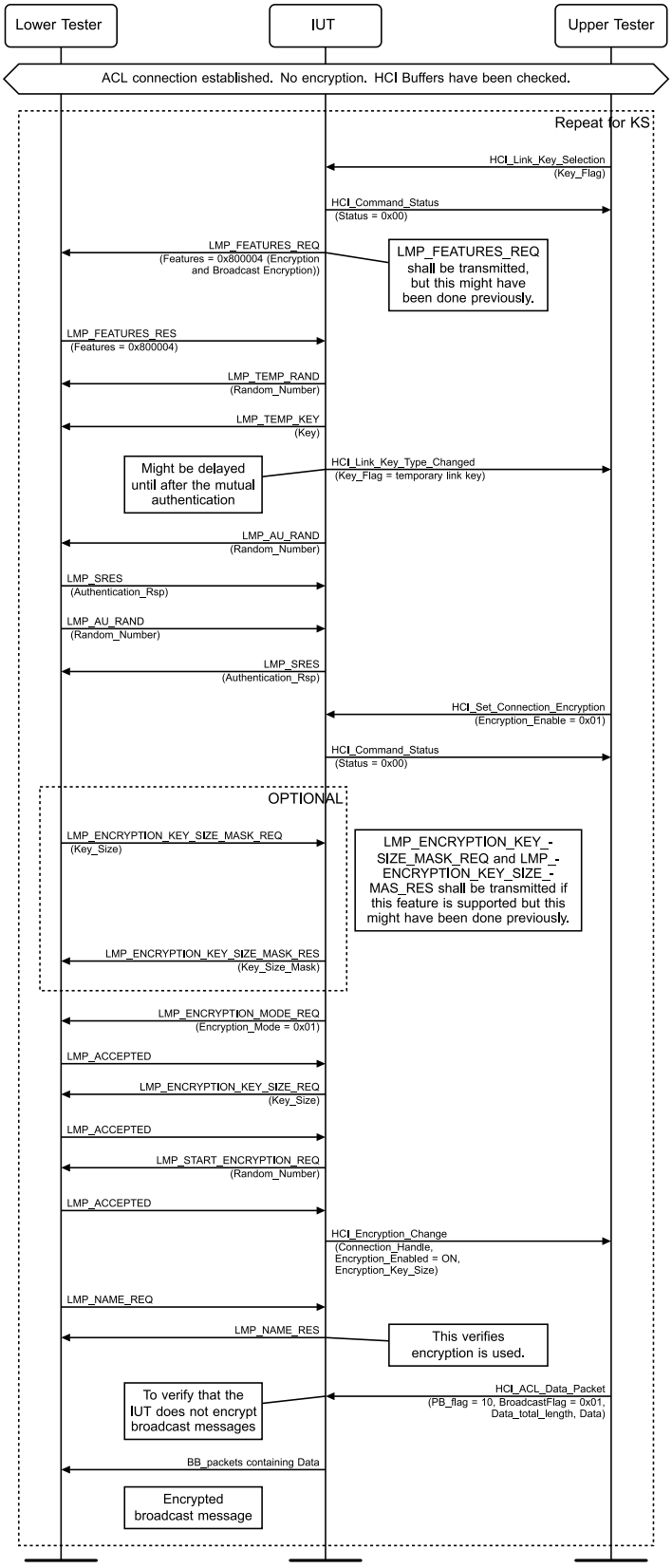


Figure 4.6-31: LMP/ENC/BV-83-C [Initiate Broadcast Encryption] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Upper Tester sends an HCI_Link_Key_Selection command to the IUT with Key_Flag and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester.
3. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT with Features set to 0x800004 (Encryption and Broadcast Encryption).
4. The IUT sends an LMP_TEMP_RAND PDU to the Lower Tester with a Random_Number.
5. The IUT sends an LMP_TEMP_KEY PDU to the Lower Tester with a Key.
6. The IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with Key_Flag set to the temp link key.
7. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
8. The Lower Tester sends an LMP_SRES PDU to the IUT with Authentication_Rsp.
9. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
10. The IUT sends an LMP_SRES PDU to the Lower Tester with Authentication_Rsp.
11. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT and receives a successful HCI_Command_Status in response.
12. If the IUT supports the Encryption Key Size Mask, execute Step 12.
    12A. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_MASK_REQ PDU to the IUT with a Key_Size set to KS.
    12B. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_MASK_RES PDU to the Lower Tester with Key_Size_Mask.
13. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
14. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
15. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with Key_Size set to KS. If the IUT sends the LMP PDU when KS < 7, the test fails.
16. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
17. The IUT sends an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled ON and an Encryption_Key_Size.
18. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester.
19. The Lower Tester sends an LMP_NAME_RES PDU to the IUT.
20. The Upper Tester sends an HCI ACL Data Packet to the IUT with PB_Flag set to 10, Broadcast_Flag set to 0x01, Data_total_length, and Data.
21. The IUT sends BB packet containing data to the Lower Tester.
22. The Lower Tester disconnects the ACL link.

- Test Condition

  It must be possible to control the IUT to initiate encryption.

- Expected Outcome

  Pass verdict

  The IUT initiates the broadcast encryption negotiation and uses point-to-point and broadcast encryption afterwards. The PDU LMP_FEATURES_REQ has to be sent at least once by the IUT before starting the encryption.

  Fail verdict

  In Step 15, the IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU when KS < 7.

- Notes

  The suggested Key_Size must be within the Lower Tester's Key_Size range. The Lower Tester does not initiate exchange of supported Features. Broadcast and Unicast use different HCI connection handles.

LMP/ENC/BV-13-C [Initiate Semi-permanent Link Key Change]

- Test Purpose

  Verify that the IUT can initiate a change to the semi-permanent link key and that the IUT stops the encryption.

- Reference

  [1] 4.2.4.2

- Initial Condition

  - Broadcast encryption is used.
  - The IUT is the Central and the Lower Tester is the Peripheral.

- Test Procedure

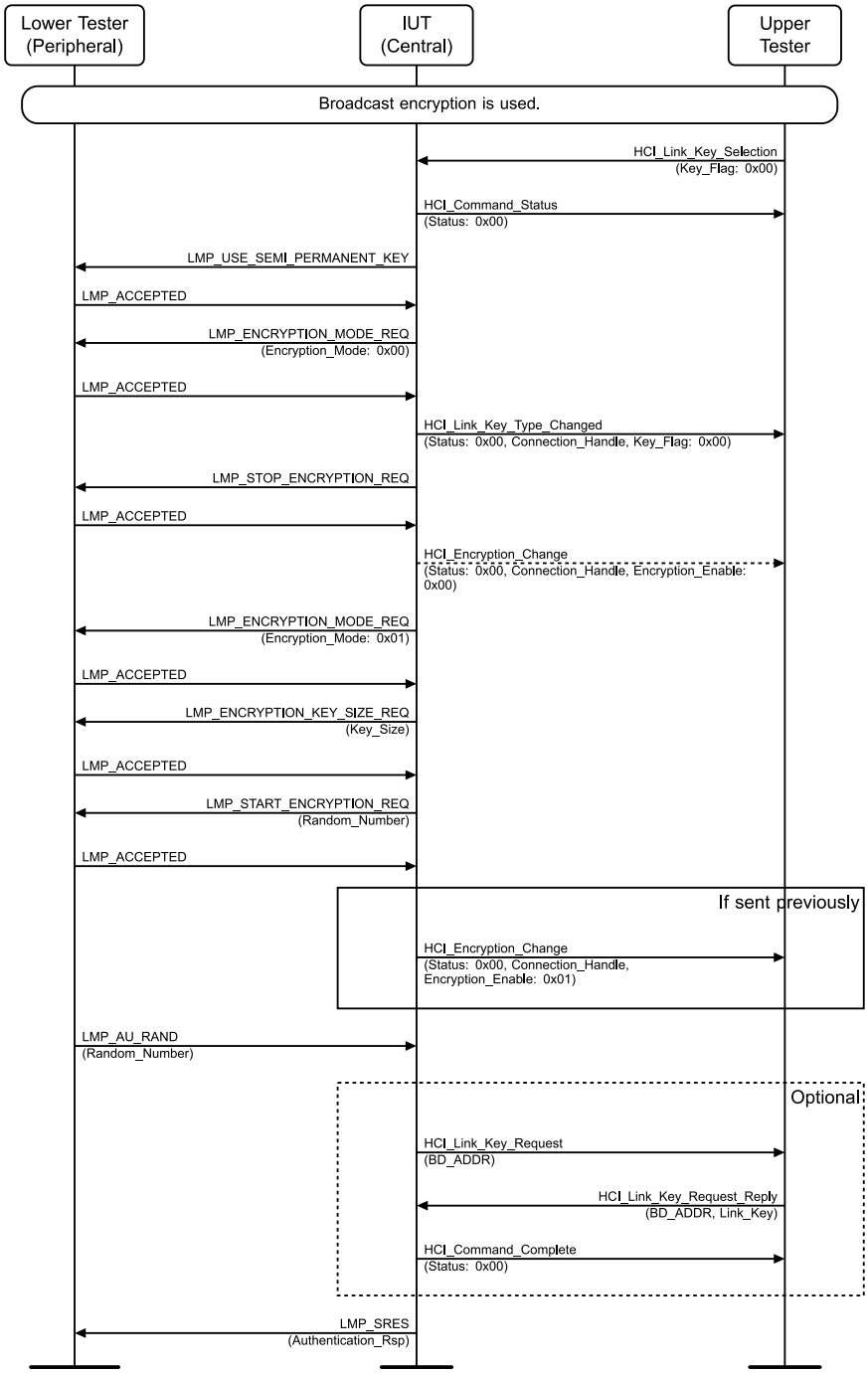


Figure 4.6-32: LMP/ENC/BV-13-C [Initiate Semi-permanent Link Key Change] MSC

1. The Upper Tester sends an HCI_Link_Key_Selection command to the IUT with Key_Flag set to 0x00 (Use semi-permanent Link Keys) and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_USE_SEMI_PERMANENT_KEY PDU to the Lower Tester.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_USE_SEMI_PERMANENT_KEY PDU Opcode.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x00.

5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
6. The IUT sends a successful HCI_Link_Key_Type_Changed event with the Connection_Handle and Key_Flag set to 0x00 (Use semi-permanent Link Keys).
7. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
8. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
9. Optionally, the IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_enable set to 0x00.
10. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
12. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with a Key_Size.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
14. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
15. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
16. If Step 9 occurred, the IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_enable set to 0x01.
17. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
18. Optionally, the IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
19. If Step 18 occurs, the Upper Tester sends an HCI_Link_Key_Request_Reply command to the IUT with the BD_ADDR and Link_Key and receives a successful HCI_Command_Complete event in response.
20. The IUT sends an LMP_SRES PDU to the Lower Tester with the Authentication_Rsp.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_USE_SEMI_PERMANENT_KEY PDU, LMP_ENCRYPTION_MODE_REQ PDUs, and LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester. Encryption is restarted. The link key used is a semi-permanent key.

LMP/ENC/BV-84-C [Initiate Semi-permanent Link Key Change]

- Test Purpose

  Verify that the IUT can initiate a change to the semi-permanent link key. Verify that the IUT stops the encryption. The IUT is Central. The Lower Tester is Peripheral.

- Reference

  [1] 4.2.4.2

- Initial Condition

  - See Figure 4.6-33.

- Test Procedure

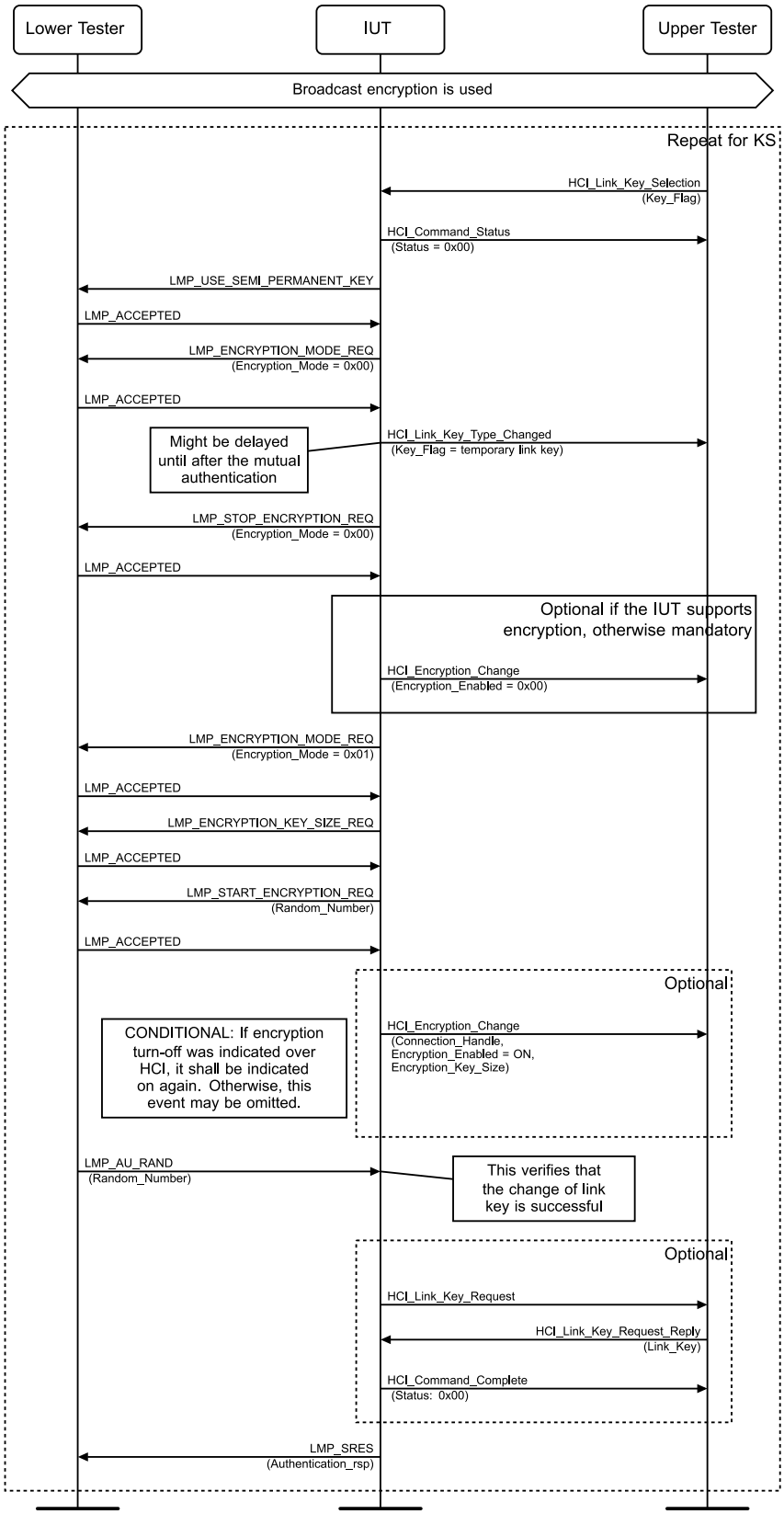


Figure 4.6-33: LMP/ENC/BV-84-C [Initiate Semi-permanent Link Key Change] MSC

Repeat the test procedure for encryption Key_Size value KS in the interval [16, 1].

1. The Upper Tester sends an HCI_Link_Key_Selection command to the IUT with Key_Flag and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_USE_SEMI_PERMANENT_KEY PDU to the Lower Tester.
3. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x00.
5. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
6. The IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with a Key_Flag.
7. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester with Encryption_Mode set to 0x00.
8. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
9. If the IUT supports encryption, the IUT may send an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled set to 0x00.
10. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
11. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester.
12. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with a Key_Size set to KS. If the IUT sends the LMP PDU when KS < 7, the test fails.
13. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
14. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
15. The IUT may send an HCI_Encryption_Change event to the Upper Tester with Encryption_Enabled ON and an Encryption_Key_Size.
16. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
17. The IUT may execute Step 17.
    - 17A. The IUT sends an HCI_Link_Key_Request event to the Upper Tester.
    - 17B. The Upper Tester sends an HCI_Link_Key_Request_Reply command to the IUT with Link_Key.
    - 17C. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
18. The IUT sends an LMP_SRES to the Lower Tester with Authentication_rsp.
19. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT transmits LMP_USE_SEMI_PERMANENT_KEY, LMP_ENCRYPTION_MODE_REQ and LMP_STOP_ENCRYPTION_REQ. Encryption is restarted. The link key must be the semi-permanent key.

  Fail verdict

  In Step 12, the IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU when KS < 7.

4.6.3.3 Pausing and Resuming Encryption without Disabling Encryption_Mode – Central Initiated

- Test Purpose

  Verify that the IUT as the Central can pause and resume encryption without disabling the Encryption_Mode.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition

  - The Lower Tester is the Peripheral and the IUT is the Central.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-12.
  - Both devices are sending data to each other.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-14-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Central Initiated]	E0
LMP/ENC/BV-37-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Central Initiated as a result of change connection link Key]	AES-CCM

Table 4.6-12: Pausing and Resuming Encryption without Disabling Encryption_Mode – Central Initiated test cases

- Test Procedure

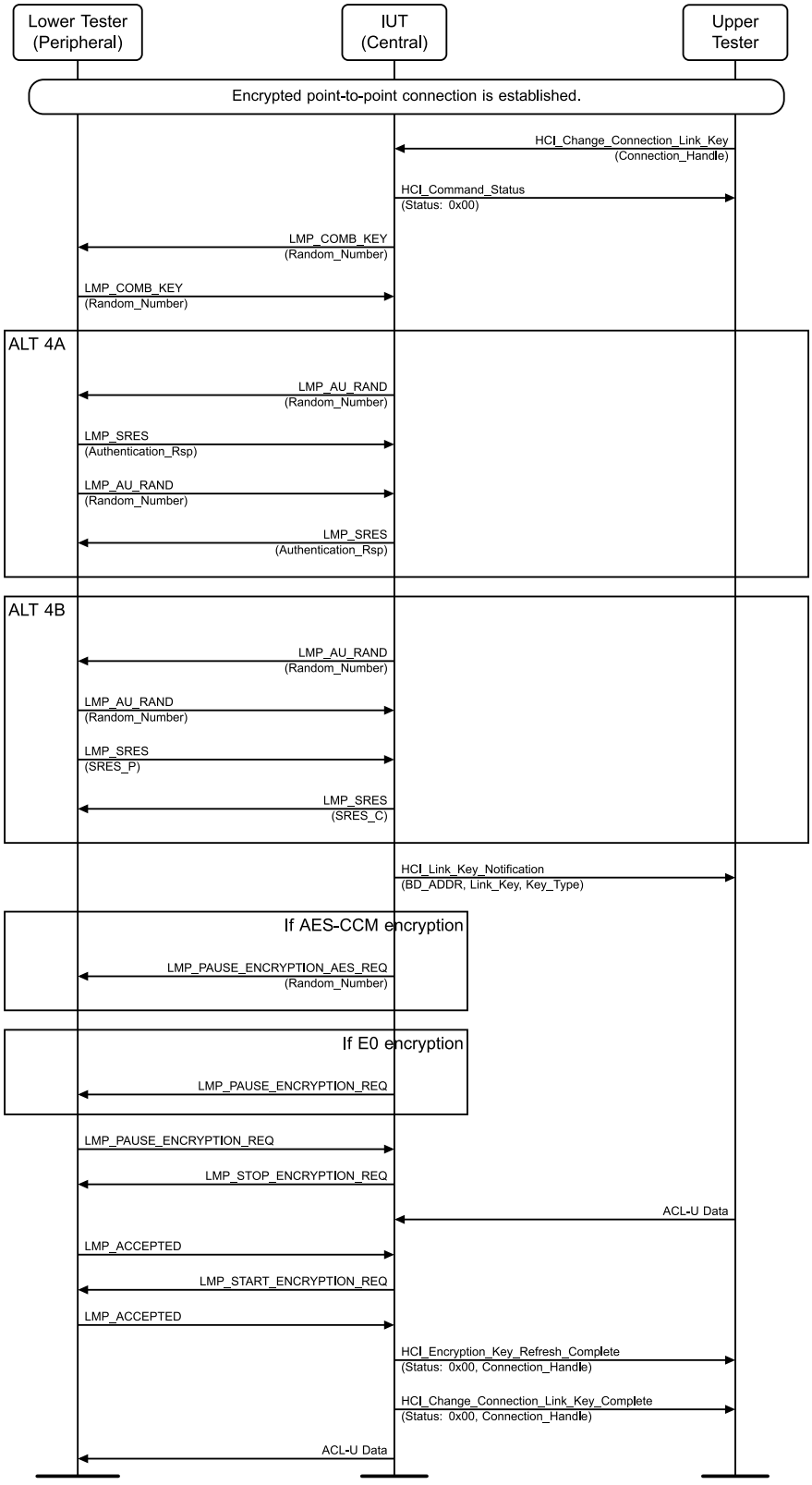


Figure 4.6-34: Pausing and Resuming Encryption without Disabling Encryption_Mode – Central Initiated MSC

1. The Upper Tester sends an HCI_Change_Connection_Link_Key command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
4. Perform either alternative 4A or 4B depending on the encryption indicated in Table 4.6-12.
   Alternative 4A (The connection uses E0 encryption):
   - 4A.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
   - 4A.2. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.
   - 4A.3. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
   - 4A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the Authentication_Rsp.

   Alternative 4B (The connection uses AES-CCM encryption):
   - 4B.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
   - 4B.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
   - 4B.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P.
   - 4B.4. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_C.
5. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with the BD_ADDR, Link_Key, and Key_Type.
6. If AES-CCM encryption is being used as indicated in Table 4.6-12, the IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number; otherwise, the IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
7. The Lower Tester responds to the IUT with an LMP_PAUSE_ENCRYPTION_REQ PDU.
8. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
9. The Upper Tester sends ACL-U Data to the IUT.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
11. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
13. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event with the Connection_Handle and has sent a successful HCI_Change_Connection_Link_Key_Complete event with the Connection_Handle to the Upper Tester.
14. The IUT continues sending ACL-U Data to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT pauses encryption to the Lower Tester using the LMP_PAUSE_ENCRYPTION_REQ PDU, if using E0 encryption, or the LMP_PAUSE_ENCRYPTION_AES_REQ PDU, if using AES-CCM encryption.

  The IUT sends no data packets to the Lower Tester while encryption is paused.

  The IUT resumes sending data packets to the Lower Tester after encryption is resumed.

- Notes

  The HCI_Change_Connection_Link_Key_Complete event may be sent any time after the HCI_Link_Key_Notification event.

4.6.3.4 Pausing and Resuming Encryption without Disabling Encryption_Mode – Central Initiated with Role Switch

- Test Procedure

  Verify that the IUT as the Central can pause and resume encryption without disabling the Encryption_Mode as part of the role switch procedure.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition

  - The Lower Tester is the Peripheral and the IUT is the Central.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-13.
  - Both devices are sending data to the other device.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-16-C [Pausing and Resuming Encryption without Disabling Encryption_Mode - Central Initiated with Role Switch]	E0
LMP/ENC/BV-41-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Central Initiated with role switch]	AES-CCM

Table 4.6-13: Pausing and Resuming Encryption without Disabling Encryption_Mode – Central Initiated with Role Switch test cases

- Test Procedure

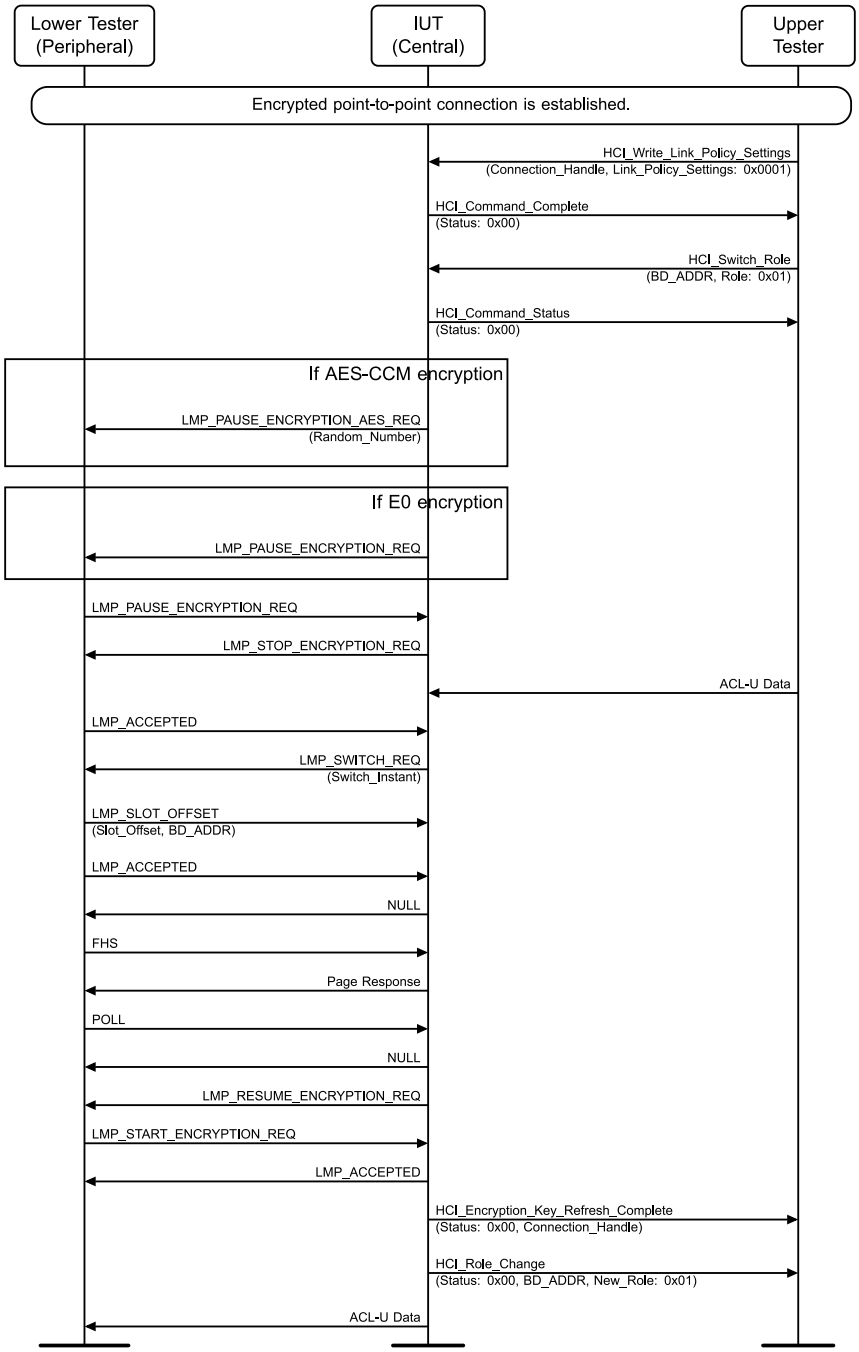


Figure 4.6-35: Pausing and Resuming Encryption without Disabling Encryption_Mode – Central Initiated with Role Switch MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0001 (Enable Role Switch) and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x01 (Peripheral) and receives a successful HCI_Command_Status event in response.
3. If AES-CCM encryption is being used as indicated in Table 4.6-13, the IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number; otherwise, the IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.

4. The Lower Tester responds to the IUT with an LMP_PAUSE_ENCRYPTION_REQ PDU.
5. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
6. The Upper Tester sends ACL-U Data to the IUT.
7. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
8. The IUT sends an LMP_SWITCH_REQ PDU to the Lower Tester with the Switch_Instant.
9. The Lower Tester responds to the IUT with an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
10. The IUT sends a NULL packet to the Lower Tester.
11. The Lower Tester sends an FHS packet to the IUT.
12. The IUT sends a Page Response packet to the Lower Tester.
13. The Lower Tester sends a POLL packet to the IUT.
14. The IUT sends a NULL packet to the Lower Tester.
15. The IUT sends an LMP_RESUME_ENCRYPTION_REQ PDU to the Lower Tester.
16. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
18. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle and a successful HCI_Role_Change event with the BD_ADDR and New_Role set to 0x01 (Peripheral).
19. The IUT continues sending ACL-U Data to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT pauses encryption to the Lower Tester using the LMP_PAUSE_ENCRYPTION_REQ PDU, if using E0 encryption, or the LMP_PAUSE_ENCRYPTION_AES_REQ PDU, if using AES-CCM encryption.

  The IUT sends no data packets while encryption is paused.

  The IUT sends data packets after encryption is resumed.

  The role switch succeeds.

- Notes

  If the test procedure fails due to the role switch, repeat the test.

  The HCI_Role_Change event may be received any time after the role switch.

LMP/ENC/BV-18-C [Starting and Stopping Encryption with Legacy Device - Central Initiated]

- Test Procedure

  Verify that the IUT as the Central can stop and restart encryption with a device that does not support Encryption Pause Resume.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition
  - The Lower Tester is the Peripheral and the IUT is the Central.
  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester.
  - The Lower Tester is a device that does not support Encryption Pause Resume.
- Test Procedure

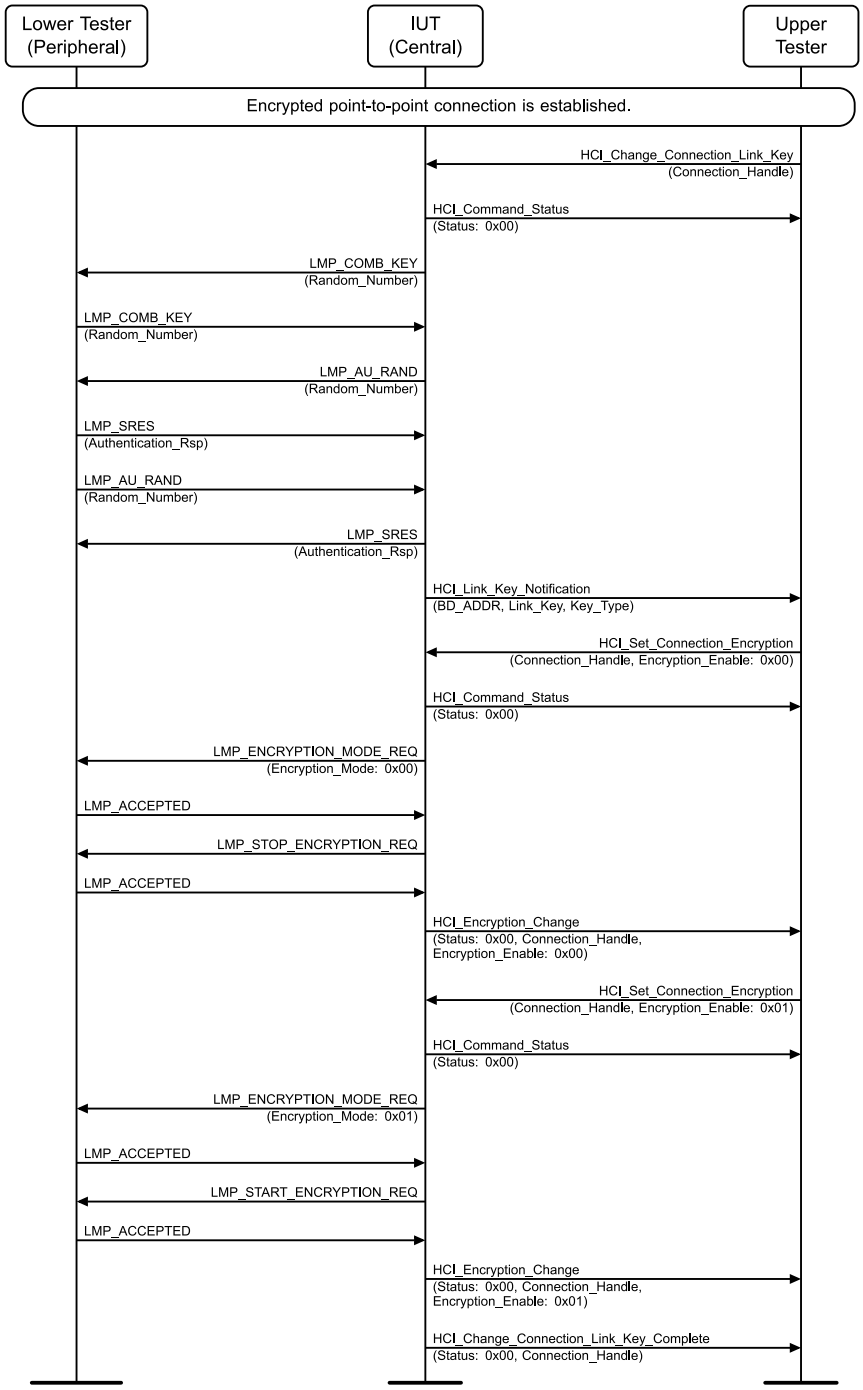


Figure 4.6-36: LMP/ENC/BV-18-C [Starting and Stopping Encryption with Legacy Device - Central Initiated] MSC

1. The Upper Tester sends an HCI_Change_Connection_Link_Key command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
3. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
4. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
5. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.
6. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
7. The IUT responds with an LMP_SRES PDU to the Lower Tester with the Authentication_Rsp.
8. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with the BD_ADDR, Link_Key, and Key_Type.
9. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with Encryption_Enable set to 0x00 and receives a successful HCI_Command_Status event in response.
10. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x00.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
12. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
14. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enable set to 0x00.
15. The Upper Tester sends an HCI_Set_Connection_Encryption PDU to the IUT with Encryption_Enable set to 0x01 and receives a successful HCI_Command_Status event in response.
16. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
17. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
18. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
20. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
21. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
22. The IUT sends a successful HCI_Encryption_Change event with the Connection_Handle and Encryption_Enable set to 0x01 and has sent a successful HCI_Change_Connection_Link_Key event with the Connection_Handle to the Upper Tester.

- Expected Outcome

  Pass verdict

  Encryption is stopped and then restarted.

- Notes

  The HCI_Change_Connection_Link_Key_Complete event may be received any time after the HCI_Link_Key_Notification event.

4.6.3.5 Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated as a Result of Change Connection Link Key

- Test Purpose

  Verify that the IUT as the Central can respond to a Peripheral-initiated pause and resume of encryption without disabling the Encryption_Mode as part of the change connection link key procedure.

- Reference

  [1] 4.2.3, 4.2.5.3, 4.2.5.5

- Initial Condition

  - The Lower Tester is the Peripheral and the IUT is the Central.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-14.
  - Both devices are sending data to each other.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-20-C [Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated as a Result of Change Connection Link Key]	E0
LMP/ENC/BV-39-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Peripheral Initiated as a result of change connection link Key]	AES-CCM

Table 4.6-14: Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated as a Result of Change Connection Link Key test cases

- Test Procedure

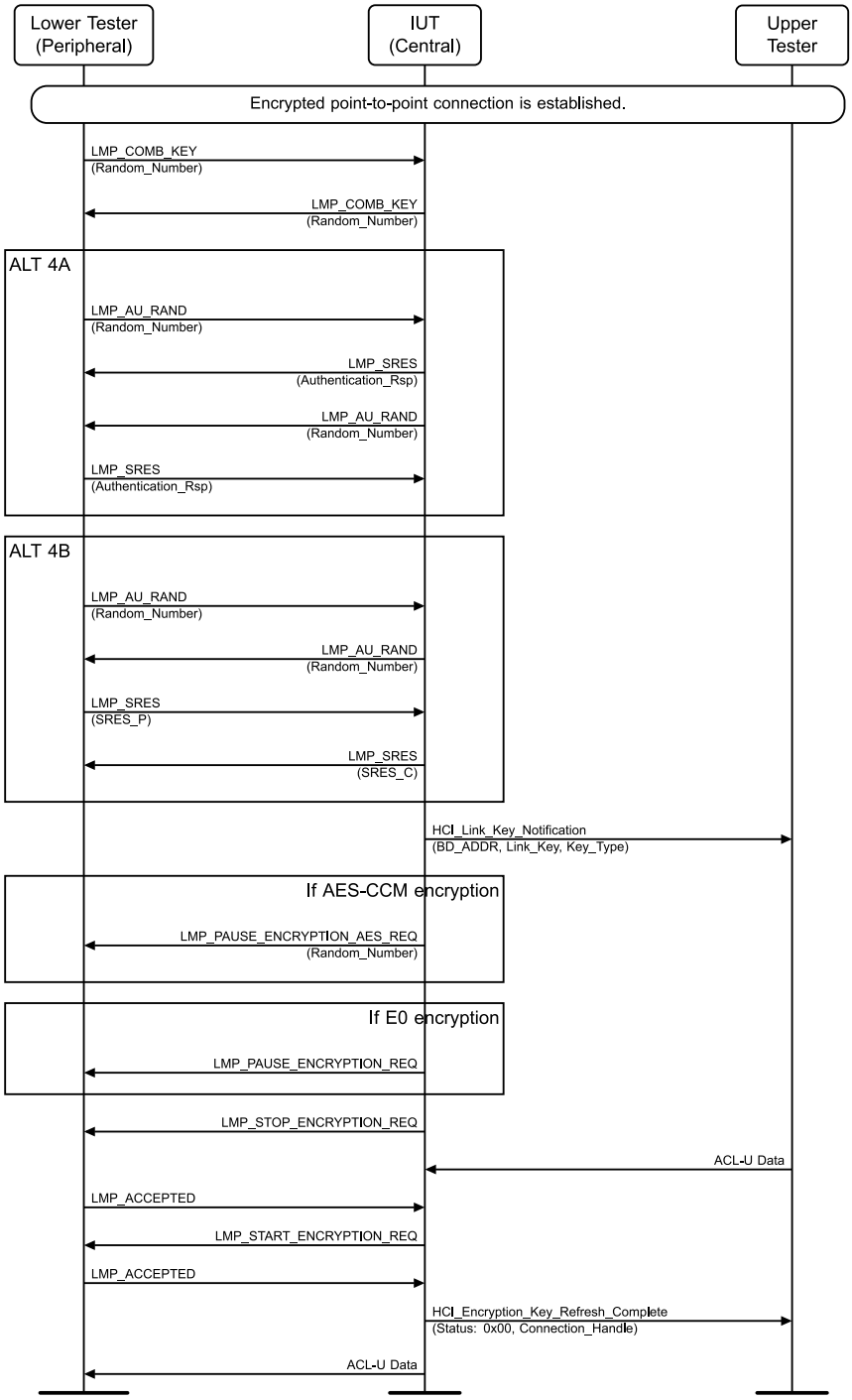


Figure 4.6-37: Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated as a Result of Change Connection Link Key MSC

1. The Lower Tester sends an LMP_COMB_KEY PDU to the IUT with a Random_Number.
2. The IUT responds to the Lower Tester with an LMP_COMB_KEY PDU with a Random_Number.
3. Perform either alternative 3A or 3B depending on the encryption indicated in Table 4.6-14.
   Alternative 3A (The connection uses E0 encryption):
   - 3A.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
   - 3A.2. The IUT responds to the Lower Tester with an LMP_SRES PDU with the Authentication_Rsp.
   - 3A.3. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
   - 3A.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp.

   Alternative 3B (The connection uses AES-CCM encryption):
   - 3B.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
   - 3B.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.
   - 3B.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES_P.
   - 3B.4. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES_C.
4. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with BD_ADDR, Link_Key, and Key_Type set to 0x00 (Combination Key), if E0 encryption is being used, or 0x06, if AES-CCM encryption is being used.
5. If AES-CCM encryption is being used as indicated in Table 4.6-14, the IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number; otherwise, the IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
6. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
7. The Upper Tester sends ACL-U Data to the IUT.
8. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
9. The Lower Tester sends an LMP_RESUME_ENCRYPTION_REQ PDU to the IUT.
10. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
12. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.
13. The IUT sends the ACL-U Data to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 6, the IUT pauses encryption by sending the LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.

  The IUT sends no data packets while encryption is paused.

  The IUT sends data packets after encryption is resumed.

4.6.3.6 Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated with Role Switch

- Test Purpose

  Verify that the IUT as the Central can respond to a Peripheral-initiated pause and resume of encryption without disabling the Encryption_Mode as part of the role switch procedure.

- Reference

  [1] 4.2.5.3, 4.2.5.5, 4.4.2

- Initial Condition

  - The Lower Tester is the Peripheral and the IUT is the Central.
  - A point-to-point connection has been established between the IUT and the Lower Tester using the encryption specified in Table 4.6-15.
  - Both devices are sending data to each other.

- Test Case Configuration

Test Case	Encryption
LMP/ENC/BV-21-C [Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated with Role Switch]	E0
LMP/ENC/BV-43-C [Pausing and Resuming without Disabling AES-CCM Encryption, IUT Central, Peripheral Initiated with role switch]	AES-CCM

Table 4.6-15: Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated with Role Switch test cases

- Test Procedure

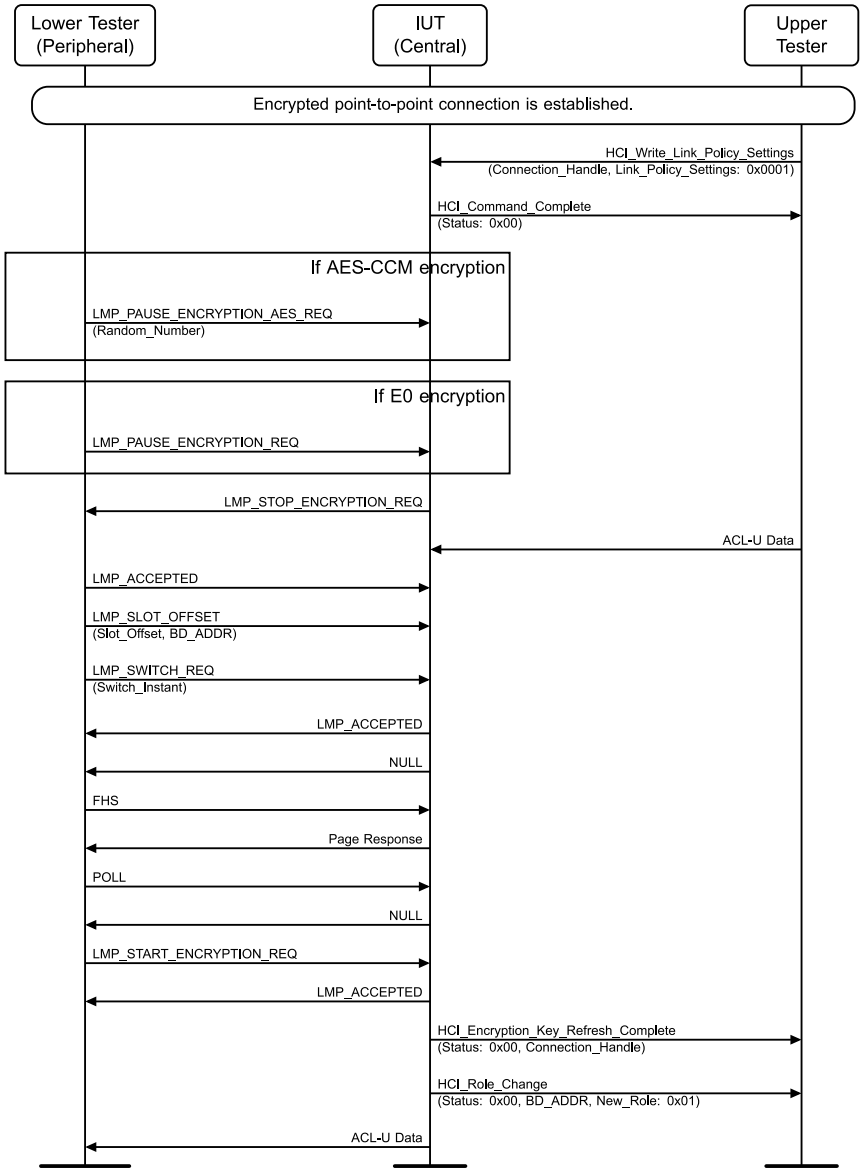


Figure 4.6-38: Pausing and Resuming without Disabling Encryption, IUT Central, Peripheral Initiated with Role Switch MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command with the Connection Handle and Link Policy Settings set to 0x00001 (Enable Role Switch) and receives a successful HCI_Command_Complete event.
2. If AES-CCM encryption is being used as indicated in Table 4.6-15, the Lower Tester sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the IUT with a Random_Number; otherwise, the Lower Tester sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the IUT.
3. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
4. The Upper Tester sends ACL-U Data to the IUT.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
6. The Lower Tester sends an LMP_SWITCH_REQ PDU with the Switch_Instant and an LMP_SLOT_OFFSET PDU with the Slot_Offset and the BD_ADDR to the IUT.

7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
8. The IUT sends a NULL packet to the Lower Tester.
9. The Lower Tester sends an FHS packet to the IUT.
10. The IUT sends a Page Response to the Lower Tester.
11. The Lower Tester sends a POLL packet to the IUT.
12. The IUT sends a NULL packet to the Lower Tester.
13. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT.
14. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_START_ENCRYPTION_REQ PDU Opcode.
15. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event with the Connection_Handle and a successful HCI_Role_Change event with the BD_ADDR and New_Role set to 0x01 (Peripheral) to the Upper Tester.
16. The IUT sends ACL-U Data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT pauses encryption by sending the LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.

  The IUT sends no data packets while encryption is paused.

  The IUT sends data packets after encryption is resumed.

  The role switch succeeds.

LMP/ENC/BV-45-C [Broadcast Encryption is not used with AES-CCM encryption]

- Test Purpose

  Verify that the IUT as the Central rejects a request from the Upper Tester to use the Temporary Link Key when AES-CCM encryption has been enabled on a point-to-point link with a Peripheral and does not use encryption on broadcast messages.

- Reference

  [1] 4.2.4.1

- Initial Condition

  - The Lower Tester is the Peripheral and the IUT is the Central.
  - An AES-CCM encrypted point-to-point connection has been established between the IUT and the Lower Tester.

- Test Procedure

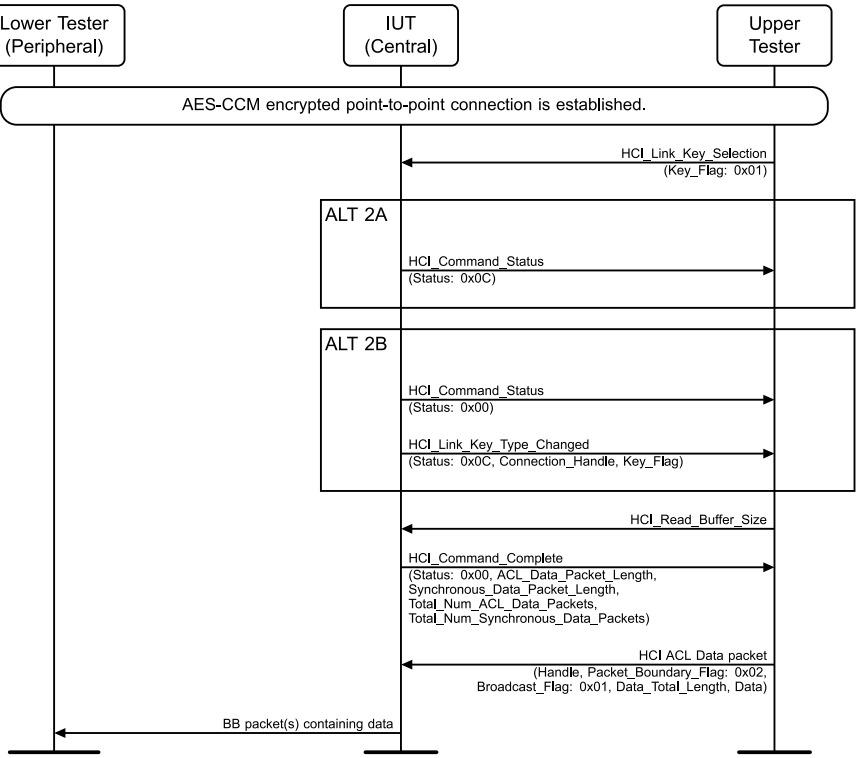


Figure 4.6-39: LMP/ENC/BV-45-C [Broadcast Encryption is not used with AES-CCM encryption] MSC

1. The Upper Tester sends an HCI_Link_Key_Selection command to the IUT with Key_Flag set to 0x01 (Use Temporary Link Key).
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT initially rejects the HCI_Link_Key_Selection command):
   - 2A.1. The IUT sends an HCI_Command_Status event to the Upper Tester with Status set to 0x0C (Command Disallowed).

   Alternative 2B (The IUT initially accepts the HCI_Link_Key_Selection command):
   - 2B.1. The IUT sends a successful HCI_Command_Status event to the Upper Tester.
   - 2B.2. The IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with the Status set to 0x0C (Command Disallowed), Connection_Handle, and Key_Flag.
3. The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event with ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, and Total_Num_Synchronous_Data_Packets.
4. The Upper Tester sends HCI ACL Data packets to the IUT.
5. The IUT sends BB data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  In alternative 2A, the IUT sends an HCI_Command_Status event to the Lower Tester with Error_Code set to 0x0C (Command Disallowed).

In alternative 2B, the IUT sends a successful HCI_Command_Status event followed by an HCI_Link_Key_Type_Changed event with Status set to 0x0C (Command Disallowed) to the Upper Tester.

In Step 5, broadcast messages are sent unencrypted.

4.6.3.7 Key_Size Negotiation as Central

- Test Purpose

  Verify that the IUT in the Central role correctly reports the negotiated encryption Key_Size.

- Reference

  [1] 4.2.5

- Initial Condition

  - The TSPX_min_supported_encryption_key_size IXIT statement gives the value for the minimum encryption Key_Size.
  - The TSPX_max_supported_encryption_key_size IXIT statement gives the value for the maximum encryption Key_Size.
  - The Lower Tester has a minimum encryption Key_Size set to 1.
  - See Initial Condition in Table 4.6-16.

- Test Case Configuration

Test Case	Initial Condition	HCI Set Min Encryption Key Size Support	Encryption Type	Encryption_ Enabled (Step 9)
LMP/ENC/BV-53-C [Key_Size Negotiation as Central - E0]	Section 4.2.3 Encryption	Yes	E0	0x01
LMP/ENC/BV-54-C [Key_Size Negotiation as Central - AES]	Section 4.2.7 AES-CCM Encryption	Yes	AES	0x02
LMP/ENC/BV-63-C [Key_Size Negotiation as Central - E0]	Section 4.2.3 Encryption	No	E0	0x01
LMP/ENC/BV-64-C [Key_Size Negotiation as Central - AES]	Section 4.2.7 AES-CCM Encryption	No	AES	0x02

Table 4.6-16: Key_Size Negotiation as Central test cases

- Test Procedure

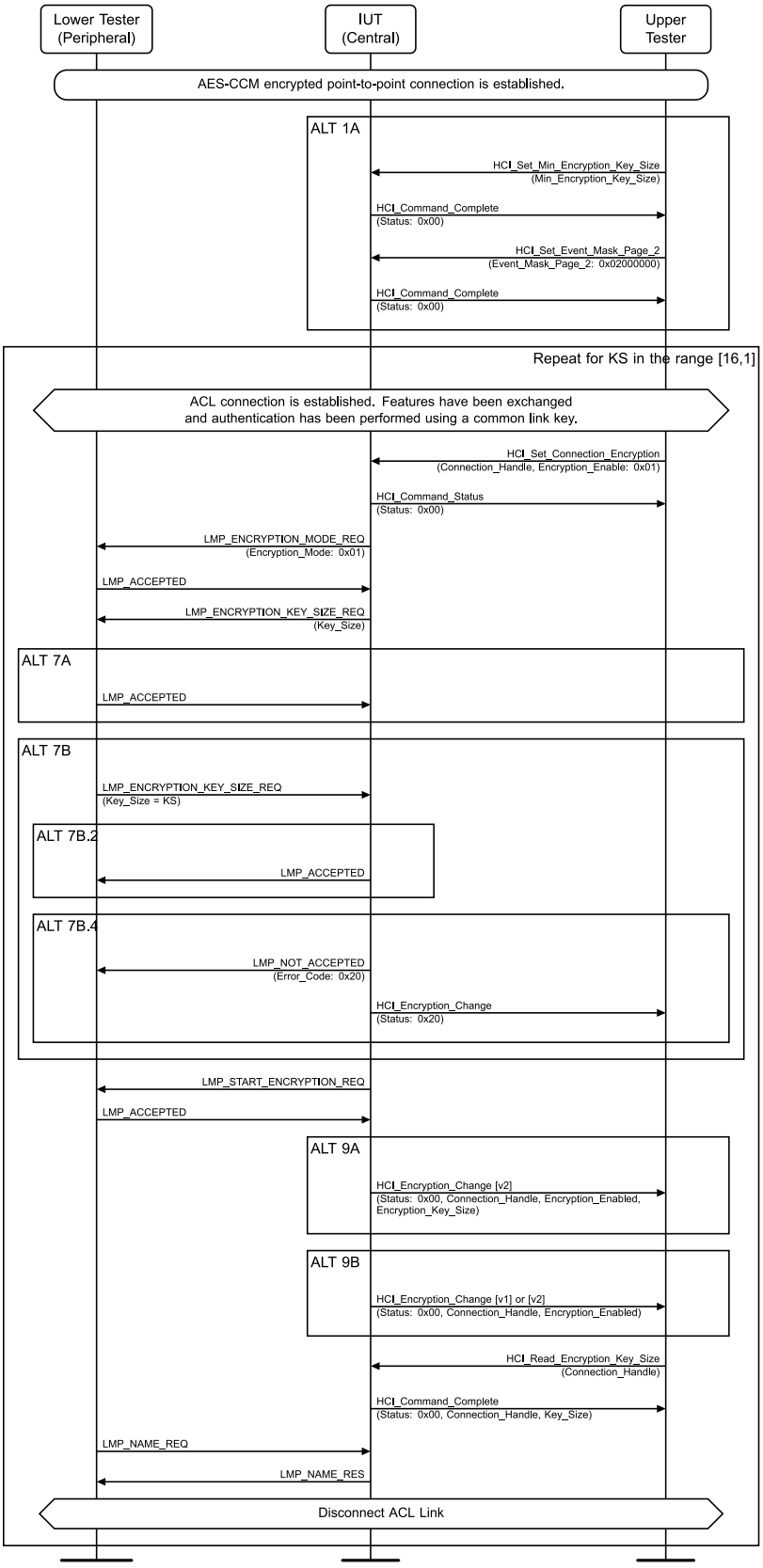


Figure 4.6-40: Key_Size Negotiation as Central MSC

Repeat Steps 2–11 for each encryption Key_Size value KS in the interval [16, 1].

1. Perform either alternative 1A or 1B depending on the IUT's support for HCI_Set_Min_Encryption_Key_Size as specified in Table 4.6-16.
   Alternative 1A (The IUT supports HCI_Set_Min_Encryption_Key_Size):
   - 1A.1. The Upper Tester sends the HCI_Set_Min_Encryption_Key_Size command with the Min_Encryption_Key_Size set to INT((TSPX_min_supported_encryption_key_size + TSPX_max_supported_encryption_key_size) / 2).
   - 1A.2. The Upper Tester sends an HCI_Set_Event_Mask_Page_2 command to the IUT with the Event_Mask_Page_2 set to 0x02000000 and receives a successful HCI_Command_Complete in return.

   Alternative 1B (The IUT does not support HCI_Set_Min_Encryption_Key_Size):
   - 1B.1. The Min_Encryption_Key_Size is considered the same as TSPX_min_supported_encryption_key_size for the rest of the test case below.
2. Establish an ACL connection between the IUT and the Lower Tester.
3. The IUT, at any time before starting the encryption procedure, initiates the exchange of all supported Features (LMP_FEATURES_REQ PDU and, if relevant, LMP_FEATURES_REQ_EXT). The Lower Tester indicates support for both Host and controller Secure Connections support only when AES encryption is indicated in Table 4.6-16.
4. The Lower Tester initiates authentication using a random link key known to the IUT and the Upper Tester.
5. The Upper Tester orders the IUT to enable link encryption, and the IUT sends an LMP_ENCRYPTION_MODE_REQ PDU with the Encryption_Mode field set to 0x01; the Lower Tester replies with an LMP_ACCEPTED PDU.
6. The IUT sends the Lower Tester an LMP_ENCRYPTION_KEY_SIZE_REQ PDU containing a suggested Key_Size that is equal to TSPX_max_supported_encryption_key_size. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
7. Perform either alternative 7A or 7B depending on the suggested Key_Size.
   Alternative 7A (Suggested Key_Size >= KS):
   - 7A.1. The Lower Tester accepts the suggested Key_Size.

   Alternative 7B (Suggested Key_Size < KS):
   - 7B.1. The Lower Tester responds with its own LMP_ENCRYPTION_KEY_SIZE_REQ PDU, suggesting a Key_Size equal to KS.

   The IUT executes either Step 7B.2, 7B.3, or 7B.4.
   - 7B.2. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
   - 7B.3. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
   - 7B.4. Reject the suggested Key_Size by sending an LMP_NOT_ACCEPTED PDU with reason Unsupported LMP Parameter Value/Unsupported LL Parameter (0x20) to the Lower Tester. The IUT sends an HCI_Encryption_Change event with an Unsupported LMP Parameter Value/Unsupported LL Parameter Value HCI error (0x20) to the Upper Tester. Skip to Step 12.
8. The IUT continues the link encryption procedure by sending an LMP_START_ENCRYPTION_REQ PDU, which the Lower Tester accepts with an LMP_ACCEPTED PDU.

9. Perform either alternative 9A or 9B depending on the IUT's support for HCI_Set_Min_Encryption_Key_Size as specified in Table 4.6-16.
   Alternative 9A (The IUT supports HCI_Set_Min_Encryption_Key_Size):
   9A.1. The IUT sends to the Upper Tester an HCI_Encryption_Change [v2] event with the Status field set to 0x00, the Encryption_Key_Size set to the negotiated Key_Size in Step 6, and the Encryption_Enabled field set to the value indicated in Table 4.6-16 for this test.
   Alternative 9B (The IUT does not support HCI_Set_Min_Encryption_Key_Size):
   9B.1. The IUT sends either an HCI_Encryption_Change [v1] or an HCI_Encryption_Change [v2] event to the IUT with the Status field set to 0x00 and the Encryption_Enabled field set to the value indicated in Table 4.6-16 for this test.
10. The Upper Tester issues the HCI Read Encryption Key_Size command, and the IUT responds with an HCI Command Complete event with the Key_Size parameter equal to the negotiated Key_Size in Step 7.
11. The Lower Tester sends an LMP_NAME_REQ PDU, and the IUT replies with an LMP_NAME_RES PDU, verifying that the encryption uses the negotiated Key_Size.
12. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT correctly reports the negotiated encryption Key_Size for each accepted Key_Size value, if the HCI_Read_Encryption_Key_Size command is supported.

  At least one Key_Size value is accepted by the IUT.

  If the HCI_Set_Min_Encryption_Key_Size command is supported, then each accepted Key_Size value in Step 9A >= the Min_Encryption_Key_Size from Step 1.

4.6.3.8 Key_Size Negotiation as Central

- Test Purpose

  Verify that the IUT in the Central role correctly reports the negotiated encryption Key_Size.

- Reference

  [1] 4.2.5

- Initial Condition

  - An IXIT statement, TSPX_min_supported_encryption_key_size, gives the value for the minimum encryption Key_Size.
  - An IXIT statement, TSPX_max_supported_encryption_key_size, gives the value for the maximum encryption Key_Size.
  - The Lower Tester has a minimum encryption Key_Size set to 1.
  - See Initial Condition in Table 4.6-17.

- Test Case Configuration

Test Case	Initial Condition	HCI Set Min Encryption Key Size Support	Encryption Type	Encryption_ Enabled (Step 9)
LMP/ENC/BV-87-C [Key_Size Negotiation as Central - E0]	See the “Default Settings: Encryption” section.	Yes	E0	0x01
LMP/ENC/BV-88-C [Key_Size Negotiation as Central - AES]	See the “Default Settings: AES-CCM Encryption” section.	Yes	AES	0x02
LMP/ENC/BV-89-C [Key_Size Negotiation as Central - E0]	See the “Default Settings: Encryption” section.	No	E0	0x01
LMP/ENC/BV-90-C [Key_Size Negotiation as Central - AES]	See the “Default Settings: AES-CCM Encryption” section.	No	AES	0x02

Table 4.6-17: Key_Size Negotiation as Central test cases

- Test Procedure

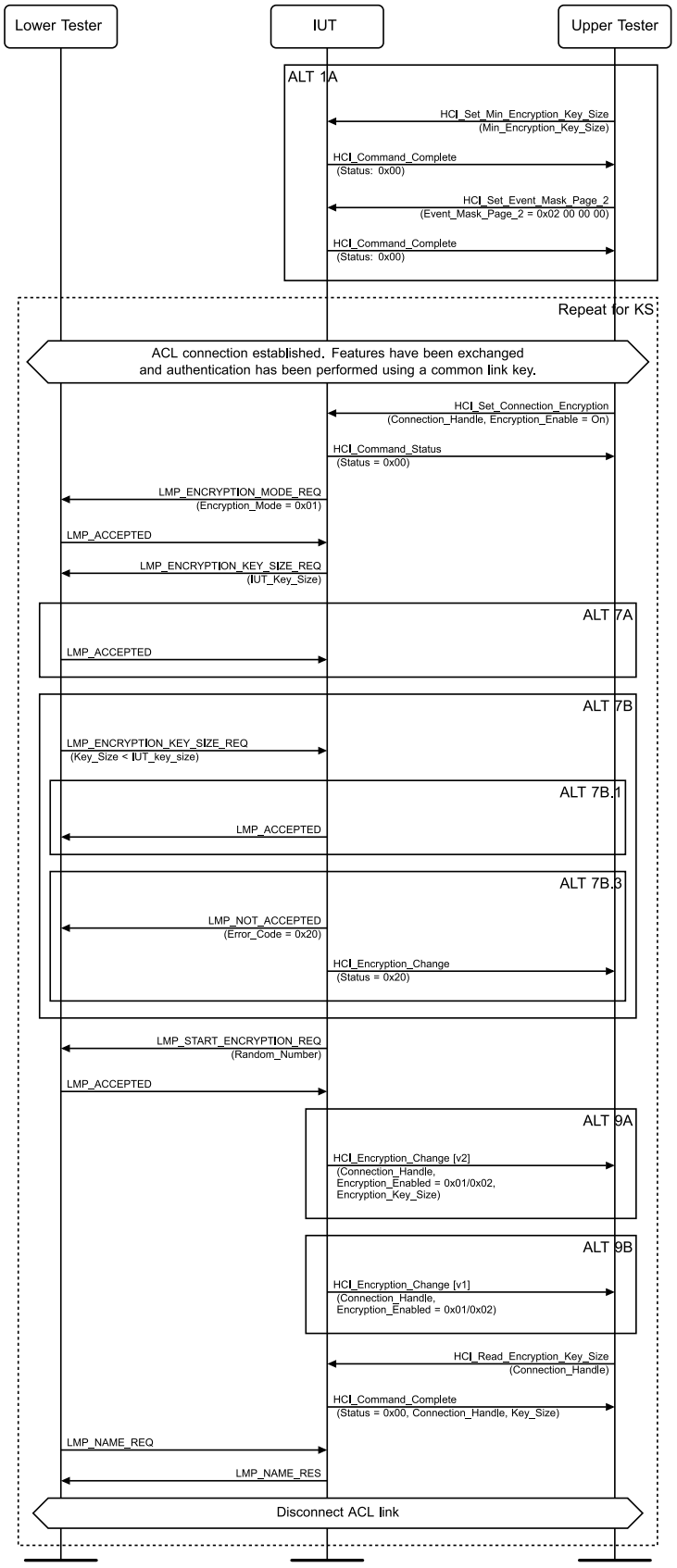


Figure 4.6-41: Key_Size Negotiation as Central MSC

Repeat Steps 2–11 for each encryption Key_Size value KS in the interval [16, 1]:

1. Perform either alternative 1A or 1B depending on the IUT support for HCI Set Min Encryption Key Size as specified in Table 4.6-17.
   Alternative 1A (The IUT supports HCI Set Min Encryption Key Size):
   - 1A.1. The Upper Tester sends the HCI_Set_Min_Encryption_Key_Size command with the Min_Encryption_Key_Size set INT((TSPX_min_supported_encryption_key_size + TSPX_max_supported_encryption_key_size) / 2).
   - 1A.2. The Upper Tester sends an HCI_Set_Event_Mask_Page_2 command to the IUT with the Event_Mask_Page_2 set to 0x02000000 and receives a successful HCI_Command_Complete in return.

   Alternative 1B (The IUT does not support HCI Set Min Encryption Key Size):
   - 1B.1. The Min_Encryption_Key_Size is considered the same as TSPX_min_supported_encryption_key_size for the rest of the test case below.
2. Establish an ACL connection between the IUT and the Lower Tester.
3. The IUT, at any time before starting the encryption procedure, initiates the exchange of all supported Features (LMP_FEATURES_REQ PDU and, if relevant, LMP_FEATURES_REQ_EXT). The Lower Tester indicates support for both Host and controller Secure Connections support only when AES encryption is indicated in Table 4.6-17.
4. The Lower Tester initiates authentication using a random link key known to the IUT and the Upper Tester.
5. The Upper Tester orders the IUT to enable link encryption, and the IUT sends an LMP_ENCRYPTION_MODE_REQ PDU with the encryption_mode field set to 0x01; the Lower Tester replies with an LMP_ACCEPTED PDU.
6. The IUT sends the Lower Tester an LMP_ENCRYPTION_KEY_SIZE_REQ PDU containing a suggested Key_Size that is equal to TSPX_max_supported_encryption_key_size. If the IUT sends a suggested Key_Size that is smaller than 7, the test ends with a Fail verdict.
7. Perform either alternative 7A or 7B depending on the suggested Key_Size.
   Alternative 7A (Suggested Key_Size >= KS):
   - 7A.1. The Lower Tester accepts the suggested Key_Size.

   Alternative 7B (Suggested Key_Size < KS):
   - 7B.1. The Lower Tester responds with its own LMP_ENCRYPTION_KEY_SIZE_REQ PDU, suggesting a Key_Size equal to KS.

   The IUT executes either Step 7B.2, 7B.3, or 7B.4.
   - 7B.2. Accept the suggested Key_Size if KS >= Min_Encryption_Key_Size and KS <= TSPX_max_encryption_key_size.
   - 7B.3. Accept the suggested Key_Size even if KS < Min_Encryption_Key_Size. In this case, the test ends with a Fail verdict.
   - 7B.4. Reject the suggested Key_Size by sending an LMP_NOT_ACCEPTED PDU with reason Unsupported LMP Parameter Value/Unsupported LL Parameter (0x20) to the Lower Tester. The IUT sends an HCI_Encryption_Change event with an Unsupported LMP Parameter Value/Unsupported LL Parameter Value HCI error (0x20) to the Upper Tester. Skip to Step 12.
8. The IUT continues the link encryption procedure by sending an LMP_START_ENCRYPTION_REQ PDU, which the Lower Tester accepts with an LMP_ACCEPTED PDU.

9. Perform either alternative 9A or 9B depending on the IUT support for HCI Set Min Encryption Key Size specified in Table 4.6-17.
   Alternative 9A (The IUT supports HCI Set Min Encryption Key Size):
   9A.1. The IUT sends to the Upper Tester an HCI_Encryption_Change [v2] event with the Status field set to 0x00, the Encryption_Key_Size set to the negotiated Key_Size in Step 6, and the Encryption_Enabled field set to the value indicated in Table 4.6-17 for this test.
   Alternative 9B (The IUT does not support HCI Set Min Encryption Key Size):
   9B.1. The IUT sends either an HCI_Encryption_Change [v1] or an HCI_Encryption_Change [v2] event to the IUT with the Status field set to 0x00 and the Encryption_Enabled field set to the value indicated in Table 4.6-17 for this test.
11. The Upper Tester issues the HCI Read Encryption Key_Size command, and the IUT responds with an HCI Command Complete event with the Key_Size parameter equal to the negotiated Key_Size in Step 7.
10. The Lower Tester sends an LMP_NAME_REQ PDU and the IUT replies with an LMP_NAME_RES PDU, verifying that the encryption uses the negotiated Key_Size.
11. The Lower Tester disconnects the ACL link.

- Expected Outcome

  Pass verdict

  The IUT correctly reports the negotiated encryption Key_Size for each accepted Key_Size value, if HCI Read Encryption Key_Size Command is supported.

  At least one Key_Size value is accepted by the IUT.

  If the HCI_Set_Min_Encryption_Key_Size command is supported, then each accepted Key_Size value in Step 9a >= the Min_Encryption_Key_Size from Step 1.

LMP/ENC/BV-57-C [Broadcast Encryption, Link Key Selection Request]

- Test Purpose

  Verify that a Central IUT properly handles a Link Key Selection request when the Lower Tester rejects the request.

- Reference

  [7] 7.1.18

- Initial Condition

  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester, with the IUT as the Central and the Temporary Link Key in use.

- Test Procedure

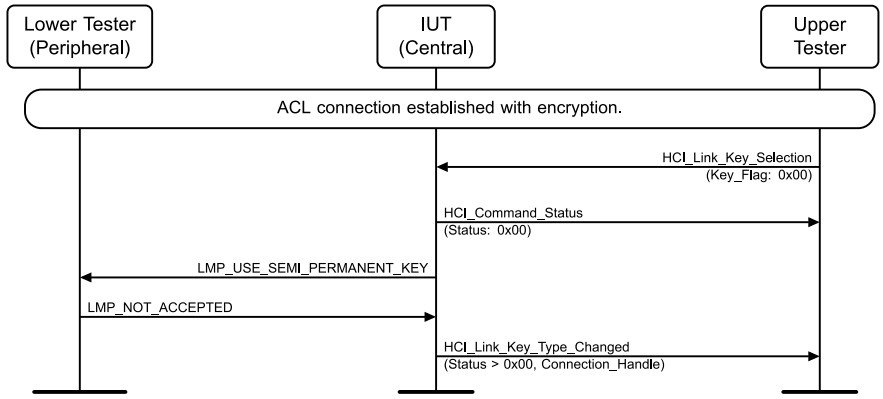


Figure 4.6-42: LMP/ENC/BV-57-C [Broadcast Encryption, Link Key Selection Request] MSC

1. The Upper Tester sends an HCI_Link_Key_Selection command to the IUT with Key_Flag set to 0x00 to indicate use of the semi-permanent link key.
2. The IUT sends a successful HCI_Command_Status event to the Upper Tester.
3. The IUT sends an LMP_USE_SEMI_PERMANENT_KEY PDU to the Lower Tester.
4. The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT with the LMP_USE_SEMI_PERMANENT_KEY PDU Opcode.
5. The IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with Status set to any valid error code.

- Expected Outcome

  Pass verdict

  In Step 5, the IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with Status set to any valid error code and Connection_Handle set to the value of the current connection to the Lower Tester.

LMP/ENC/BV-58-C [Point-To-Point Encryption, Link Key Selection Request]

- Test Purpose

  Verify that a Central IUT properly handles a Link Key Selection request when the Lower Tester rejects the request.

- Reference

  [7] 7.1.18

- Initial Condition

  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester, with the IUT as the Central and the semi-permanent link key in use.

- Test Procedure

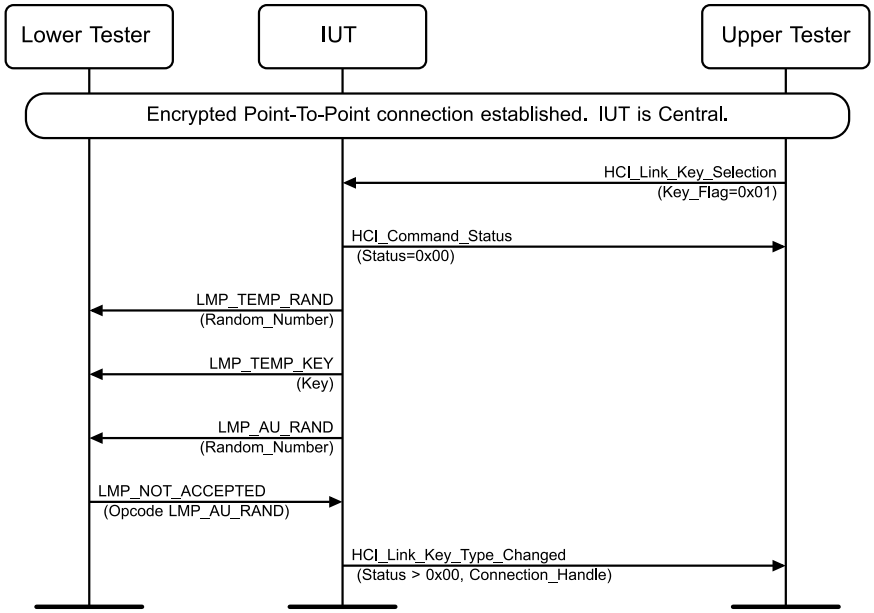


Figure 4.6-43: LMP/ENC/BV-58-C [Point-To-Point Encryption, Link Key Selection Request] MSC

1. The Upper Tester sends an HCI_Link_Key_Selection command to the IUT with Key_Flag set to 0x01 to indicate use of the Temporary Link Key.
2. The IUT sends a successful HCI_Command_Status event to the Upper Tester.
3. The IUT sends an LMP_TEMP_RAND PDU to the Lower Tester.
4. The IUT sends an LMP_TEMP_KEY PDU to the Lower Tester.
5. The IUT sends an LMP_AU_RAND PDU to the Lower Tester.
6. The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT with the LMP_AU_RAND PDU Opcode.
7. The IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with Status set to any valid error code.

- Expected Outcome

  Pass verdict

  In Step 7, the IUT sends an HCI_Link_Key_Type_Changed event to the Upper Tester with an error Status and Connection_Handle set to the value of the current connection to the Lower Tester.

LMP/ENC/BI-02-C [Encryption, Central, Reject Role Switch]

- Test Purpose

  Verify that the IUT rejects a role switch request during the Encryption process.

- Reference

  [1] 4.2.5, 4.4.2

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble.
  - The IUT is the Central and the Lower Tester is the Peripheral.

- The Lower Tester supports Role Switch.
- The Lower Tester initiates authentication and optionally expects a mutual authentication.

- Test Procedure

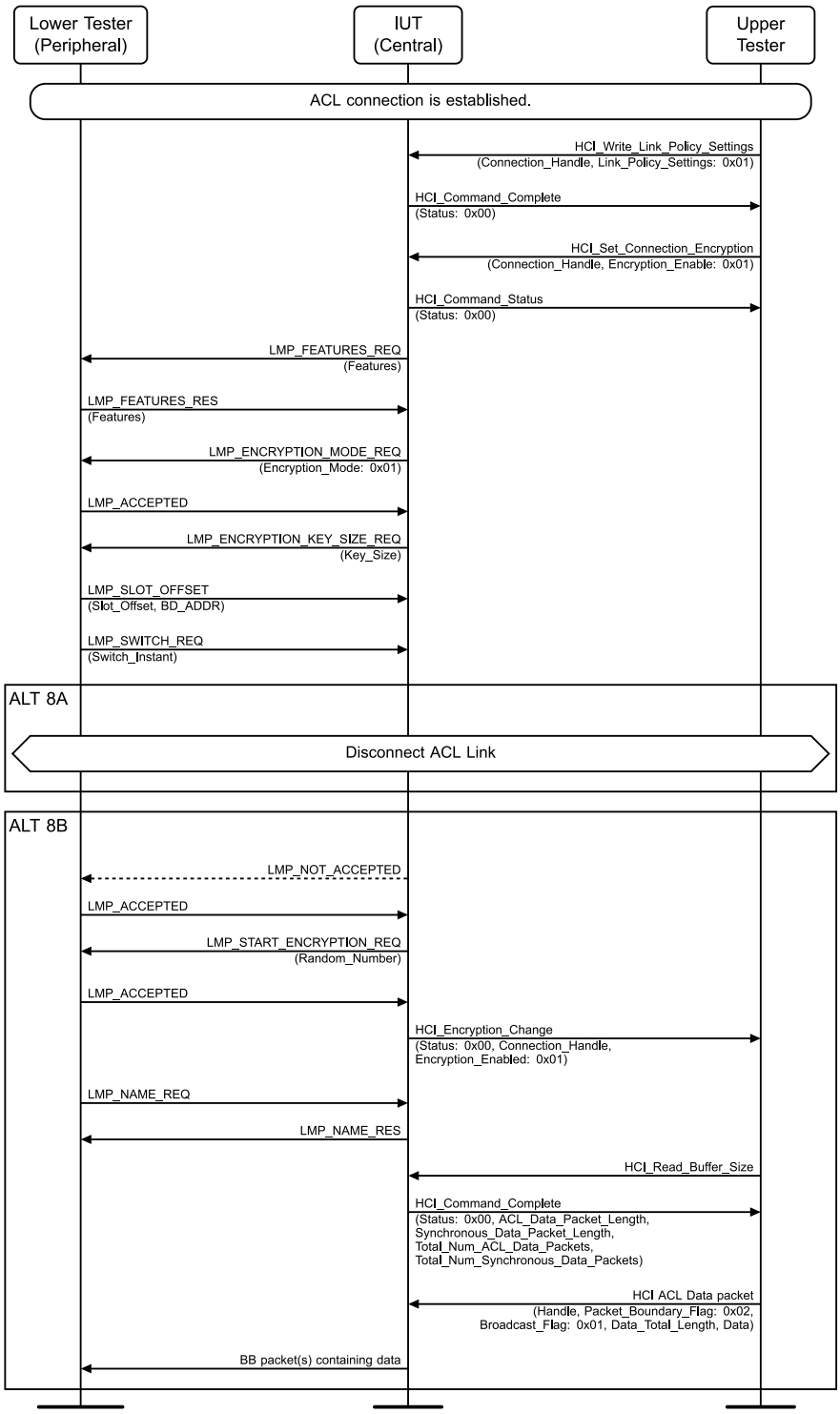


Figure 4.6-44: LMP/ENC/BI-02-C [Encryption, Central, Reject Role Switch] MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x01 and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 and receives a successful HCI_Command_Complete event in response.
3. If the IUT and the Lower Tester have not exchanged LMP Features, the IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter and receives in response an LMP_FEATURES_RES PDU with the Features parameter.
4. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
6. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
7. The Lower Tester sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
8. Perform either alternative 8A or 8B depending on the IUT's response.

   Alternative 8A (The IUT disconnects the ACL Link):

   8A.1. The IUT disconnects the ACL Link.

   Alternative 8B (The IUT continues with the encryption procedure):

   8B.1. Optionally, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
   8B.2. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
   8B.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
   8B.4. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_START_ENCRYPTION_REQ PDU Opcode.
   8B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.
   8B.6. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
   8B.7. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.
   8B.8. The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event with ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, and Total_Num_Synchronous_Data_Packets.
   8B.9. The Upper Tester sends HCI ACL Data packets to the IUT.
   8B.10. The IUT sends BB data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  In alternative 8A, the IUT disconnects the ACL Link.

  In alternative 8B, the IUT optionally sends the LMP_NOT_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester. The IUT sends an HCI_Encryption_Change event to the Upper Tester.

- Notes

  The suggested Key_Size must be within the Lower Tester's Key_Size range.

4.6.4 Encryption – Both Connected Roles

4.6.4.1 Reject Encryption Commands, Unencrypted Connection

- Test Purpose

  Verify that the IUT rejects encryption commands when the IUT and the Lower Tester have an unencrypted connection.

- Reference

  [1] 4.2.5.1, 4.2.5.5

- Initial Condition

  - The IUT is in the role specified in Table 4.6-18.
  - An unencrypted point-to-point connection has been established between the IUT and the Lower Tester.

- Test Case Configuration

Test Case	IUT Role	LMP PDU	Parameter
LMP/ENC/BI-04-C [Reject Encryption Commands, Unencrypted Connection, Peripheral, Encryption Mode Request]	Peripheral	LMP_ENCRYPTION_MODE_REQ	Encryption_Mode = 0
LMP/ENC/BI-05-C [Reject Encryption Commands, Unencrypted Connection, Peripheral, Pause Encryption Request]	Peripheral	LMP_PAUSE_ENCRYPTION_REQ	N/A
LMP/ENC/BI-06-C [Reject Encryption Commands, Unencrypted Connection, Peripheral, Pause Encryption AES Request]	Peripheral	LMP_PAUSE_ENCRYPTION_AES_REQ	N/A
LMP/ENC/BI-07-C [Reject Encryption Commands, Unencrypted Connection, Central, Encryption Mode Request]	Central	LMP_ENCRYPTION_MODE_REQ	Encryption_Mode = 0
LMP/ENC/BI-08-C [Reject Encryption Commands, Unencrypted Connection, Central, Pause Encryption Request]	Central	LMP_PAUSE_ENCRYPTION_REQ	N/A
LMP/ENC/BI-09-C [Reject Encryption Commands, Unencrypted Connection, Central, Pause Encryption AES Request]	Central	LMP_PAUSE_ENCRYPTION_AES_REQ	N/A

Table 4.6-18: Reject Encryption Commands, Unencrypted Connection test cases

- Test Procedure

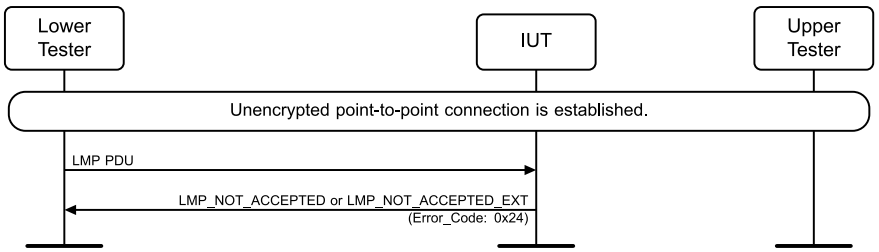


Figure 4.6-45: Reject Encryption Commands, Unencrypted Connection MSC

1. The Lower Tester sends an LMP PDU as specified in Table 4.6-18 to the IUT with the Parameter specified in Table 4.6-18.
2. The IUT sends an LMP_NOT_ACCEPTED or LMP_NOT_ACCEPTED_EXT (as appropriate) with Reason set to 0x24 (LMP PDU not allowed).

- Expected Outcome

  Pass verdict

  The IUT responds to the LMP PDU with an LMP_NOT_ACCEPTED or LMP_NOT_ACCEPTED_EXT (as appropriate) with Reason set to 0x24.

4.6.4.2 Stop AES-CCM Encryption from Host

- Test Purpose

  Verify that the IUT rejects a request to stop AES-CCM encryption upon receiving the appropriate HCI command from the Host.

- Reference

  [1] 4.2.5.4

- Initial Condition

  - The IUT is in the role specified in Table 4.6-19.
  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester.

- Test Case Configuration

Test Case	Role
LMP/ENC/BV-28-C [Stop AES-CCM Encryption from Host]	Peripheral
LMP/ENC/BV-35-C [Initiate AES-CCM Encryption Stop]	Central

Table 4.6-19: Stop AES-CCM Encryption from Host test cases

- Test Procedure

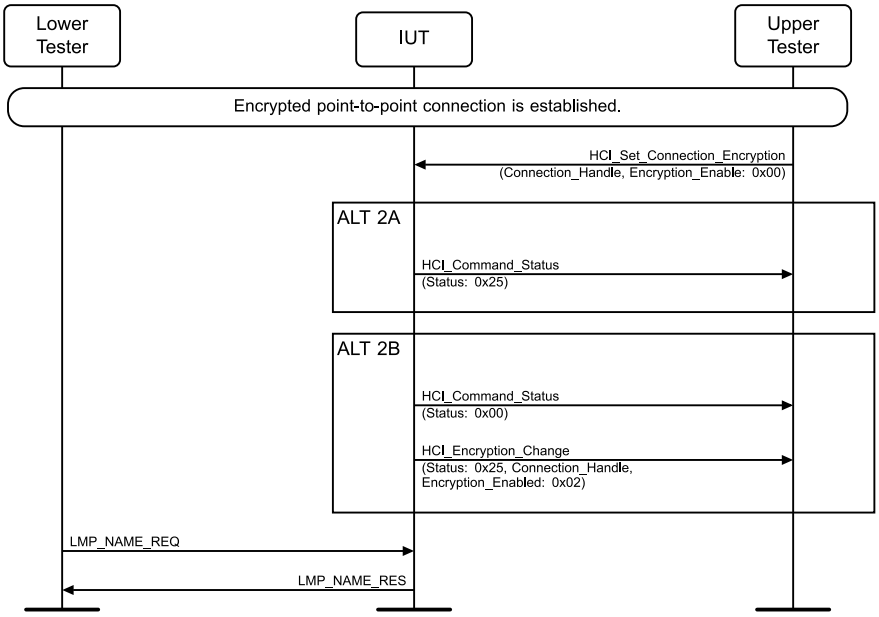


Figure 4.6-46: Stop AES-CCM Encryption from Host MSC

1. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x00.
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT rejects the HCI_Set_Connection_Encryption command):
   - 2A.1. The IUT sends an HCI_Command_Status event to the Upper Tester with Status set to 0x25 (Encryption Mode Not Acceptable).

   Alternative 2B (The IUT accepts the HCI_Set_Connection_Encryption command):
   - 2B.1. The IUT sends a successful HCI_Command_Status event to the Upper Tester.
   - 2B.2. The IUT sends an HCI_Encryption_Change event to the Upper Tester with the Status set to 0x25 (Encryption Mode Not Acceptable), Connection_Handle, and Encryption_Enabled set to 0x02.
3. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
4. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.

- Expected Outcome

  Pass verdict

  In alternative 2A, the IUT sends an HCI_Command_Status event to the Lower Tester with Error_Code set to 0x25 (Encryption Mode Not Acceptable).

  In alternative 2B, the IUT sends a successful HCI_Command_Status event followed by an HCI_Encryption_Change event with Error_Code set to 0x25 (Encryption Mode Not Acceptable) to the Upper Tester.

  In Step 4, the IUT sends the LMP_NAME_RES PDU and proves that encryption is still used.

4.6.4.3 Combating Forged Acknowledgments when AES-CCM Encryption is Enabled

- Test Purpose

  Verify that the IUT periodically sends an LMP_PING_REQ on an idle ACL link on which AES-CCM encryption has been enabled in order to force the other side to transmit an ACL packet (LMP_PING_RES).

- Reference

  [1] 4.1.13

- Initial Condition

  - The IUT is in the role specified in Table 4.6-20.
  - An AES-CCM encrypted point-to-point connection has been established between the IUT and the Lower Tester.

- Test Case Configuration

Test Case	Role
LMP/ENC/BV-29-C [Combating forged acknowledgements when AES-CCM Encryption is enabled]	Peripheral
LMP/ENC/BV-46-C [Combating forged acknowledgements when AES-CCM Encryption is enabled]	Central

Table 4.6-20: Combating Forged Acknowledgments when AES-CCM Encryption is Enabled test cases

- Test Procedure

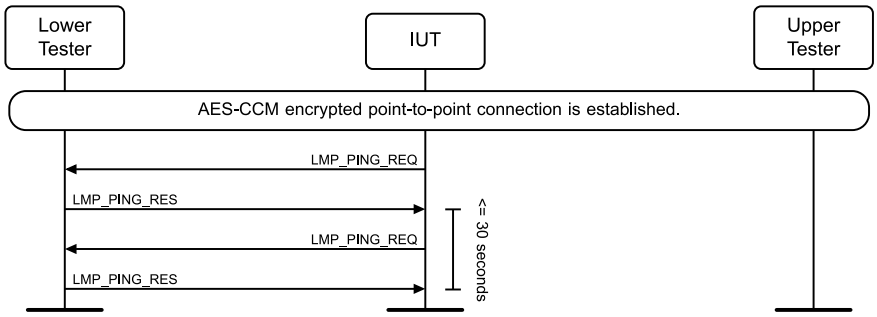


Figure 4.6-47: Combating Forged Acknowledgments when AES-CCM Encryption is Enabled MSC

1. The ACL connection is kept idle, i.e., no ACL-U or ACL-C traffic is exchanged for 60 seconds.
2. The IUT sends the LMP_PING_REQ PDU to the Lower Tester such that the LMP_PING_RES PDUs successfully received by the IUT are less than (or equal to) 30 seconds apart.
3. The Lower Tester responds with an LMP_PING_RES PDU.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_PING_REQ PDU to the Lower Tester so that the LMP_PING_RES PDUs successfully received by the IUT are less than (or equal to) 30 seconds apart.

- Notes

  The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 30 seconds.

4.6.4.4 Stop AES-CCM Encryption

- Test Purpose

  Verify that the IUT rejects a request to stop AES-CCM encryption from the Lower Tester.

- Reference

  [1] 4.2.5.4

- Initial Condition

  - The IUT is in the role specified in Table 4.6-21.
  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester.
  - The IUT has defined its supported LMP Features.

- Test Case Configuration

Test Case	Role
LMP/ENC/BV-27-C [Stop AES-CCM Encryption from Central]	Peripheral
LMP/ENC/BV-36-C [Stop AES-CCM Encryption, Peripheral request]	Central

Table 4.6-21: Stop AES-CCM Encryption test cases

- Test Procedure

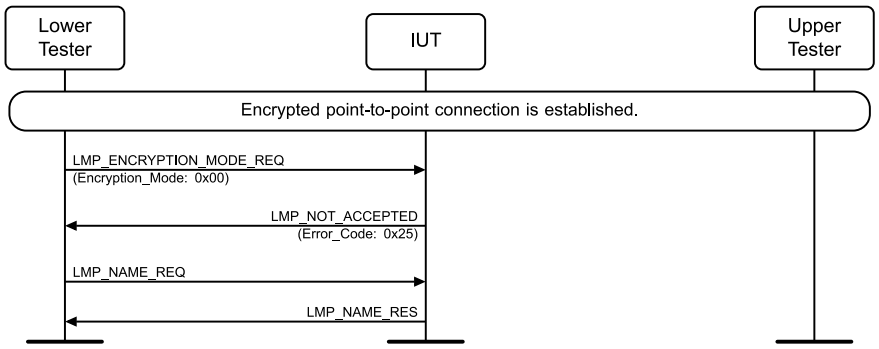


Figure 4.6-48: Stop AES-CCM Encryption MSC

1. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x00.
2. The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_ENCRYPTION_MODE_REQ PDU Opcode and Error_Code set to 0x25 (Encryption Mode Not Allowed).
3. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT.
4. The IUT sends an LMP_NAME_RES PDU to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x25 (Encryption Mode Not Allowed) upon reception of the LMP_ENCRYPTION_MODE_REQ PDU from the Lower Tester and does not stop using encryption.

  In Step 4, the IUT sends the LMP_NAME_RES PDU and proves that encryption is still used.

4.6.4.5 Responding to LMP_PING_REQ when AES-CCM Encryption is Enabled

- Test Purpose

  Verify that the IUT responds to an LMP_PING_REQ sent by the Lower Tester when AES-CCM encryption has been enabled.

- Reference

  [1] 4.1.13

- Initial Condition

  - The IUT is in the role specified in Table 4.6-22.
  - An AES-CCM encrypted point-to-point connection has been established between the IUT and the Lower Tester.

- Test Case Configuration

Test Case	Role
LMP/ENC/BV-30-C [Responding to LMP_PING_REQ when AES-CCM Encryption is enabled]	Peripheral
LMP/ENC/BV-47-C [Responding to LMP_PING_REQ when AES-CCM Encryption is enabled]	Central

Table 4.6-22: Responding to LMP_PING_REQ when AES-CCM Encryption is Enabled test cases

- Test Procedure

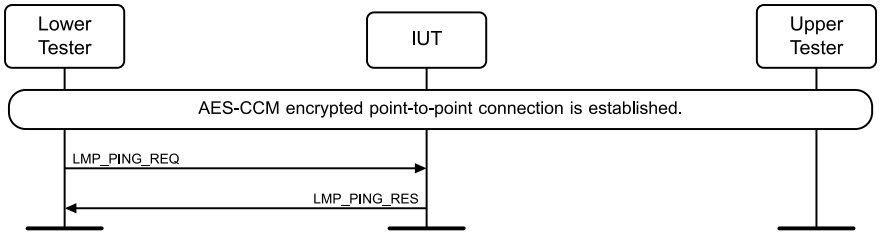


Figure 4.6-49: Responding to LMP_PING_REQ when AES-CCM Encryption is enabled

1. The Lower Tester sends an LMP_PING_REQ PDU.
2. The IUT responds with an LMP_PING_RES PDU.

- Expected Outcome

  Pass verdict

  The IUT responds to every LMP_PING_REQ PDU with an LMP_PING_RES PDU.

4.6.4.6 No Response to LMP_PING_REQ

- Test Purpose

  Verify that the IUT generates the HCI_Authenticated_Payload_Timeout_Expired event when the Lower Tester does not respond to an LMP_PING_REQ sent by the IUT within the Authenticated_Payload_Timeout interval.

- Reference

  [1] 4.1.13

- Initial Condition
  - The IUT is in the role specified in Table 4.6-23.
  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester.
- Test Case Configuration

Test Case	Role
LMP/ENC/BV-31-C [No response to LMP_PING_REQ]	Peripheral
LMP/ENC/BV-48-C [No response to LMP_PING_REQ]	Central

Table 4.6-23: No Response to LMP_PING_REQ test cases

- Test Procedure

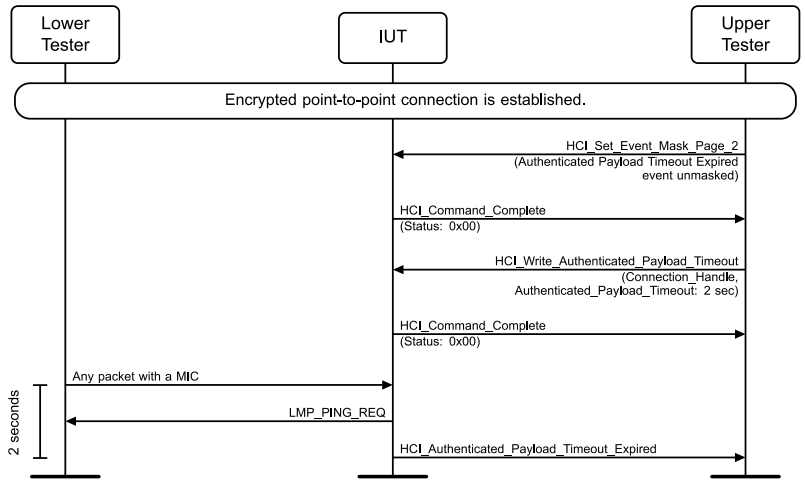


Figure 4.6-50: No response to LMP_PING_REQ MSC

1. The Upper Tester sends an HCI_Set_Event_Mask_Page_2 command to the IUT with the Authenticated Payload Timeout Expired event unmasked and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Write_Authenticated_Payload_Timeout command to the IUT with the Connection_Handle and Authenticated_Payload_Timeout set to 2 seconds and receives a successful HCI_Command_Complete event in response.
3. The ACL connection is kept idle, i.e., no ACL-U or ACL-C traffic is exchanged for 10 seconds.
4. The IUT sends an LMP_PING_REQ PDU to the Lower Tester.
5. The Lower Tester does not respond with an LMP_PING_RES PDU.
6. The IUT sends an HCI_Authenticated_Payload_Timeout_Expired event to the Upper Tester 2 seconds after the last packet that contained a MIC was received by the IUT from the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_PING_REQ PDU to the Lower Tester and sends an HCI_Authenticated_Payload_Timeout_Expired event to the Upper Tester when the Lower Tester does not respond to an LMP_PING_REQ PDU with an LMP_PING_RES PDU.

4.6.4.7 Modified Authentication Payload Timeout

- Test Purpose

  Verify that the IUT as the Peripheral uses the correct value of the Authenticated Payload Timeout set by the Upper Tester.

- Reference

  [1] 4.1.13

- Initial Condition

  - The IUT is in the role specified in Table 4.6-24.
  - An encrypted point-to-point connection has been established between the IUT and the Lower Tester.

- Test Case Configuration

Test Case	Role
LMP/ENC/BV-32-C [Modified Authentication Payload Timeout]	Peripheral
LMP/ENC/BV-49-C [Modified Authentication Payload Timeout]	Central

Table 4.6-24: Modified Authentication Payload Timeout test cases

- Test Procedure

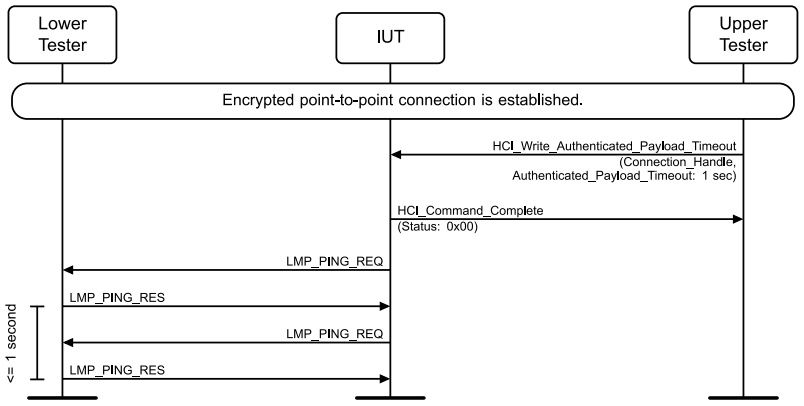


Figure 4.6-51: Modified Authentication Payload Timeout MSC

1. The Upper Tester sends an HCI_Write_Authenticated_Payload_Timeout command to the IUT with the Connection_Handle and Authenticated_Payload_Timeout set to 1 second and receives a successful HCI_Command_Complete event in response.
2. The ACL connection is kept idle, i.e., no ACL-U or ACL-C traffic is exchanged for 2 seconds.
3. The IUT sends the LMP_PING_REQ PDUs to the Lower Tester such that the LMP_PING_RES PDUs successfully received by the IUT are less than (or equal to) 1 second apart.
4. The Lower Tester responds to the IUT with an LMP_PING_RES PDU.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_PING_REQ PDUs such that the LMP_PING_RES PDUs successfully received by the IUT are less than (or equal to) 1 second apart.

- Notes

  The Lower Tester should attempt to not transmit any packets that contain a MIC. However, if this is not possible and the Lower Tester autonomously transmits a data packet that contains a MIC, the Lower Tester should wait another 1 second.

4.7 Information Requests

Verify the correct implementation of the Information requests services.

4.7.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.7.2 Clock_Offset Request – Peripheral

Verify that the Central can request the Clock_Offset anytime during the connection. The IUT is the Peripheral.

LMP/INF/BV-01-C [Clock_Offset Response]

- Test Purpose

  Verify that the IUT responds with the Clock_Offset upon request from the Lower Tester.

- Reference

  [1] 4.3.2

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central.
  - An ACL connection has been established.
- Test Procedure

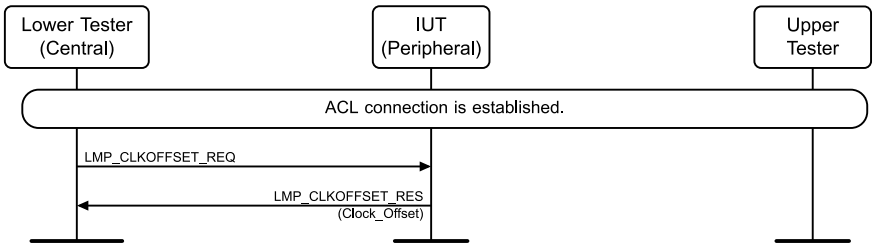


Figure 4.7-1: LMP/INF/BV-01-C [Clock_Offset Response] MSC

1. The Lower Tester sends an LMP_CLKOFFSET_REQ PDU to the IUT.
2. The IUT responds to the Lower Tester with an LMP_CLKOFFSET_RES PDU with the Clock_Offset.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_CLKOFFSET_RES PDU to the Lower Tester upon reception of the LMP_CLKOFFSET_REQ PDU from the Lower Tester.

4.7.3 Clock_Offset Request – Central

Verify that the Central can request the Clock_Offset any time during the connection. The IUT is the Central.

LMP/INF/BV-02-C [Clock_Offset Request]

- Test Purpose

  Verify that the IUT can request the Lower Tester’s Clock_Offset.

- Reference

  [1] 4.3.2

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Central. The Lower Tester is the Peripheral.
  - The IUT must page the Lower Tester to become the Central of the piconet.

- Test Procedure

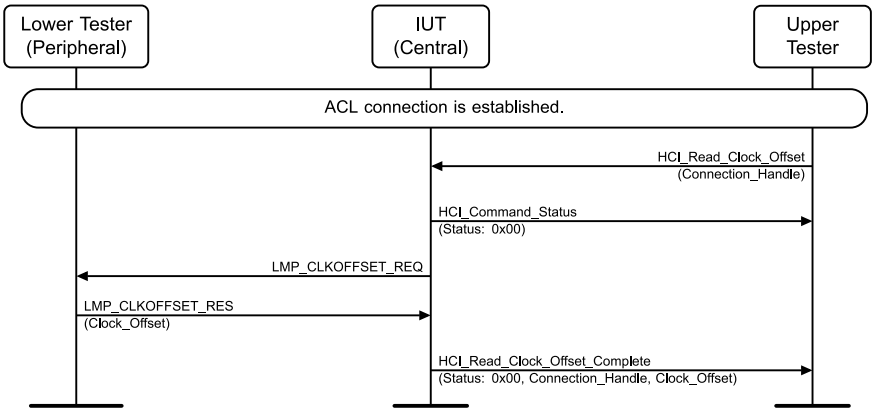


Figure 4.7-2: LMP/INF/BV-02-C [Clock_Offset Request] MSC

1. The Upper Tester sends an HCI_Read_Clock_Offset command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_CLKOFFSET_REQ PDU to the Lower Tester.
3. The Lower Tester responds to the IUT with an LMP_CLKOFFSET_RES PDU with the Clock_Offset.
4. The IUT sends a successful HCI_Read_Clock_Offset_Complete event to the Upper Tester with the Connection_Handle and the Clock_Offset.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_CLKOFFSET_REQ PDU to the Lower Tester after a request by the Upper Tester.

4.7.4 Timing Accuracy Information Request – Both Central and Peripheral

Verify that the IUT can respond to a request for timing accuracy information. The role of the IUT is of no importance.

LMP/INF/BV-05-C [Timing Accuracy Response]

- Test Purpose

  Verify that the IUT responds with timing accuracy information upon request from the Lower Tester.

- Reference

  [1] 4.3.1

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.
  - Drift is defined by TSPX_timing_accuracy_drift_iut, and Jitter is defined by TSPX_timing_accuracy_jitter_iut.

- Test Procedure

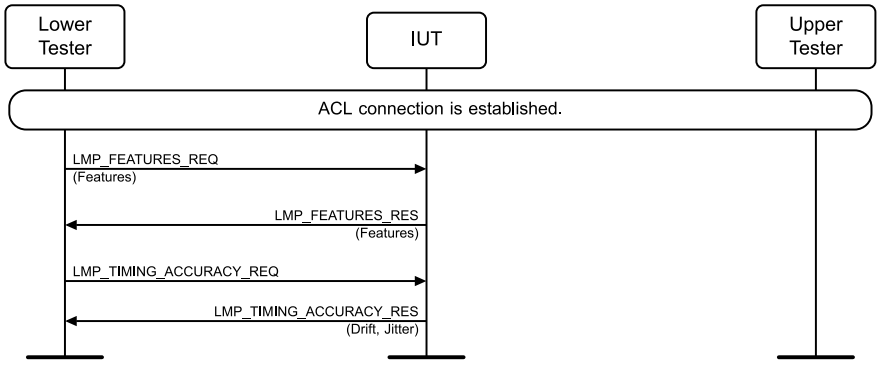


Figure 4.7-3: LMP/INF/BV-05-C [Timing Accuracy Response] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The Lower Tester sends an LMP_TIMING_ACCURACY_REQ PDU to the IUT.
4. The IUT responds to the Lower Tester with an LMP_TIMING_ACCURACY_RES PDU with the Drift and Jitter.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_TIMING_ACCURACY_RES PDU to the Lower Tester containing the defined Drift as TSPX_timing_accuracy_drift_iut and Jitter as TSPX_timing_accuracy_jitter_iut upon reception of the LMP_TIMING_ACCURACY_REQ PDU from the Lower Tester.

4.7.5 LMP Version – Both Central and Peripheral

Verify that the IUT can request the version of the LM protocol or respond to a request for the version of the LM protocol. The role of the IUT is of no importance.

LMP/INF/BV-08-C [Version/Company ID Response]

- Test Purpose

  Verify that the IUT responds with the correct Version number and Company ID upon request from the Lower Tester.

- Reference

  [1] 4.3.3

  Bluetooth Assigned Numbers

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.

- Company ID is defined by TSPX_company_id_iut, the LMP version is defined by TSPX_lmp_version_number_iut, and the Subversion is defined by TSPX_subversion_number_iut.

- Test Procedure

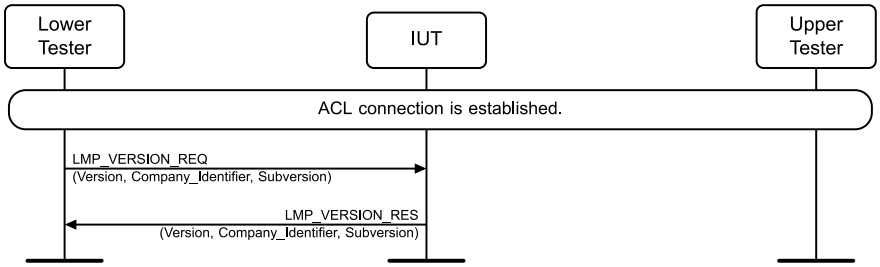


Figure 4.7-4: LMP/INF/BV-08-C [Version/Company ID Response] MSC

1. The Lower Tester sends an LMP_VERSION_REQ PDU to the IUT with the Version, Company_Identifier, and Subversion.
2. The IUT responds to the Lower Tester with an LMP_VERSION_RES PDU with the Version, Company_Identifier, and Subversion.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_VERSION_RES PDU to the Lower Tester containing Version defined by TSPX_lmp_version_number_iut, Company_Identifier defined by TSPX_company_id_iut, and Subversion defined by TSPX_subversion_number_iut upon reception of the LMP_VERSION_REQ PDU from the Lower Tester.

  The Version value sent by the IUT matches the Specification version to which conformance is claimed.

LMP/INF/BV-09-C [Request LMP Version]

- Test Purpose

  Verify that the IUT can request the LMP version from the Lower Tester.

- Reference

  [1] 4.3.3

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT initiates the service.
  - Company ID is defined by TSPX_company_id_iut, the LMP version is defined by TSPX_lmp_version_number_iut, and the Subversion is defined by TSPX_subversion_number_iut.

- Test Procedure

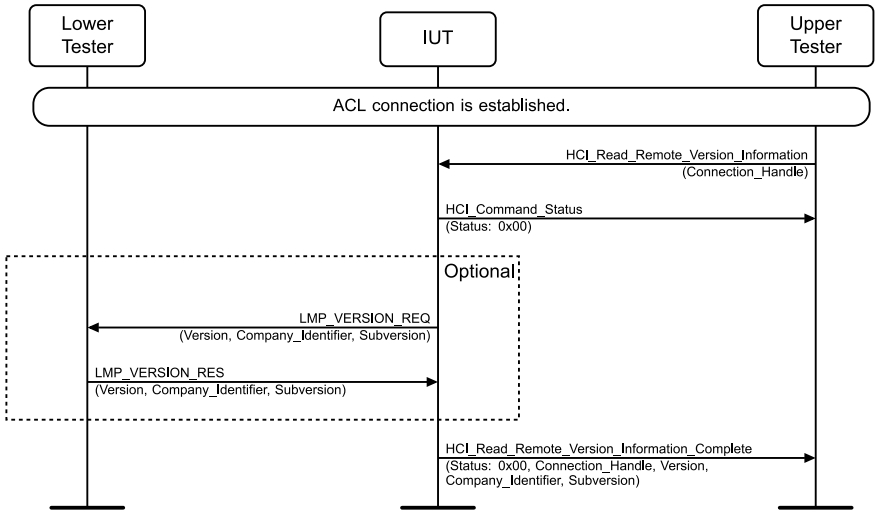


Figure 4.7-5: LMP/INF/BV-09-C [Request LMP Version] MSC

1. The Upper Tester sends an HCI_Read_Remote_Version_Information command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. Optionally, the IUT sends an LMP_VERSION_REQ PDU to the Lower Tester with the Version, Company_Identifier, and Subversion.
3. If Step 2 occurs, the Lower Tester sends an LMP_VERSION_RES PDU to the IUT with the Version, Company_Identifier, and Subversion.
4. The IUT sends a successful HCI_Read_Remote_Version_Information_Complete event to the Upper Tester with the Connection_Handle, Version, Company_Identifier, and Subversion from the LMP_VERSION_RES PDU sent by the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_VERSION_REQ PDU to the Lower Tester containing Version defined by TSPX_lmp_version_number_iut, Company_Identifier defined by TSPX_company_id_iut, and Subversion defined by TSPX_subversion_number_iut.

  The IUT sends the HCI_Read_Remote_Version_Information_Complete event to the Upper Tester containing the values received in the LMP_VERSION_RES PDU from the Lower Tester.

  The Version value sent by the IUT in the LMP_VERSION_REQ matches the specification version to which conformance is claimed.

4.7.6 Supported Features – Both Central and Peripheral

Verify that the IUT can request the LMP Features or respond to a request for the LMP Features. The role of the IUT is of no importance.

LMP/INF/BV-10-C [Supported Features Response]

- Test Purpose

  Verify that the IUT responds with the correct Features supported upon request from the Lower Tester. The Lower Tester initiates the service.

- Reference

  [1] 4.3.4

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.

- Test Procedure

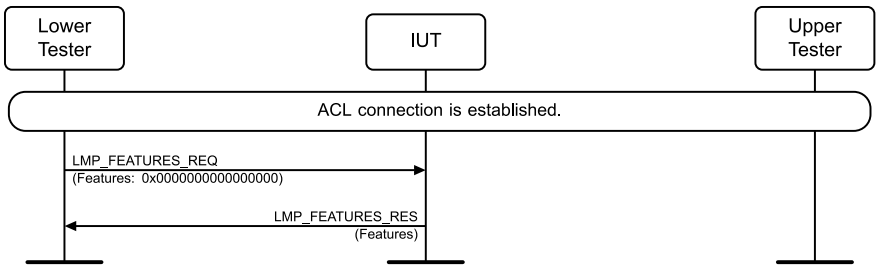


Figure 4.7-6: LMP/INF/BV-10-C [Supported Features Response] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with Features set to 0x0000000000000000 indicating that the Lower Tester does not support any feature.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_FEATURES_RES PDU to the Lower Tester containing Features supported as indicated by Table 2 in the LMP ICS [3] upon reception of the LMP_FEATURES_REQ PDU from the Lower Tester.

LMP/INF/BV-11-C [Supported Features Request]

- Test Purpose

  Verify that the IUT can request for the Features supported by the Lower Tester.

- Reference

  [1] 4.3.4

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT initiates the service.

- Test Procedure

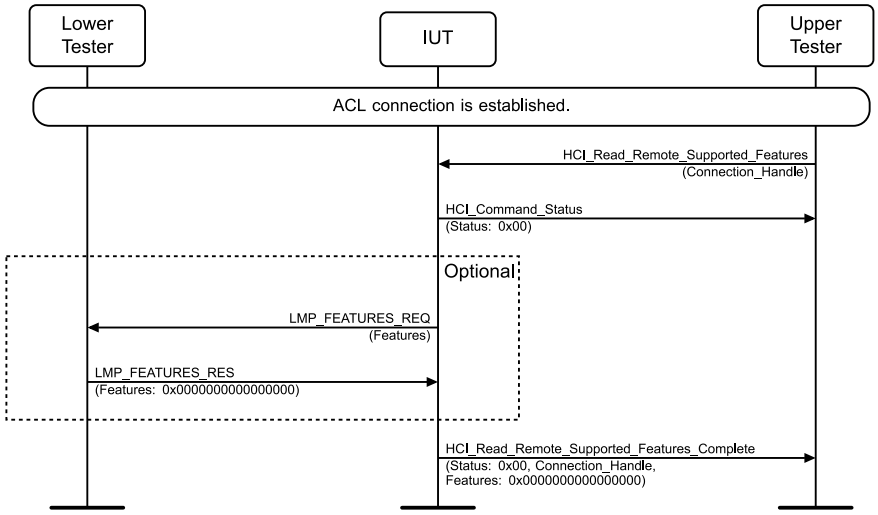


Figure 4.7-7: LMP/INF/BV-11-C [Supported Features Request] MSC

1. The Upper Tester sends an HCI_Read_Remote_Supported_Features command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. Optionally, the IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
3. If Step 2 occurs, the Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with Features set to 0x0000000000000000 indicating that the Lower Tester does not support any feature.
4. The IUT sends a successful HCI_Read_Remote_Supported_Features_Complete event to the Upper Tester with the Connection_Handle and LMP_Features set to 0x0000000000000000.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_FEATURES_REQ PDU to the Lower Tester containing Features supported as indicated by Table 2 in the LMP ICS [3].

  The IUT sends the HCI_Read_Remote_Supported_Features_Complete event to the Upper Tester containing the Features value received in the LMP_FEATURES_RES PDU from the Lower Tester.

LMP/INF/BV-16-C [Extended_Features Request]

- Test Purpose

  Verify that the IUT asks for Extended_Features supported.

- Reference

  [1] 4.3.4

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester’s LMP extended feature bit is set.

- Test Procedure

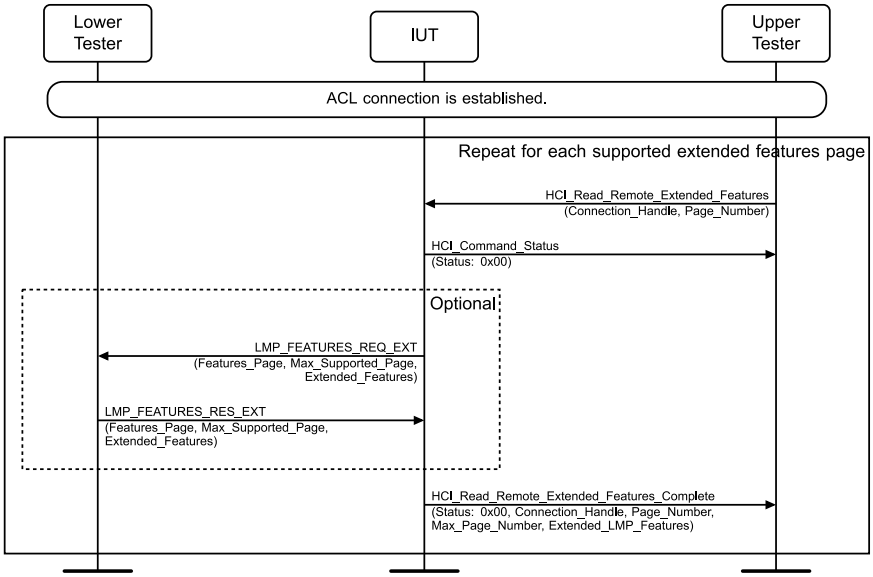


Figure 4.7-8: LMP/INF/BV-16-C [Extended_Features Request] MSC

Repeat Steps 1–4 for each supported extended features page of the Lower Tester starting from 1 to Max_Supported_Page returned in the LMP_FEATURES_RES_EXT PDU, incrementing after every round as Extended Features Page.

1. The Upper Tester sends an HCI_Read_Remote_Extended_Features command to the IUT with the Connection_Handle and Page_Number set to the current Extended Features Page and receives a successful HCI_Command_Status event in response.
2. Optionally, the IUT sends an LMP_FEATURES_REQ_EXT PDU to the Lower Tester with the Features_Page set to the current Extended Features Page, Max_Supported_Page, and Extended_Features.
3. If Step 2 occurs, the Lower Tester responds to the IUT with an LMP_FEATURES_RES_EXT PDU with Features_Page set to the current Extended Features Page, Max_Supported_Page, and Extended_Features.
4. The IUT sends a successful HCI_Read_Remote_Extended_Features_Complete event to the Upper Tester with the Connection_Handle, Page_Number set to the current Extended Features Page, and Max_Page_Number and Extended_LMP_Features from the LMP_FEATURES_RES_EXT.

- Expected Outcome

  Pass verdict

  For each round, the IUT sends an LMP_FEATURES_REQ_EXT PDU to the Lower Tester containing the Extended_Features supported by the IUT as indicated by Table 2 in the LMP ICS [3].

  For each round, the IUT sends an HCI_Read_Remote_Extended_Features_Complete event to the Upper Tester.

LMP/INF/BV-17-C [Extended_Features Response]

- Test Purpose

  Verify that the IUT responds with the correct Extended_Features supported when requested by the Lower Tester.

- Reference

  [1] 4.3.4

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

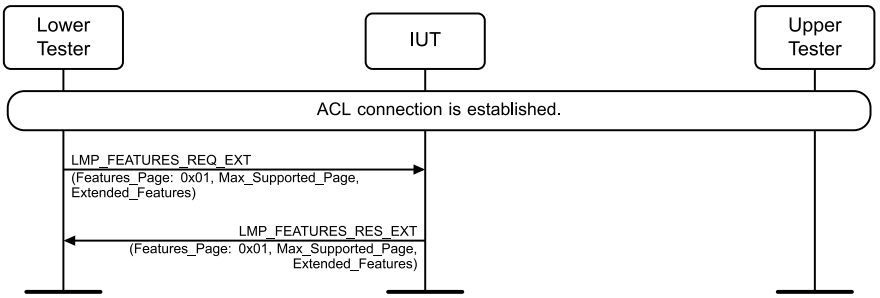


Figure 4.7-9: LMP/INF/BV-17-C [Extended_Features Response] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ_EXT PDU to the IUT with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES_EXT PDU with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_FEATURES_RES_EXT PDU to the Lower Tester containing Extended_Features supported by the IUT as indicated by Table 2 in the LMP ICS [3] upon reception of the LMP_FEATURES_REQ_EXT PDU by the Lower Tester.

4.7.7 Name Request – Both Central and Peripheral

Verify that the IUT can request the name or respond to a request for the name. The role of the IUT is of no importance.

LMP/INF/BV-12-C [Name Response]

- Test Purpose

  Verify that the IUT responds with the correct name upon request from the Lower Tester.

- Reference

  [1] 4.3.5

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.
  - The name of the IUT must have been entered.
- Test Procedure

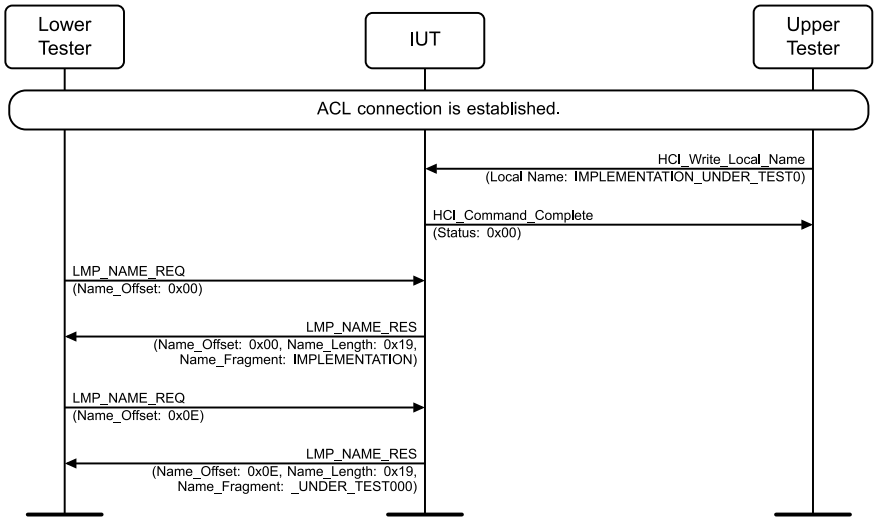


Figure 4.7-10: LMP/INF/BV-12-C [Name Response] MSC

1. The Upper Tester sends an HCI_Write_Local_Name command to the IUT with Local_Name set to IMPLEMENTATION_UNDER_TEST0 and receives a successful HCI_Command_Complete event in response.
2. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT with Name_Offset set to 0x00.
3. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU with Name_Offset set to 0x00, Name_Length set to 0x19, and Name_Fragment set to IMPLEMENTATION.
4. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT with Name_Offset set to 0x0E.
5. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU with Name_Offset set to 0x0E, Name_Length set to 0x19, and Name_Fragment set to _UNDER_TEST000.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NAME_RES PDU to the Lower Tester upon reception of the LMP_NAME_REQ PDU from the Lower Tester.

LMP/INF/BV-13-C [Name Request]

- Test Purpose

  Verify that the IUT can request the name from the Lower Tester.

- Reference

  [1] 4.3.5

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT initiates the service.
  - The name of the IUT must have been entered.
- Test Procedure

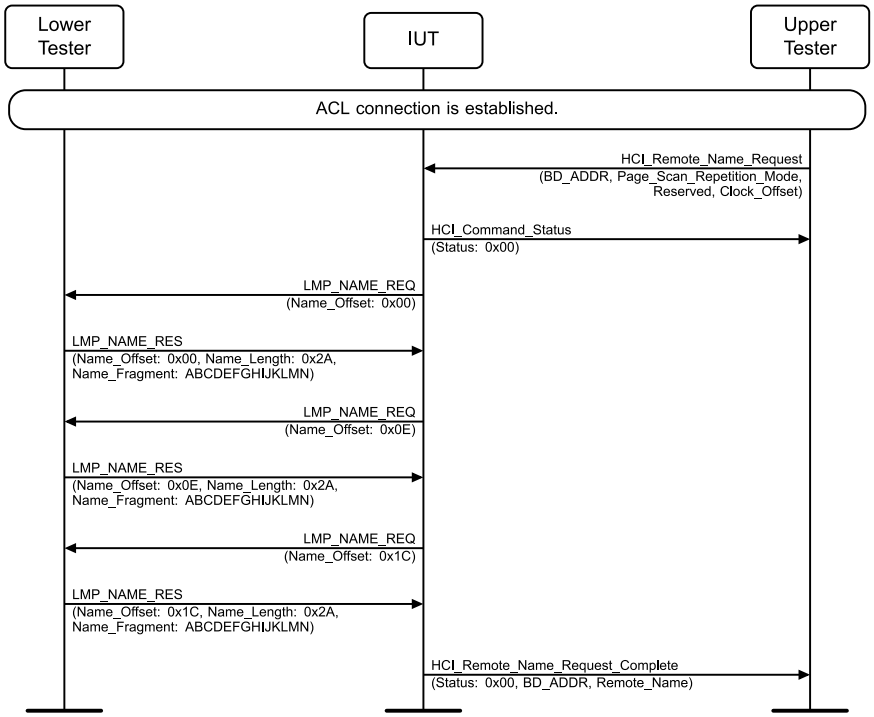


Figure 4.7-11: LMP/INF/BV-13-C [Name Request] MSC

1. The Upper Tester sends an HCI_Remote_Name_Request command to the IUT with the BD_ADDR, Page_Scan_Repetition_Mode, Reserved, and Clock_Offset and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester with Name_Offset set to 0x00.
3. The Lower Tester responds with an LMP_NAME_RES PDU with Name_Offset set to 0x00, Name_Length set to 0x2A, and Name_Fragment set to ABCEDFGHIJKLMN.
4. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester with Name_Offset set to 0x0E.
5. The Lower Tester responds with an LMP_NAME_RES PDU with Name_Offset set to 0x0E, Name_Length set to 0x2A, and Name_Fragment set to ABCEDFGHIJKLMN.
6. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester with Name_Offset set to 0x1C.
7. The Lower Tester responds with an LMP_NAME_RES PDU with Name_Offset set to 0x1C, Name_Length set to 0x2A, and Name_Fragment set to ABCEDFGHIJKLMN.
8. The IUT sends a successful HCI_Remote_Name_Request_Complete event to the Upper Tester with the BD_ADDR and Remote_Name of the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT continues sending LMP_NAME_REQ PDUs to the Lower Tester until the full name is retrieved.

  In Step 8, the correct Remote_Name of the Lower Tester is sent to the Upper Tester in the HCI_Remote_Name_Request_Complete event.

4.7.7.1 Remote Name Request – IUT Initiator

- Test Purpose

  Verify that the IUT responds correctly to a Remote Name Request command from the Host.

- Reference

  [1] 4.3.5

  [7] 7.1.19

  [8] 2.1

- Initial Condition

  - See Section 4.1.4, “Baseband assumptions”.
  - The Lower Tester’s LMP Extended Features bit (bit 63) is set if specified as Yes in Table 4.7-1; otherwise, it is not set.
  - Secure Simple Pairing Mode is enabled by the Host.

- Test Case Configuration

Test Case	Extended Features Bit Enabled on Lower Tester
LMP/INF/BV-18-C [Remote Name Request - IUT Initiator]	Yes
LMP/INF/BV-19-C [Remote Name Request - Legacy remote device without Extended_Features - IUT Initiator]	No

Table 4.7-1: Remote Name Request – IUT Initiator test cases

- Test Procedure

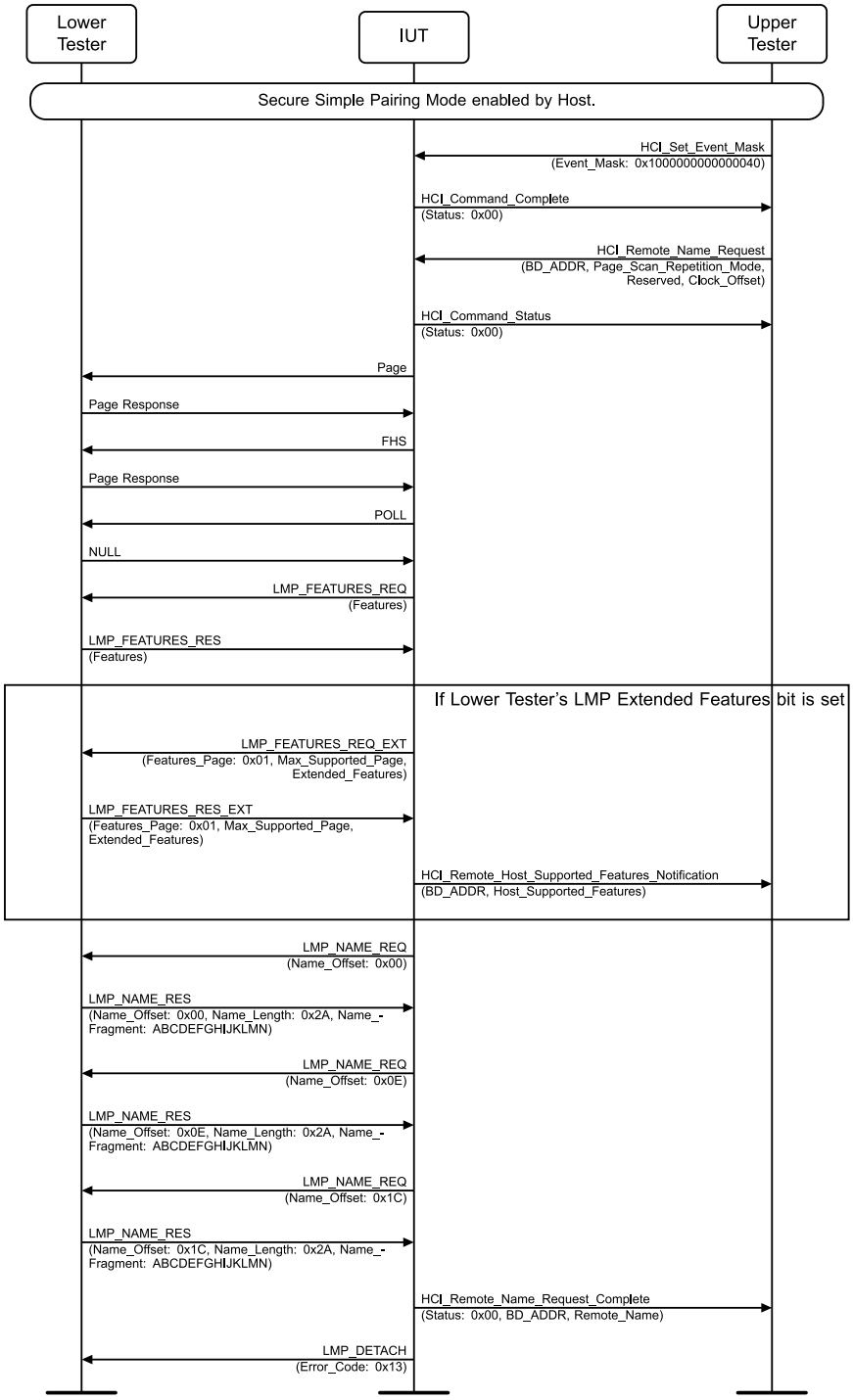


Figure 4.7-12: Remote Name Request – IUT Initiator MSC

1. The Upper Tester sends an HCI_Set_Event_Mask command to the IUT with Event_Mask set to 0x1000000000000040 and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Remote_Name_Request command to the IUT with BD_ADDR, Page_Scan_Repetition_Mode, Reserved, and Clock_Offset and receives a successful HCI_Command_Status event in response.
3. The IUT sends a Page packet to the Lower Tester.
4. The Lower Tester responds to the IUT with a Page Response packet.

5. The IUT sends an FHS packet to the Lower Tester.
6. The Lower Tester responds to the IUT with a Page Response packet.
7. The IUT sends a POLL packet to the Lower Tester.
8. The Lower Tester responds to the IUT with a NULL packet.
9. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
10. The Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with the Features parameter.

If the Lower Tester's LMP Extended Features bit (bit 63) is set as specified in Table 4.7-1, execute Steps 11–13; otherwise, skip to Step 14.

11. The IUT sends an LMP_FEATURES_REQ_EXT PDU to the Lower Tester with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features.
12. The Lower Tester responds to the IUT with an LMP_FEATURES_RES_EXT PDU with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features.
13. The IUT sends an HCI_Remote_Host_Supported_Features_Notification event to the Upper Tester with the BD_ADDR and Host_Supported_Features of the Lower Tester.
14. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester with Name_Offset set to 0x00.
15. The Lower Tester responds with an LMP_NAME_RES PDU with Name_Offset set to 0x00, Name_Length set to 0x2A, and Name_Fragment set to ABCEDFGHIJKLMN.
16. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester with Name_Offset set to 0x0E.
17. The Lower Tester responds with an LMP_NAME_RES PDU with Name_Offset set to 0x0E, Name_Length set to 0x2A, and Name_Fragment set to ABCEDFGHIJKLMN.
18. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester with Name_Offset set to 0x1C.
19. The Lower Tester responds with an LMP_NAME_RES PDU with Name_Offset set to 0x1C, Name_Length set to 0x2A, and Name_Fragment set to ABCEDFGHIJKLMN.
20. The IUT sends a successful HCI_Remote_Name_Request_Complete event to the Upper Tester with the BD_ADDR and Remote_Name of the Lower Tester.
21. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x13 (Remote User Terminated Connection).

- Expected Outcome

  Pass verdict

  If the Lower Tester's LMP Extended Features bit (bit 63) is set as specified in Table 4.7-1:

  - In Step 13, the IUT sends the HCI_Remote_Host_Supported_Features_Notification event to the Upper Tester with Host_Supported_Features matching the Extended_Features parameter in the LMP_FEATURES_RES_EXT PDU.
  - The IUT sends the HCI_Remote_Host_Supported_Features_Notification event to the Upper Tester before sending the HCI_Remote_Name_Request_Complete event to the Host.

  If the Lower Tester's LMP Extended Features bit (bit 63) is not set as specified in Table 4.7-1:

  - The IUT does not send the LMP_FEATURES_REQ_EXT PDU to the Lower Tester, nor does the IUT send the HCI_Remote_Host_Supported_Features_Notification event to the Upper Tester.

  The IUT continues sending LMP_NAME_REQ PDUs to the Lower Tester until the full name is retrieved.

  In Step 20, the correct Remote_Name of the Lower Tester is sent to the Upper Tester in the HCI_Remote_Name_Request_Complete event.

4.7.7.2 LE Features in LMP Feature Set

- Test Purpose

  Verify that the IUT responds with the correct LE Features and extended LE Features supported upon request from the Lower Tester.

- Reference

  [1] 3.2

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Responder.
  - The Lower Tester initiates the service.

- Test Case Configuration

Test Case	LE Supported (Controller)	Simultaneous LE and BR/EDR (Controller)
LMP/INF/BV-22-C [LE Features in LMP Feature Set]	Bit Not Set	Bit Not Set
LMP/INF/BV-23-C [LE Features in LMP Feature Set, LE Supported]	Bit Set	Bit Not Set
LMP/INF/BV-24-C [LE Features in LMP Feature Set, LE Supported and Simultaneous LE and BR/EDR]	Bit Set	Bit Set

Table 4.7-2: LE Features in LMP Feature Set test cases

- Test Procedure

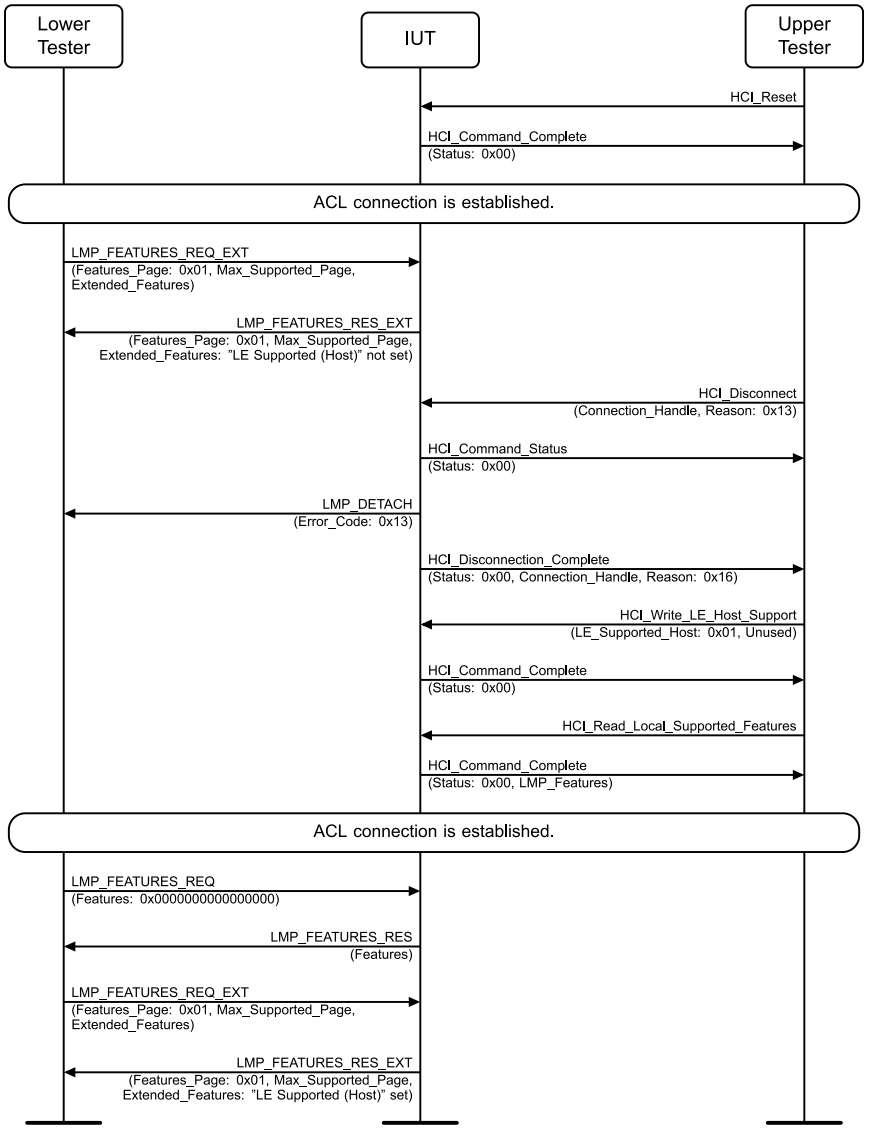


Figure 4.7-13: LE Features in LMP Feature Set MSC

1. The Upper Tester sends an HCI_Reset command to the IUT and receives a successful HCI_Command_Complete event in response.
2. An ACL link is established between the IUT and the Lower Tester.
3. The Lower Tester sends an LMP_FEATURES_REQ_EXT PDU to the IUT with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features.
4. The IUT responds to the Lower Tester with an LMP_FEATURES_RES_EXT PDU with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features with “LE Supported (Host)” bit not set.
5. The Upper Tester sends an HCI_Disconnect command to the IUT with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection) and receives a successful HCI_Command_Status event in response.
6. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x13 (Remote User Terminated Connection).
7. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x16 (Connection Terminated By Local Host).

8. The Upper Tester sends an HCI_Write_LE_Host_Support command to the IUT with LE_Supported_Host set to 0x01 and Unused and receives a successful HCI_Command_Complete event in response.
9. The Upper Tester sends an HCI_Read_Local_Supported_Features command to the IUT and receives a successful HCI_Command_Complete event LMP_Features indicating support for “LE Supported (Controller)” and “Simultaneous LE and BR/EDR to Same Device Capable (Controller)” based on Table 4.7-2.
10. An ACL link is established between the IUT and the Lower Tester.
11. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with Features set to 0x0000000000000000.
12. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with Features indicating support for “LE Supported (Controller)” and “Simultaneous LE and BR/EDR to Same Device Capable (Controller)” based on Table 4.7-2.
13. The Lower Tester sends an LMP_FEATURES_REQ_EXT PDU to the IUT with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features.
14. The IUT responds to the Lower Tester with an LMP_FEATURES_RES_EXT PDU with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features with “LE Supported (Host)” bit set.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_FEATURES_RES_EXT PDU to the Lower Tester with Extended_Features indicating that the “LE Supported (Host)” bit is not set.

  In Step 12, the IUT sends the LMP_FEATURES_RES PDU to the Lower Tester with Features indicating support for “LE Supported (Controller)” and “Simultaneous LE and BR/EDR to Same Device Capable (Controller)” based on Table 4.7-2.

  In Step 14, the IUT sends the LMP_FEATURES_RES_EXT PDU to the Lower Tester with Extended_Features with the “LE Supported (Host)” bit set.

4.7.8 Invalid Packet Handling

LMP/LIH/BI-06-C [Ignore LLID = 0b00]

- Test Purpose

  Verify that the IUT ignores an LMP command with LLID = 0b00.

- Reference

  [1] 4.3.5

  [9] 6.6.2

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

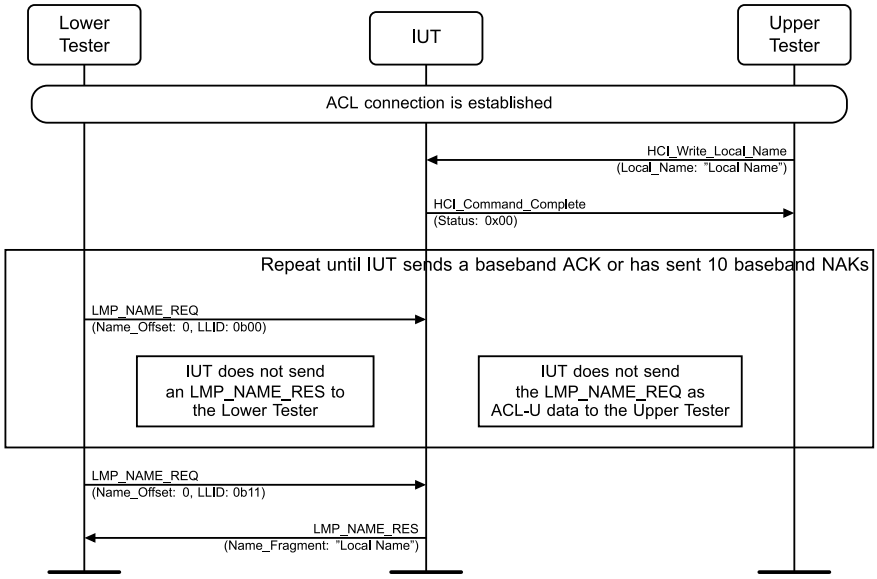


Figure 4.7-14: LMP/LIH/BI-06-C [Ignore LLID = 0b00] MSC

1. The Upper Tester sends an HCI_Write_Local_Name command to the IUT with Local_Name set to “Local Name” and receives a successful HCI_Command_Complete event in response.

Repeat Step 2 until the IUT sends a baseband ACK or has sent 10 baseband NAKs.

2. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT with Name_Offset set to 0 and LLID set to 0b00.
3. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT with Name_Offset set to 0 and LLID set to 0b11.
4. The IUT responds to the Lower Tester with an LMP_NAME_RES PDU with Name_Fragment set to “Local Name”.

- Expected Outcome

Pass verdict

After receiving each LMP_NAME_REQ PDU from the Lower Tester with LLID set to 0b00, the IUT does not send an LMP_NAME_RES PDU to the Lower Tester, nor does the IUT send the LMP_NAME_REQ as ACL-U data to the Upper Tester.

4.7.8.1 Ignore LMP packets with the wrong packet type

- Test Purpose

Verify that the IUT will correctly ignore ACL packets that have the LLID set to 0b11 (LMP) but are not DM1 and not DV packets.

- Reference

[1] 4.3.3

[9] 6.6.2

- Initial Condition
  - The IUT is in the role specified in Table 4.7-3 and in the CONNECTION state (Active mode, ACL link).
  - The IUT's supported ACL packet types other than DM1 or DV are defined by TSPX_non_LMP_ACL_Packet_Types.
- Test Case Configuration

Test Case	IUT Role
LMP/LIH/BI-07-C [Ignore LMP packets with the wrong packet type]	Central
LMP/LIH/BI-08-C [Ignore LMP packets with the wrong packet type]	Peripheral

Table 4.7-3: Ignore LMP packets with the wrong packet type test cases

- Test Procedure

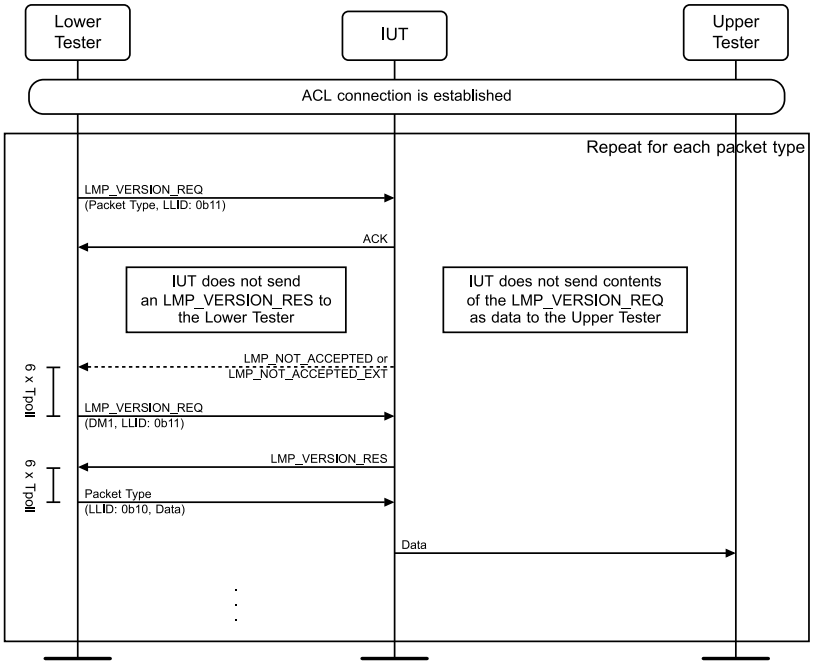


Figure 4.7-15: Ignore LMP packets with the wrong packet type MSC

Perform the following steps for each packet type listed in IXIT item TSPX_non_LMP_ACL_Packet_Types.

1. The Lower Tester sends a packet with the type specified in the IXIT item, 6 octets long, and containing the LMP_VERSION_REQ PDU, and with LLID set to 0b11.
2. The IUT acknowledges the packet in Step 1.
3. The IUT does not send an LMP_VERSION_RES PDU to the Lower Tester or the contents of the LMP_VERSION_REQ PDU as data to the Upper Tester.
4. Optionally, the IUT sends an LMP_NOT_ACCEPTED or LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester.
5. The Lower Tester waits at least 6 times T_{poll}.
6. The Lower Tester sends an LMP_VERSION_REQ PDU in a DM1 packet with LLID set to 0b11.
7. The IUT acknowledges the PDU in Step 5 and replies with an LMP_VERSION_RES PDU.
8. The Lower Tester waits at least 6 times T_{poll}.

9. The Lower Tester sends at least five packets with the type specified in the IXIT item with maximum length, random octets of content, and LLID set to 0b10.
10. The IUT acknowledges each PDU in Step 8 and sends all the data to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT does not send an LMP_VERSION_RES PDU to the Lower Tester or data to the Upper Tester.

  In Step 10, the IUT sends the data to the Upper Tester.

LMP/LIH/BI-09-C [Ignore APB Packets with LMP LLID 0b11, DM3/DM5]

- Test Purpose

  Verify that the IUT will correctly ignore APB packets with LMP LLID 0b11.

- Reference

  [1] 4.3.3

  [9] 6.6.2

- Initial Condition
  - The IUT is the Peripheral and in the CONNECTION state (Active mode, APB link).
  - The Lower Tester is the Central and in the CONNECTION state (Active mode, APB link).
  - A specific set of supported packet types is defined by TSPX_non_LMP_APB_Packet_Types.

- Test Procedure

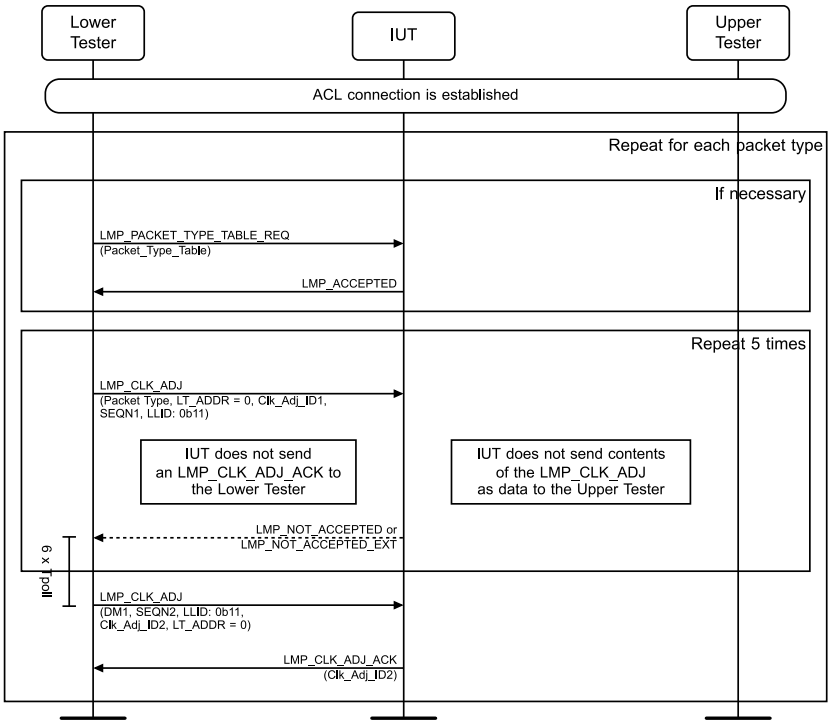


Figure 4.7-16: Ignore APB Packets with LMP LLID 0b11 MSC

Perform the following steps for each packet type listed in IXIT item TSPX_non_LMP_APB_Packet_Types.

1. If necessary, the Lower Tester sends an LMP_PACKET_TYPE_TABLE_REQ PDU with a packet type table corresponding to the type specified in the IXIT item. The IUT replies with an LMP_ACCEPTED PDU.
2. Perform Steps 3–5 a total of five times.
3. The Lower Tester sends a packet with the type specified in the IXIT item, LT_ADDR = 0, 15 octets long, and containing an LMP_CLK_ADJ PDU with LLID set to 0b11. All five instances use the same SEQN and Clk_Adj_ID values.
4. The IUT does not send an LMP_CLK_ADJ_ACK PDU to the Lower Tester or the contents of the LMP_CLK_ADJ PDU as data to the Upper Tester but optionally sends an LMP_NOT_ACCEPTED or LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester.
5. The Lower Tester waits at least 6 times T_{poll}.
6. The Lower Tester sends a DM1 packet with LT_ADDR = 0, 15 octets long, and containing an LMP_CLK_ADJ PDU with LLID set to 0b11, and using a different SEQN than in Step 3.
7. The IUT sends an LMP_CLK_ADJ_ACK PDU to the Lower Tester with Clk_Adj_ID set to the same value used in Step 6.

- Test Condition

  The values for Clk_Adj_ID in Step 3 and in Step 6 for a packet type are different from each other and from the values used for the other packet types (i.e., if there are P packet types, then 2P different values are used).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT does not send an LMP_CLK_ADJ_ACK PDU to the Lower Tester or data to the Upper Tester.

  In Step 7, the IUT sends an LMP_CLK_ADJ_ACK PDU to the Lower Tester.

LMP/LIH/BI-10-C [LMP PDU Incorrect Length]

- Test Purpose

  Verify that the IUT properly handles LMP PDUs that either have valid parameters followed by extra data or are too short to hold all the parameters. If the PDU is too long and has valid parameters followed by extra data, then the IUT either ignores the extra data or responds with an LMP_NOT_ACCEPTED PDU. If the PDU is too short to hold all the parameters, then the IUT either continues with implementation-specific values or responds with an LMP_NOT_ACCEPTED PDU.

- Reference

  [1] 2.5, 4.3.5

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

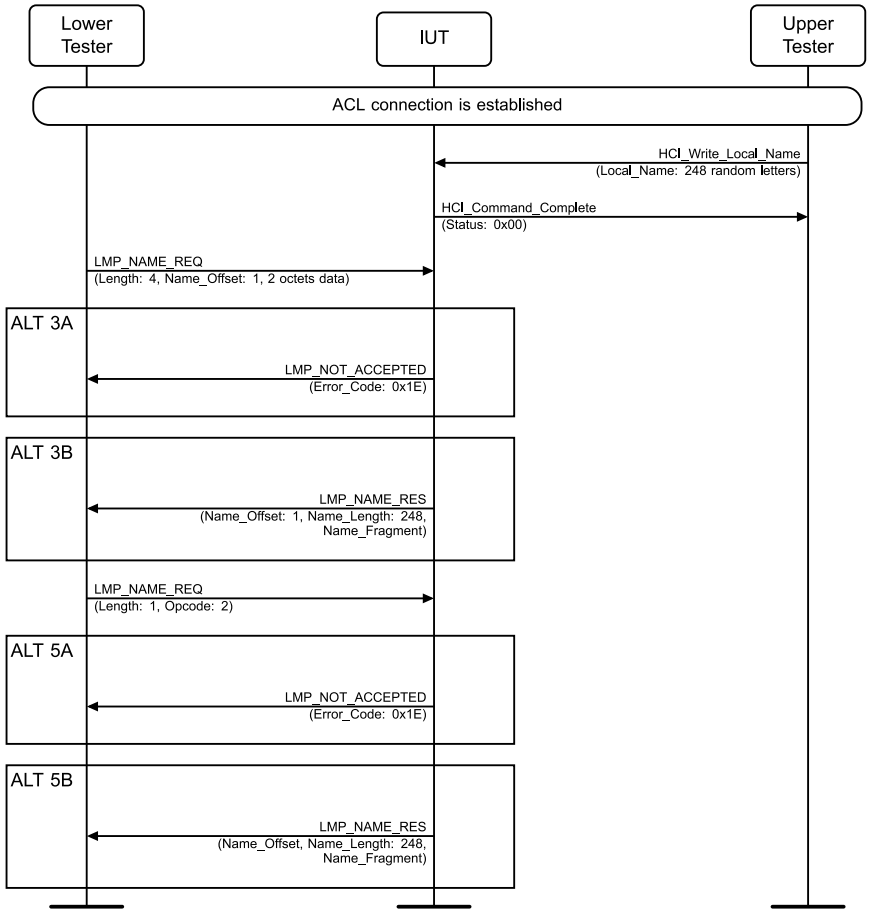


Figure 4.7-17 LMP/LIH/BI-10-C [LMP PDU Incorrect Length] MSC

1. The Upper Tester sends an HCI_Write_Local_Name command to the IUT with Local Name set to 4248 random letters in the range of 0x41 to 5A or 0x61 to 0x7A and receives a successful HCI_Command_Complete event in return.
2. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT with the PDU length set to 4, Name_Offset set to 1, and an additional 2 octets of random data.
3. Perform either alternative 3A or 3B depending on the IUT's response.
   Alternative 3A (The IUT rejects the name request):
   - 3A.1. The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1E (Invalid LMP Parameters).

   Alternative 3B (The IUT sends the name response):
   - 3B.1. The IUT sends an LMP_NAME_RES PDU to the Lower Tester with Name_Offset set to 1, Name_Length set to 248, and Name_Fragment set to octets 1 to 14 of the name sent in Step 1.
4. The Lower Tester sends an LMP_NAME_REQ PDU to the IUT with the PDU length set to 1 and the opcode set to 0x01.
5. Perform either alternative 5A or 5B depending on the IUT's response.
   Alternative 5A (The IUT rejects the name request):
   - 5A.1. The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1E (Invalid LMP Parameters).

Alternative 5B (The IUT sends the name response):

5B.1. The IUT sends an LMP_NAME_RES PDU to the Lower Tester with Name_Offset set to a value less than 248, Name_Length set to 248, and Name_Fragment set to the correct remaining octets of data sent in Step 1 starting at the position specified by Name_Offset.

- Expected Outcome

  Pass verdict

  In Steps 3A.1 and 5A.1, the IUT rejects the name request with error code set to 0x1E.

  In Step 3B.1, the IUT responds with octets 1 to 14 of the name sent in Step 1.

  In Step 5B.1, the IUT responds with Name_Offset < 248.

  In Step 5B.1, the IUT responds with correct values of the name sent in Step 1 starting at the position specified by Name_Offset in Name_Fragment.

4.8 Link Handling

Verify the correct implementation of the Link Handling services.

4.8.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.8.2 Role Switch – Peripheral

Verify that an IUT can request a role switch. The IUT is the Peripheral.

LMP/LIH/BV-01-C [Initiate Role Switch]

- Test Purpose

  Verify that the IUT can request to become a Central and carry out all necessary messages.

- Reference

  [1] 4.4.2

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Peripheral and initiates the switch. The Lower Tester is the Central.
- Test Procedure

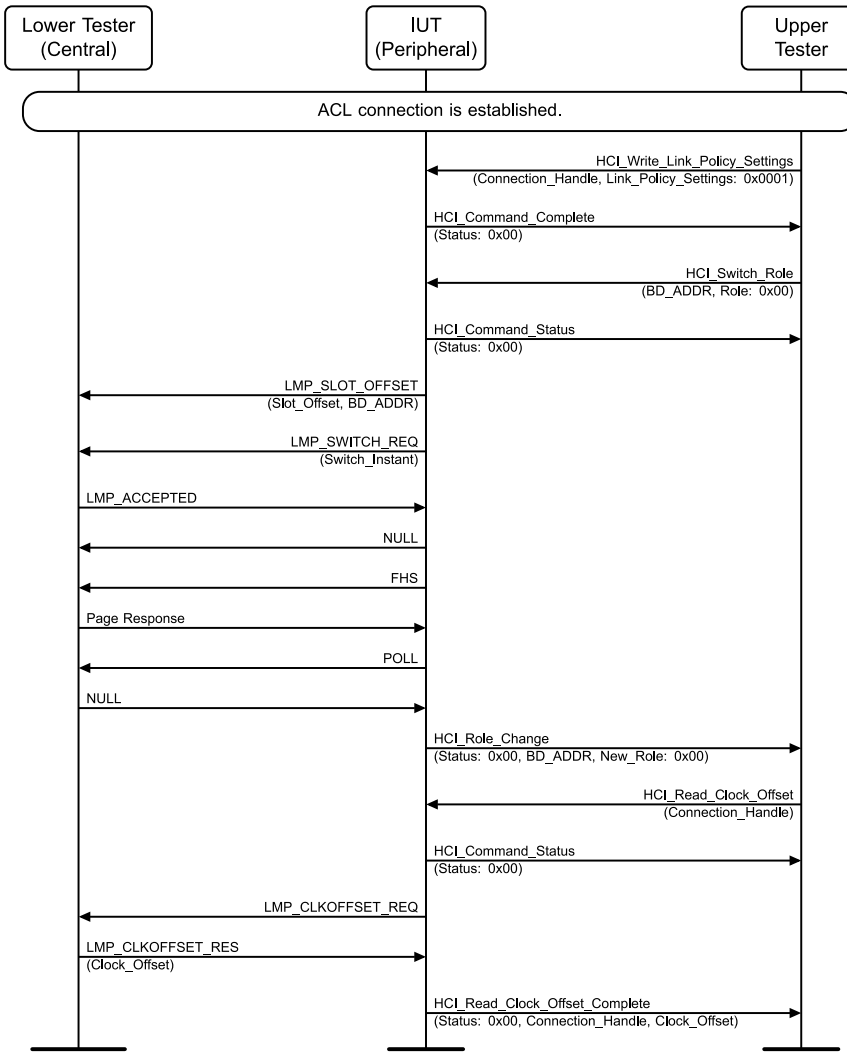


Figure 4.8-1: LMP/LIH/BV-01-C [Initiate Role Switch] MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0001 and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
3. The IUT sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the Lower Tester.
4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
5. The IUT sends a NULL packet to the Lower Tester.
6. The IUT sends an FHS packet to the Lower Tester.

7. The Lower Tester responds to the IUT with a Page Response packet.
8. The IUT sends a POLL packet to the Lower Tester.
9. The Lower Tester responds to the IUT with a NULL packet.
10. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).
11. The Upper Tester sends an HCI_Read_Clock_Offset to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
12. The IUT sends an LMP_CLKOFFSET_REQ PDU to the Lower Tester.
13. The Lower Tester responds to the IUT with an LMP_CLKOFFSET_RES PDU with the Clock_Offset.
14. The IUT sends a successful HCI_Read_Clock_Offset_Complete event to the Upper Tester with the Connection_Handle and Clock_Offset from Step 13.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SWITCH_REQ PDU to the Lower Tester.

  The IUT becomes the Central of the piconet.

LMP/LIH/BV-79-C [Role Switch at Setup, Peripheral]

- Test Purpose

  Verify that the IUT can request a role switch correctly during connection setup.

- Reference

  [1] 4.1.1

- Initial Condition

  - See the “Default settings” section.
  - The Lower Tester is the Central and the IUT is the Peripheral.

- Test Procedure

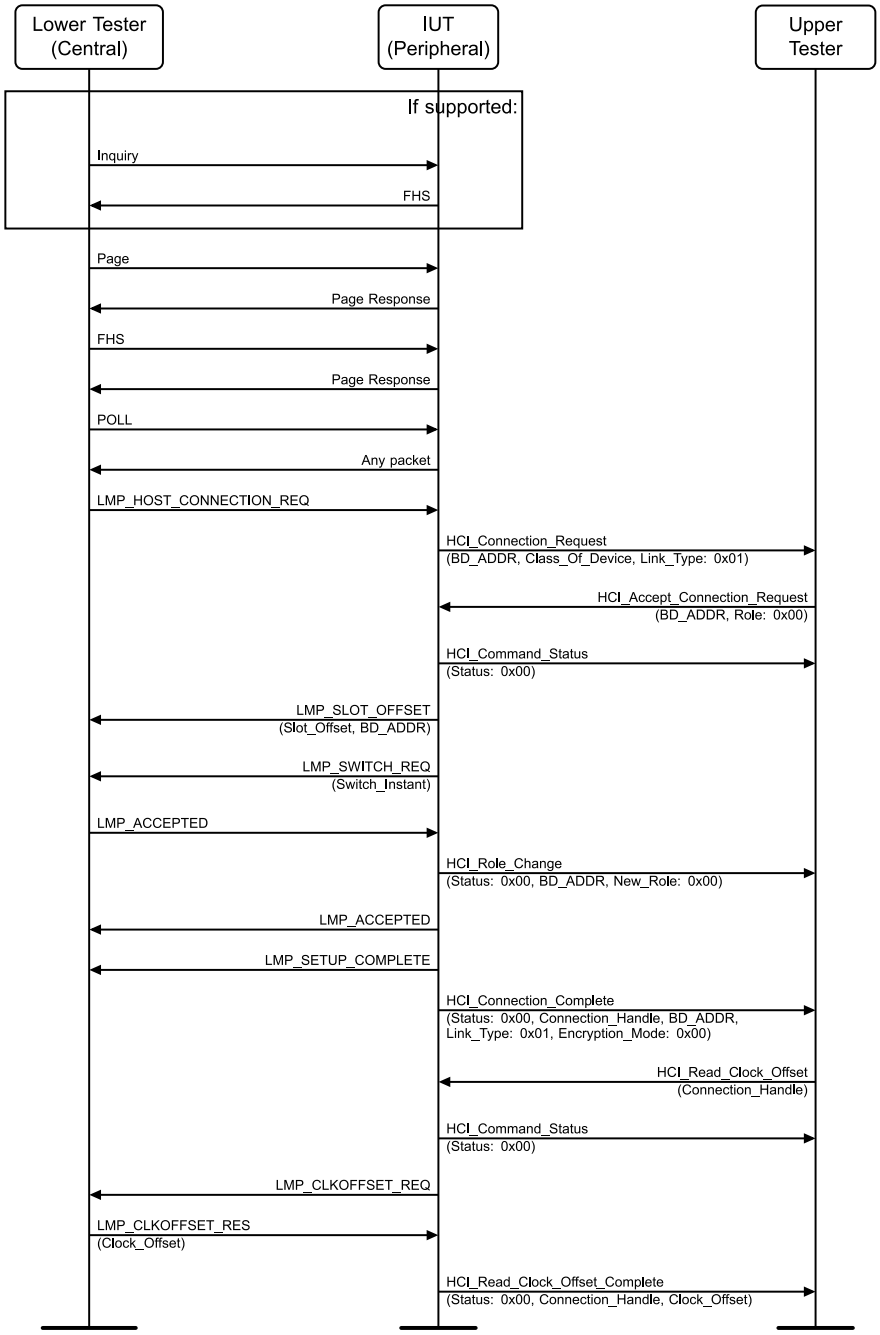


Figure 4.8-2: LMP/LIH/BV-79-C [Role Switch at Setup, Peripheral] MSC

1. If the IUT supports Inquiry, the Lower Tester sends an Inquiry packet to the IUT and the IUT responds to the Lower Tester with an FHS packet.
2. The Lower Tester sends a Page packet to the IUT.
3. The IUT responds to the Lower Tester with a Page Response packet.
4. The Lower Tester sends an FHS packet to the IUT.
5. The IUT responds to the Lower Tester with a Page Response packet.
6. The Lower Tester sends a POLL packet to the IUT.
7. The IUT responds to the Lower Tester with any packet.
8. The Lower Tester sends an LMP_HOST_CONNECTION_REQ PDU to the IUT.

9. The IUT sends an HCI_Connection_Request event to the Upper Tester with the BD_ADDR, Class_Of_Device, and Link_Type set to 0x01 (ACL).
10. The Upper Tester responds with an HCI_Accept_Connection_Request with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
11. The IUT sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the Lower Tester.
12. The Lower Tester responds with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
13. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).
14. The IUT sends an LMP_ACCEPTED PDU with the LMP_HOST_CONNECTION_REQ PDU Opcode and LMP_SETUP_COMPLETE PDU to the Lower Tester.
15. The Lower Tester responds to the IUT with an LMP_SETUP_COMPLETE PDU.
16. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).
17. The Upper Tester sends an HCI_Read_Clock_Offset command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
18. The IUT sends an LMP_CLKOFFSET_REQ PDU to the Lower Tester.
19. The Lower Tester responds to the IUT with an LMP_CLKOFFSET_RES PDU with the Clock_Offset.
20. The IUT sends a successful HCI_Read_Clock_Offset_Complete event with the Connection_Handle and Clock_Offset.

- Expected Outcome

  Pass verdict

  In Step 11, the IUT sends the LMP_SLOT_OFFSET PDU followed by the LMP_SWITCH_REQ PDU to the Lower Tester upon reception of the LMP_HOST_CONNECTION_REQ PDU from the Lower Tester.

  The IUT sends the LMP_CLKOFFSET_REQ PDU to the Lower Tester and, upon reception of the LMP_CLKOFFSET_RES PDU, sends the HCI_Read_Clock_Offset_Complete event to the Upper Tester with the Clock_Offset.

LMP/LIH/BV-144-C [Rejected Role Switch request at Setup, Peripheral]

- Test Purpose

  Verify that the IUT properly handles the Lower Tester rejecting a role switch request during connection setup.

- Reference

  [1] 4.1.1

- Initial Condition

  - See the “Default settings” section.
  - The Lower Tester is the Central and the IUT is the Peripheral.

- Test Procedure

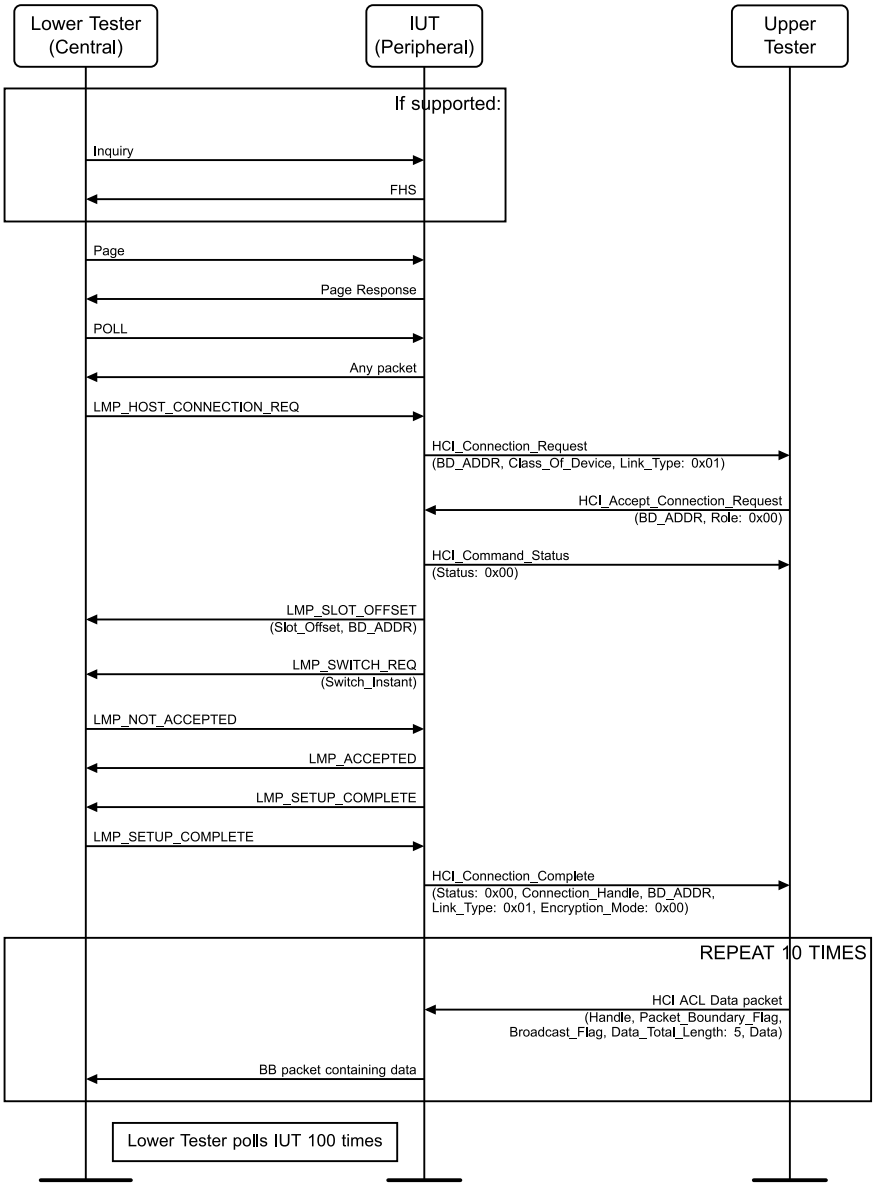


Figure 4.8-3: LMP/LIH/BV-144-C [Rejected Role Switch request at Setup, Peripheral] MSC

1. If the IUT supports Inquiry, the Lower Tester sends an Inquiry packet to the IUT and the IUT responds to the Lower Tester with an FHS packet.
2. The Lower Tester sends a Page packet to the IUT.
3. The IUT responds to the Lower Tester with a Page Response packet.
4. The Lower Tester sends a POLL packet to the IUT.
5. The IUT responds to the Lower Tester with any packet.
6. The Lower Tester sends an LMP_HOST_CONNECTION_REQ PDU to the IUT.
7. The IUT sends an HCI_Connection_Request event to the Upper Tester with the BD_ADDR, Class_Of_Device, and Link_Type set to 0x01 (ACL).
8. The Upper Tester responds with an HCI_Accept_Connection_Request with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
9. The IUT sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the Lower Tester.

10. The Lower Tester responds with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
11. The IUT sends an LMP_ACCEPTED PDU with the LMP_HOST_CONNECTION_REQ PDU Opcode and LMP_SETUP_COMPLETE PDU to the Lower Tester.
12. The Lower Tester responds to the IUT with an LMP_SETUP_COMPLETE PDU.
13. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).

Repeat Steps 14–15 10 times.

14. The Upper Tester sends an HCI ACL Data packet to the IUT.
15. The IUT sends BB packets containing data to the Lower Tester.
16. The Lower Tester sends 100 POLL packets to the IUT.

- Expected Outcome

  Pass verdict

  In Step 9, the IUT sends the LMP_SLOT_OFFSET PDU followed by the LMP_SWITCH_REQ PDU to the Lower Tester upon reception of the LMP_HOST_CONNECTION_REQ PDU from the Lower Tester.

  In Step 15, the IUT sends data packets to the Lower Tester after the role switch request was rejected and the connection completed.

  In Step 16, the IUT continues to respond to at least 90 out of 100 POLL packets sent by the Lower Tester.

LMP/LIH/BI-04-C [Reject Role Switch at Setup, Peripheral]

- Test Purpose

  Verify that the IUT rejects a role switch request during the connection setup.

- Reference

  [1] 4.1.1, 4.4.2

- Initial Condition

  - See “Link Handling” in the “Default settings” section.
  - The Lower Tester is the Central and the IUT is the Peripheral.

- Test Procedure

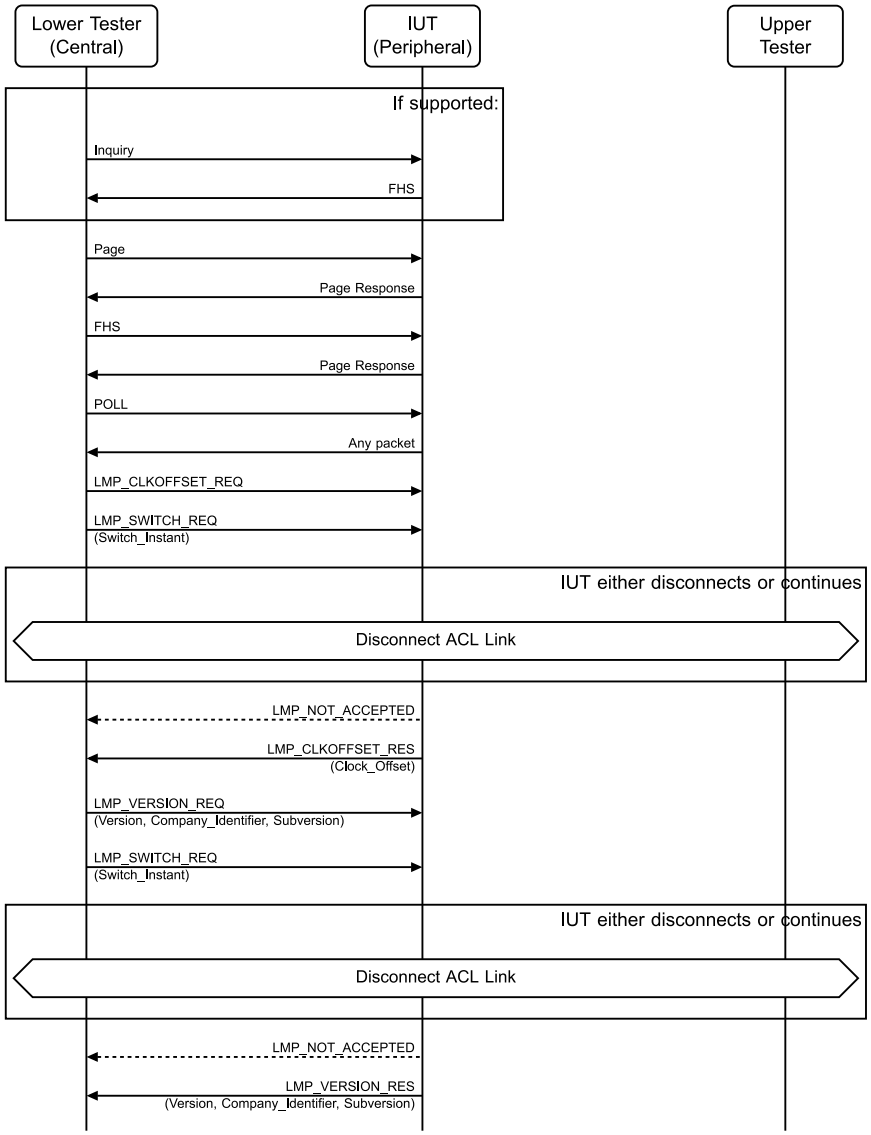


Figure 4.8-4: LMP/LIH/BI-04-C [Reject Role Switch at Setup, Peripheral] MSC – Page 1 of 2

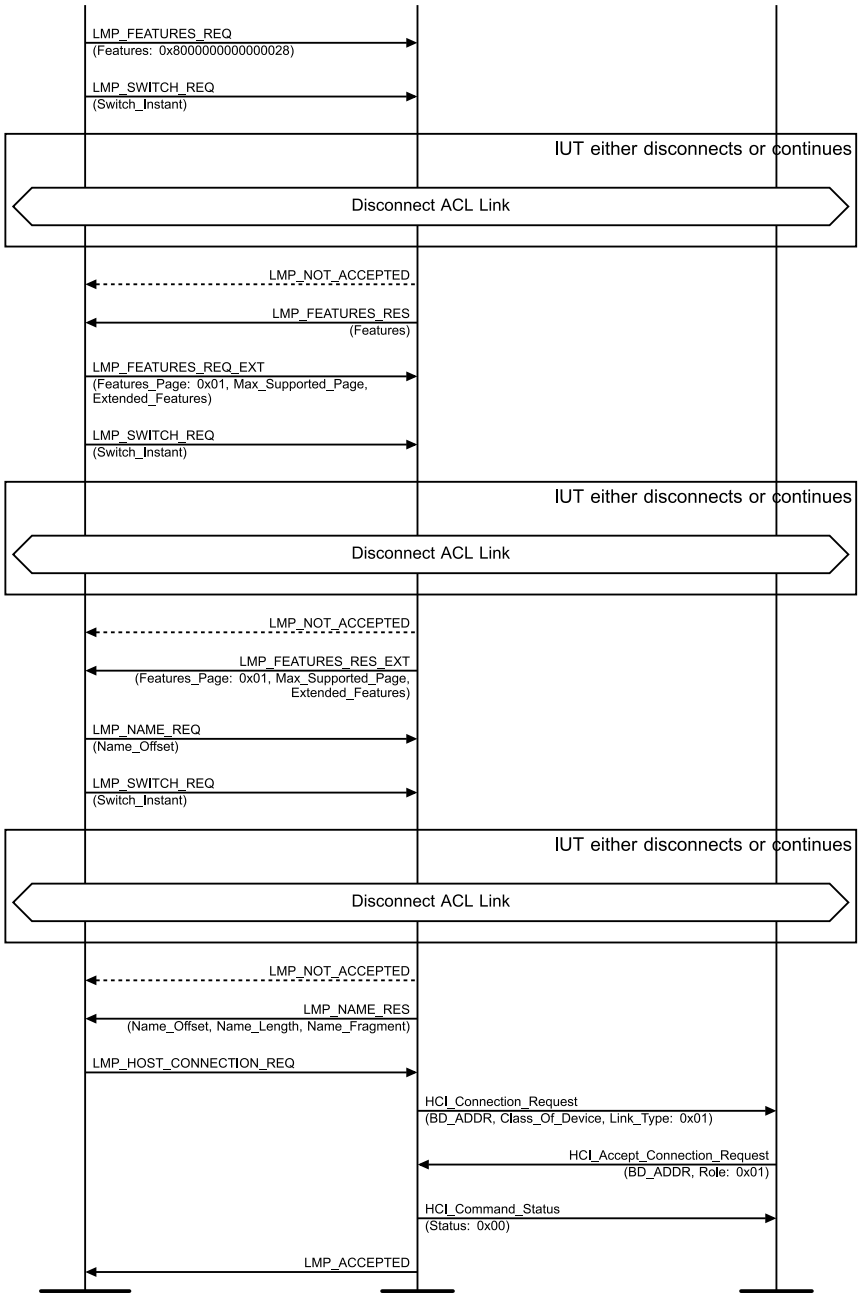


Figure 4.8-5: LMP/LIH/BI-04-C [Reject Role Switch at Setup, Peripheral] MSC – Page 2 of 2

1. If the IUT supports Inquiry, the Lower Tester sends an Inquiry packet to the IUT and the IUT responds to the Lower Tester with an FHS packet.
2. The Lower Tester sends a Page packet to the IUT.
3. The IUT responds to the Lower Tester with a Page Response packet.
4. The Lower Tester sends an FHS packet to the IUT.
5. The IUT responds to the Lower Tester with a Page Response packet.
6. The Lower Tester sends a POLL packet to the IUT.
7. The IUT responds with any packet to the Lower Tester.
8. The Lower Tester sends an LMP_CLKOFFSET_REQ PDU and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
9. If the IUT disconnects, the test has concluded; otherwise, connection setup continues.

10. Optionally, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
11. The IUT sends an LMP_CLKOFFSET_RES PDU to the Lower Tester with the Clock_Offset.
12. The Lower Tester sends an LMP_VERSION_REQ PDU with the Version, Company_Identifier, and Subversion and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
13. If the IUT disconnects, the test has concluded; otherwise, connection setup continues.
14. Optionally, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
15. The IUT sends an LMP_VERSION_RES PDU to the Lower Tester with the Version, Company_Identifier, and Subversion.
16. The Lower Tester sends an LMP_FEATURES_REQ PDU with Features set to 0x8000000000000028 (indicating support for extended features, role switch, and slot offset) and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
17. If the IUT disconnects, the test has concluded; otherwise, connection setup continues.
18. Optionally, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
19. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester with the Features parameter.
20. The Lower Tester sends an LMP_FEATURES_REQ_EXT PDU with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
21. If the IUT disconnects, the test has concluded; otherwise, connection setup continues.
22. Optionally, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
23. The IUT sends an LMP_FEATURES_RES_EXT PDU to the IUT with Features_Page set to 0x01, Max_Supported_Page, and Extended_Features.
24. The Lower Tester sends an LMP_NAME_REQ with Name_Offset and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
25. If the IUT disconnects, the test has concluded; otherwise, connection setup continues.
26. Optionally, the IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
27. The IUT sends an LMP_NAME_RES PDU to the Lower Tester with Name_Offset, Name_Length, and Name_Fragment.
28. The Lower Tester sends an LMP_HOST_CONNECTION_REQ PDU to the IUT.
29. The IUT sends an HCI_Connection_Request event to the Upper Tester with the BD_ADDR, Class_Of_Device, and Link_Type set to 0x01 (ACL).
30. The Upper Tester responds with an HCI_Accept_Connection_Request with the BD_ADDR and Role set to 0x01 (Peripheral) and receives a successful HCI_Command_Status event in response.
31. The IUT sends an LMP_ACCEPTED PDU with the LMP_HOST_CONNECTION_REQ PDU Opcode.

- Expected Outcome

  Pass verdict

  After the Lower Tester requests a role switch, the IUT either disconnects the ACL Link and the test ends with a successful verdict, or the IUT continues with connection setup.

  If the IUT continues with connection setup, the IUT optionally sends an LMP_NOT_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester.

  If the IUT continues with connection setup, the IUT sends an HCI_Connection_Request event to the Upper Tester.

4.8.3 Role Switch – Central

Verify that an IUT can request a role switch. The IUT is the Central.

LMP/LIH/BV-02-C [Accept Role Switch]

- Test Purpose

  Verify that the IUT accepts the Lower Tester's request to switch roles from Peripheral to Central and then Central to Peripheral.

- Reference

  [1] 4.4.2

- Initial Condition
  - See the "Default settings" section.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT must page the Lower Tester to become the Central of the piconet.

- Test Procedure

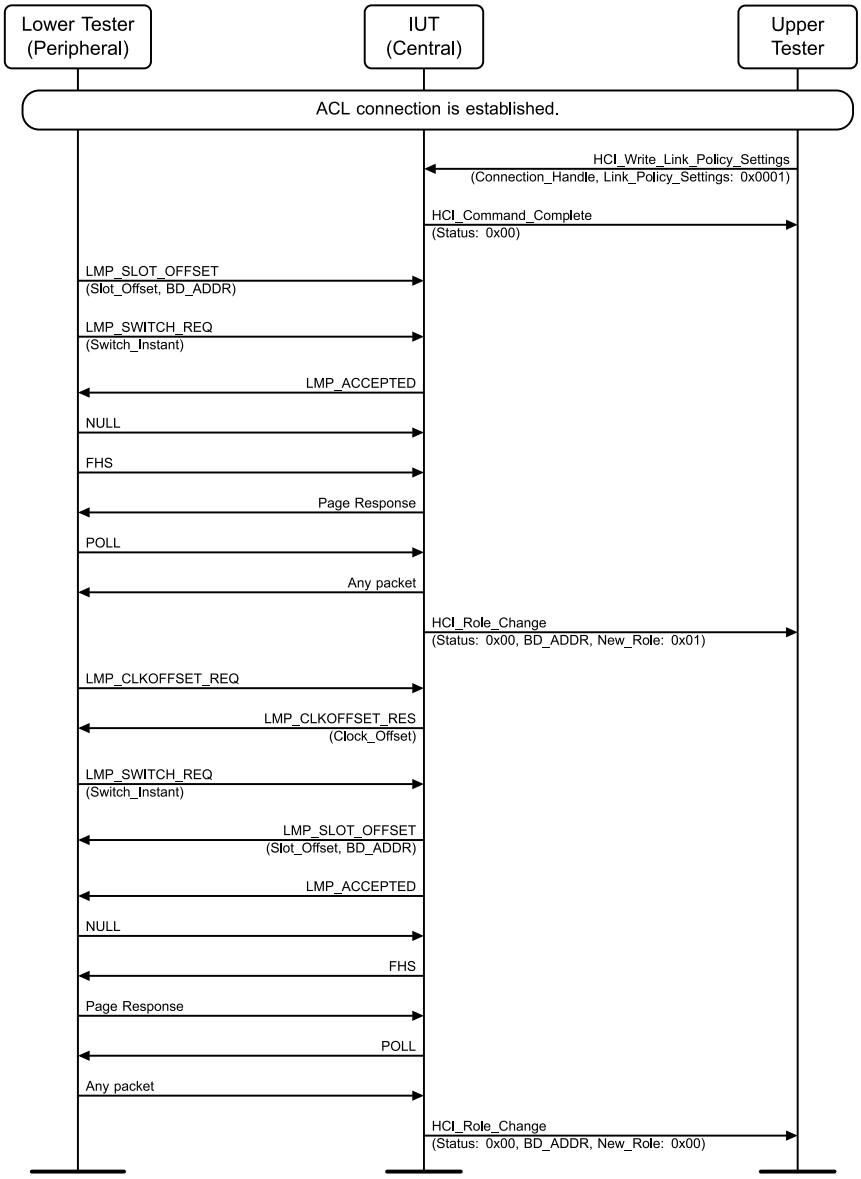


Figure 4.8-6: LMP/LIH/BV-02-C [Accept Role Switch] MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0001 (Enable Role Switch) and receives a successful HCI_Command_Complete event in response.
2. The Lower Tester sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
3. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.
4. The Lower Tester sends a NULL packet to the IUT.
5. The Lower Tester sends an FHS packet to the IUT.
6. The IUT responds to the Lower Tester with a Page Response packet.
7. The Lower Tester sends a POLL packet to the IUT.
8. The IUT responds to the Lower Tester with any packet.
9. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x01 (Peripheral).

10. The Lower Tester sends an LMP_CLKOFFSET_REQ PDU to the IUT.
11. The IUT responds to the Lower Tester with an LMP_CLKOFFSET_RES PDU with the Clock_Offset.
12. The Lower Tester sends an LMP_SWITCH_REQ PDU to the IUT with the Switch_Instant.
13. The IUT responds to the Lower Tester with an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
14. The Lower Tester sends a NULL packet to the IUT.
15. The IUT sends an FHS packet to the Lower Tester.
16. The Lower Tester responds to the IUT with a Page Response packet.
17. The IUT sends a POLL packet to the Lower Tester.
18. The Lower Tester responds to the IUT with any packet.
19. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).

- Expected Outcome

  Pass verdict

  In Step 3, the IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester.

  In Step 9, the IUT sends the HCI_Role_Change event to the Upper Tester, which shows its new role as the Peripheral.

  In Step 13, the IUT sends the LMP_SLOT_OFFSET and LMP_ACCEPTED PDUs to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester.

  In Step 19, the IUT sends the HCI_Role_Change event to the Upper Tester, which shows its new role as the Central.

4.8.3.1 Role Switch at Setup, Central

- Test Purpose

  Verify that the IUT handles a role switch request correctly during connection setup.

- Reference

  [1] 4.1.1, 4.4.2

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Central and the Lower Tester is the Peripheral.

- Test Case Configuration

Test Case	Role Switch
LMP/LIH/BV-78-C [Role Switch at Setup, Central]	ALT 14A or ALT 14B
LMP/LIH/BV-151-C [Role Switch at Setup, Central]	ALT 14A

Table 4.8-1: Role Switch at Setup, Central test cases

- Test Procedure

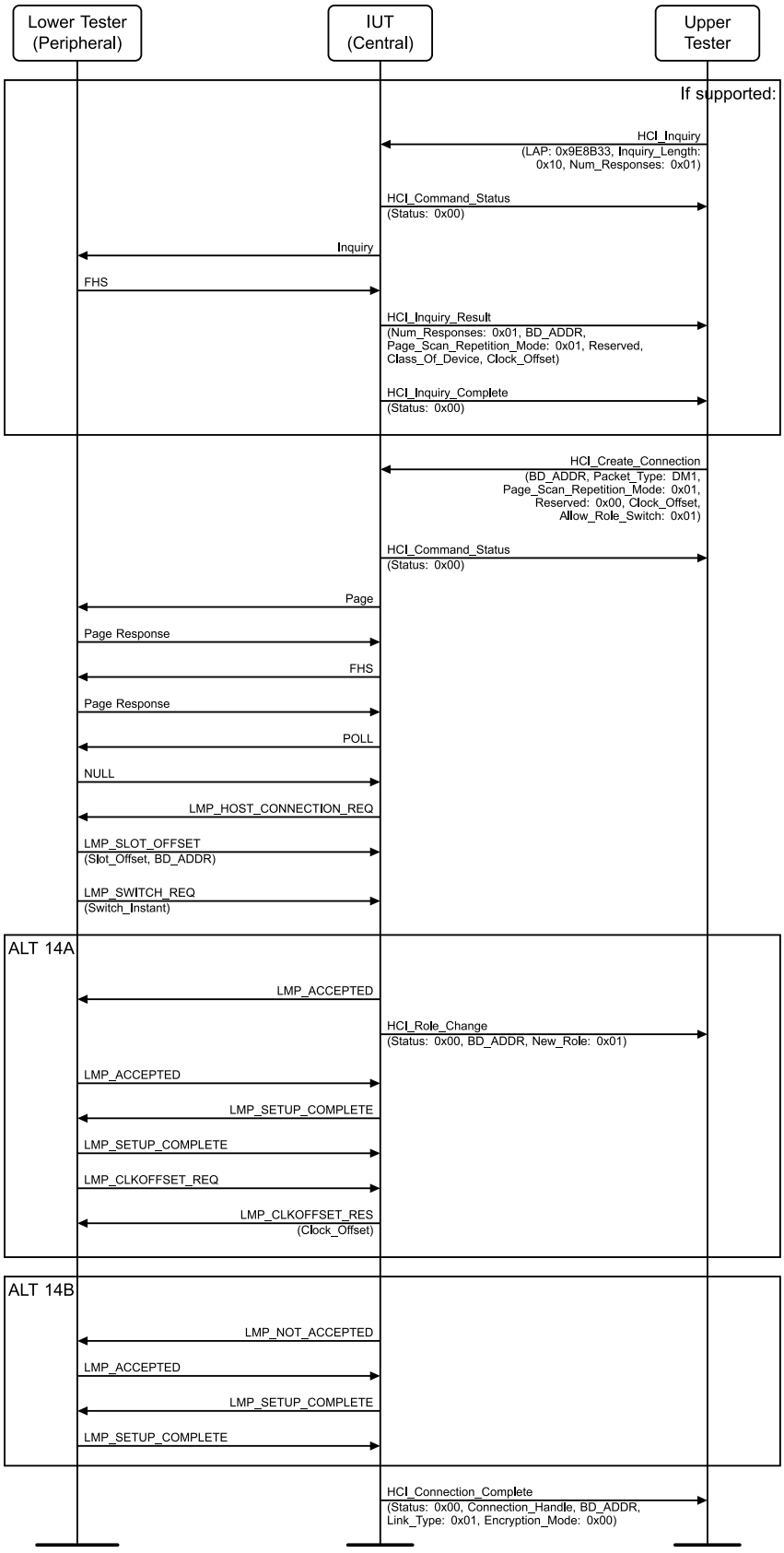


Figure 4.8-7: Role Switch at Setup, Central MSC

Execute Steps 1–4 if the IUT supports Inquiry; otherwise, start at Step 5:

1. The Upper Tester sends an HCI_Inquiry command to the IUT with LAP set to 0x9E8B33, Inquiry_Length set to 0x10, and Num_Responses set to 0x01 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an Inquiry packet to the Lower Tester.
3. The Lower Tester responds to the IUT with an FHS packet.
4. The IUT sends an HCI_Inquiry_Result event with Num_Responses set to 0x01, BD_ADDR, Page_Scan_Repetition_Mode set to 0x01 (R1), Reserved, Class_Of_Device, and Clock_Offset and a successful HCI_Inquiry_Complete event to the IUT.
5. The Upper Tester sends an HCI_Create_Connection command to the IUT with the BD_ADDR, Packet_Type set to DM1, Page_Scan_Repetition_Mode set to 0x01 (R1), Reserved set to 0x00, Clock_Offset, and Allow_Role_Switch set to 0x01 and receives a successful HCI_Command_Status event in response.
6. The IUT sends a Page packet to the Lower Tester.
7. The Lower Tester responds with a Page Response packet to the IUT.
8. The IUT sends an FHS packet to the Lower Tester.
9. The Lower Tester responds to the IUT with a Page Response packet.
10. The IUT sends a POLL packet to the Lower Tester.
11. The Lower Tester responds to the IUT with a NULL packet.
12. The IUT sends an LMP_HOST_CONNECTION_REQ PDU to the Lower Tester.
13. The Lower Tester sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
14. Perform either alternative 14A or 14B depending on the IUT's response.

    Alternative 14A (The IUT accepts the Lower Tester's role switch request):

    14A.1. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.

    14A.2. (Step 14A.2 can be sent before or after Step 14A.3.) The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x01 (Peripheral).

    14A.3. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_HOST_CONNECTION_REQ PDU Opcode.

    14A.4. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.

    14A.5. The Lower Tester responds to the IUT with an LMP_SETUP_COMPLETE PDU.

    14A.6. The Lower Tester sends an LMP_CLKOFFSET_REQ PDU to the IUT.

    14A.7. The IUT responds to the Lower Tester with an LMP_CLKOFFSET_RES PDU with the Clock_Offset.

    Alternative 14B (The IUT rejects the Lower Tester's role switch request):

    14B.1. The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_SWITCH_REQ PDU Opcode.

    14B.2. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_HOST_CONNECTION_REQ PDU Opcode.

    14B.3. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.

    14B.4. The Lower Tester responds to the IUT with an LMP_SETUP_COMPLETE PDU.
15. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).

- Expected Outcome

  Pass verdict

  In alternative 14A, the IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester and sends the LMP_CLKOFFSET_RES PDU to the Lower Tester upon reception of the LMP_CLKOFFSET_REQ PDU from the Lower Tester.

  In alternative 14B, the IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester and exchanges LMP_SETUP_COMPLETE PDUs with the Lower Tester.

  In Step 15, the IUT sends the HCI_Connection_Complete event to the Upper Tester.

LMP/LIH/BV-142-C [Reject Role Switch Request, Central]

- Test Purpose

  Verify that the IUT rejects the Lower Tester's requests to switch roles from Peripheral to Central. The IUT is Central. The Lower Tester is Peripheral and initiates the service.

- Reference

  [1] 4.4.2

- Initial Condition

  - See the "Default settings" section.
  - An ACL connection has been established.
  - The IUT is the Central and the Lower Tester is the Peripheral.
  - The IUT must page the Lower Tester to become the Central of the piconet.

- Test Procedure

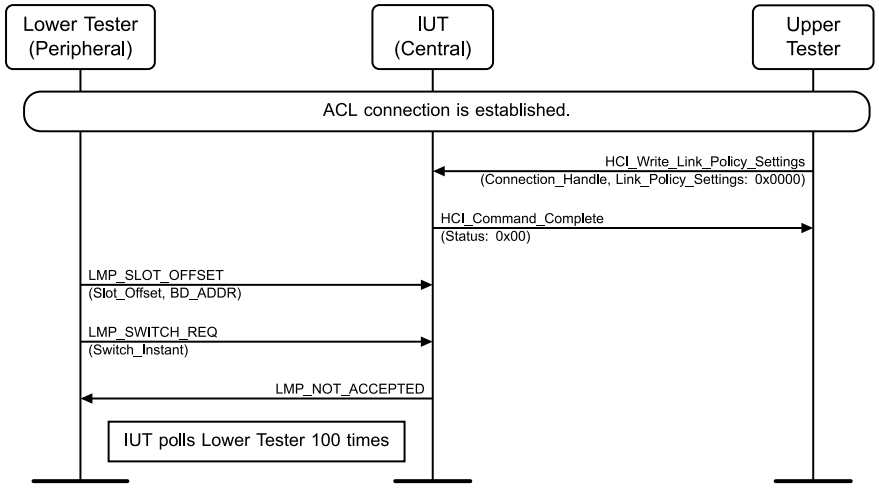


Figure 4.8-8: LMP/LIH/BV-142-C [Reject Role Switch Request] MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0000 and receives a successful HCI_Command_Complete event in response.
2. The Lower Tester sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.

3. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
4. The IUT sends 100 POLL packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT rejects the role switch and remains the Central.

  In Step 4, the IUT continues to poll the Lower Tester 100 times.

4.8.4 Role Switch – Both Central and Peripheral

Verify that the IUT declines the role switch in a correct manner. The role of the IUT is of no importance.

4.8.4.1 Rejected Role Switch Request

- Test Purpose

  Verify that the IUT properly handles the Lower Tester rejecting the role switch request. Verify that the IUT properly handles a role switch when the requested role is the same as the current role.

- Reference

  [1] 4.4.2

  [7] 7.2.8, 7.7.18

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	IUT Role	Role for HCI_Switch_Role
LMP/LIH/BV-143-C [Rejected Role Switch Request, Peripheral]	Peripheral	Central
LMP/LIH/BV-149-C [Rejected Role Switch Request, Central]	Central	Peripheral

Table 4.8-2: Rejected Role Switch Request

- Test Procedure

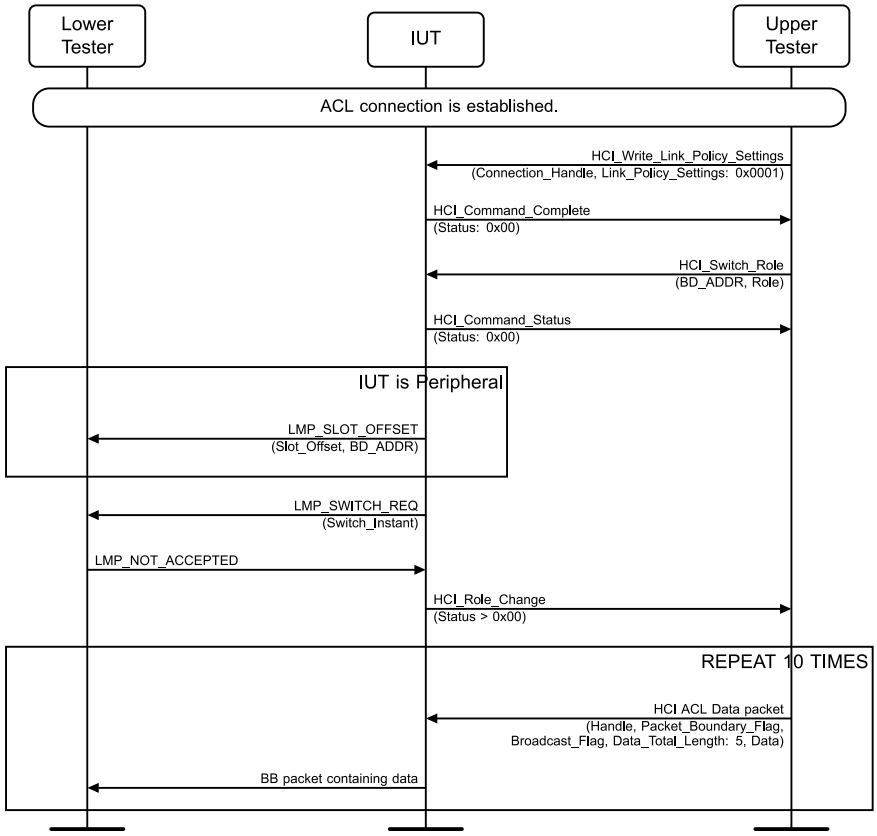


Figure 4.8-9: Rejected Role Switch Request MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0001 (Enable Role Switch) and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to first Role for HCI_Switch_Role in Table 4.8-2 and receives a successful HCI_Command_Status event in response.
3. If the IUT Role is Peripheral in Table 4.8-2, the IUT sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and the BD_ADDR.
4. The IUT sends an LMP_SWITCH_REQ PDU to the Lower Tester with the Switch_Instant.
5. The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT with the LMP_SWITCH_REQ PDU Opcode.
6. The IUT sends an HCI_Role_Change with Status > 0x00, BD_ADDR, and New_Role.
7. The Upper Tester sends an HCI ACL Data packet to the IUT.
8. The IUT sends BB packets containing data to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_SWITCH_REQ PDU.

  In Step 8, the IUT successfully sends data packets after the role switch request was rejected.

LMP/LIH/BV-03-C [Unsupported Role Switch]

- Test Purpose

  Verify that the IUT responds that it does not support role switch upon request from the Lower Tester and rejects a role switch upon request from the Upper Tester.

- Reference

  [1] 4.4.2

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

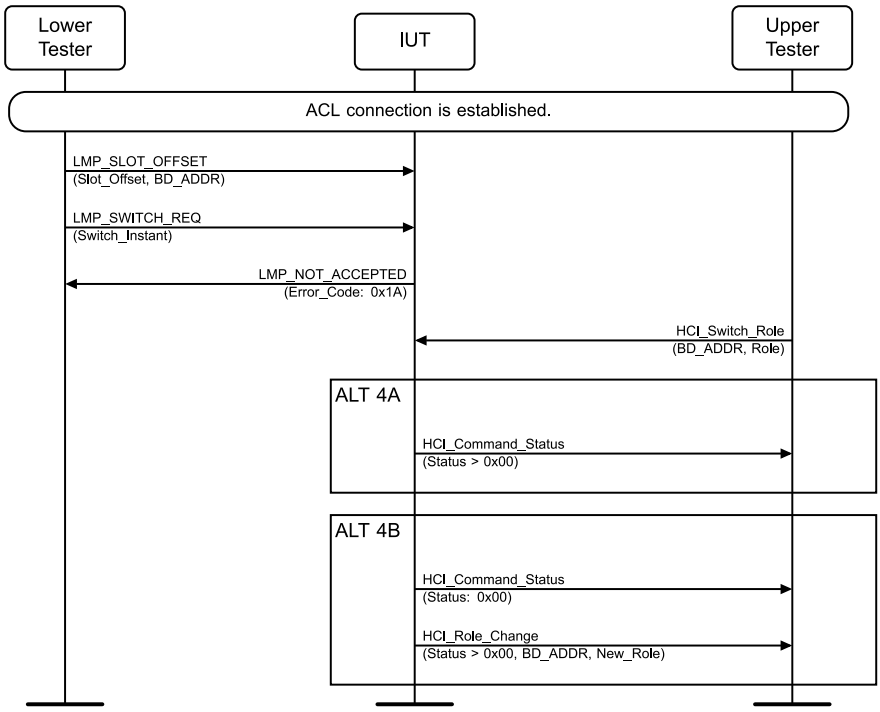


Figure 4.8-10: LMP/LIH/BV-03-C [Unsupported Role Switch] MSC

1. The Lower Tester sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode and Error_Code set to 0x1A (Unsupported LMP Feature).
3. The Upper Tester sends an HCI_Switch_Role with the BD_ADDR and Role set to the opposite role the IUT is currently in.
4. Perform either alternative 4A or 4B depending on the IUT’s response.
   Alternative 4A (The IUT rejects the HCI_Switch_Role command):
   - 4A.1. The IUT sends an HCI_Command_Status event to the Upper Tester with Status set to a valid error code.

   Alternative 4B (The IUT accepts the HCI_Switch_Role command):
   - 4B.1. The IUT sends a successful HCI_Command_Status event to the Upper Tester.
   - 4B.2. The IUT sends an HCI_Role_Change event to the Upper Tester with the Status set to a valid error code, BD_ADDR, and New_Role.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1A (Unsupported LMP Feature) upon reception of the LMP_SWITCH_REQ PDU from the Lower Tester.

  The IUT rejects the HCI_Switch_Role command with any valid error code.

4.8.5 Detach – Both Central and Peripheral

Verify that the connection between two Bluetooth devices can be closed at any time by the Central or the Peripheral. The role of the IUT is of no importance.

LMP/LIH/BV-04-C [Close Link on Request]

- Test Purpose

  Verify that the IUT closes the link upon request from the Lower Tester.

- Reference

  [1] 4.1.2

- Initial Condition

  - See the “Default settings” section.
  - The Lower Tester is the Initiator.
  - An ACL connection has been established.

- Test Procedure

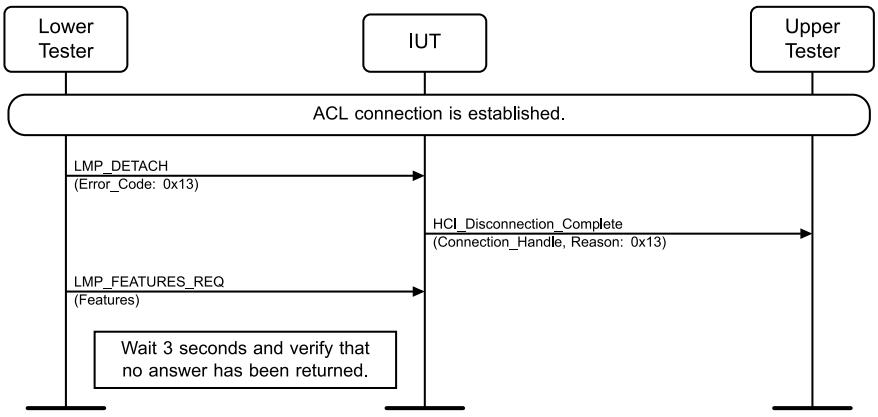


Figure 4.8-11: LMP/LIH/BV-04-C [Close Link on Request] MSC

1. The Lower Tester sends an LMP_DETACH PDU to the IUT with Error_Code 0x13 (Remote User Terminated Connection).
2. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection).
3. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT and waits 3 seconds to ensure that the IUT does not respond to the request.

- Expected Outcome

  Pass verdict

  Both the LM and BB links close down after the reception of the LMP_DETACH PDU from the Lower Tester.

LMP/LIH/BV-05-C [Close Link, HCI Command]

- Test Purpose

  Verify that the IUT can close the link.

- Reference

  [1] 4.1.2

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

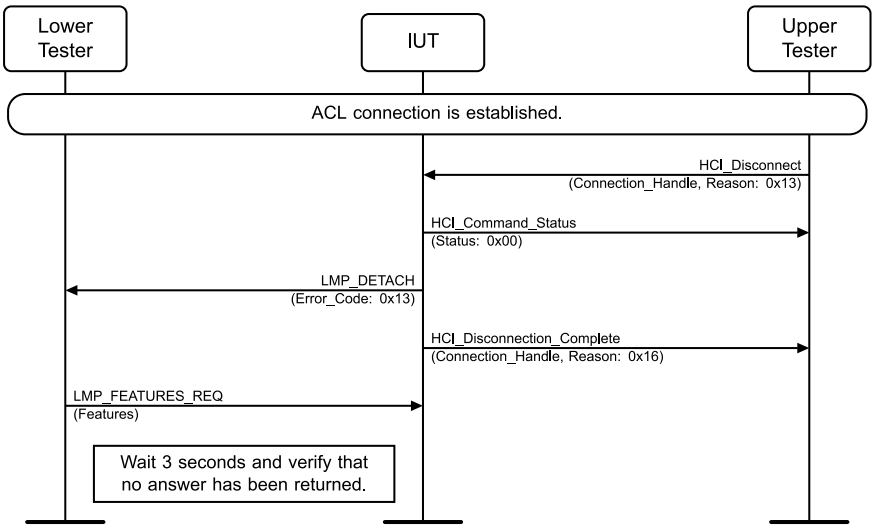


Figure 4.8-12: LMP/LIH/BV-05-C [Close Link, HCI Command] MSC

1. The Upper Tester sends an HCI_Disconnect command to the IUT with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection) and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x13 (Remote User Terminated Connection).
3. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x16 (Connection Terminated By Local Host).
4. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT and waits 3 seconds to ensure that the IUT does not respond to the request.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_DETACH PDU to the Lower Tester.

  Both BB and LM links close down after the IUT sends the HCI_Disconnection_Complete event to the Upper Tester.

LMP/LIH/BV-82-C [Setup Rejected]

- Test Purpose

  Verify that the IUT accepts that the Lower Tester rejects the connection setup.

  Verify that the IUT closes the link correctly.

  The IUT is the Central and requests an ACL link.

- Reference

  [1] 4.1.1

- Initial Condition

  - See Figure 4.8-13.

- Test Procedure

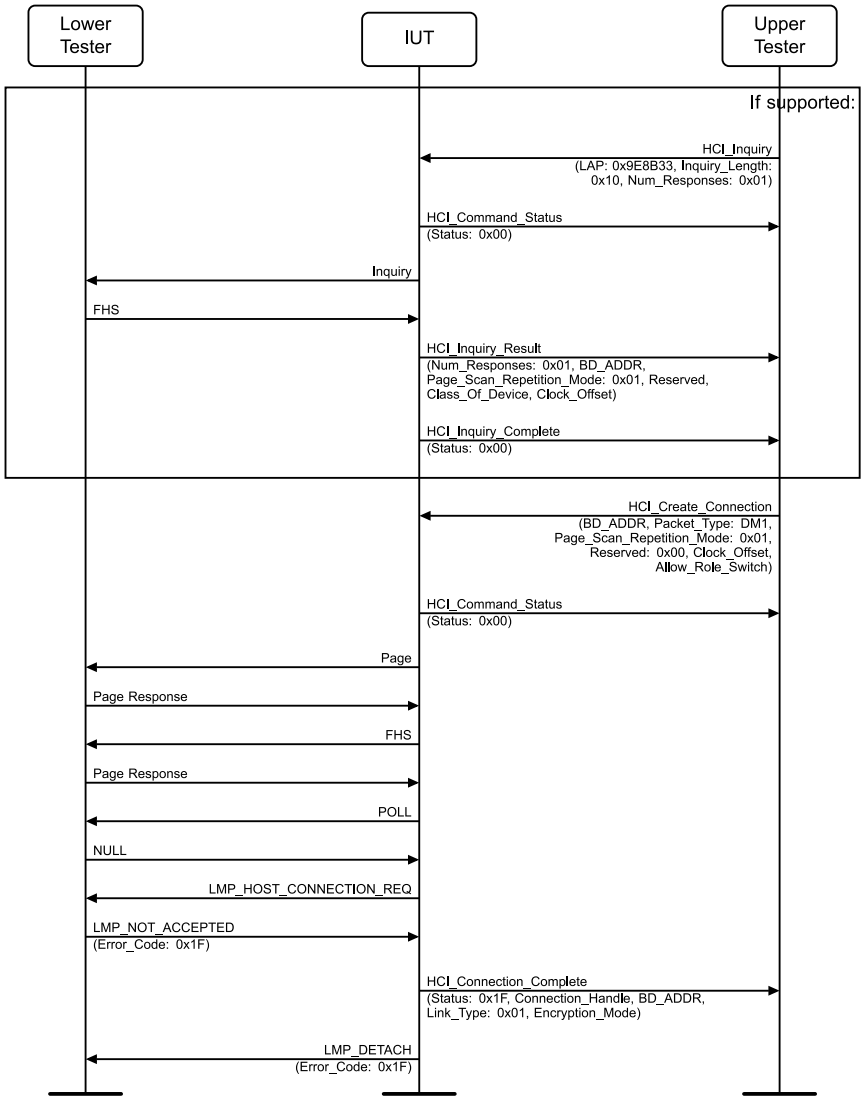


Figure 4.8-13: LMP/LIH/BV-82-C [Setup Rejected] MSC

Execute Steps 1–4 if the IUT supports Inquiry; otherwise, start at Step 5.

1. The Upper Tester sends an HCI_Inquiry command to the IUT with LAP set to 0x9E8B33, Inquiry_Length set to 0x10, and Num_Responses set to 0x01 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an Inquiry packet to the Lower Tester.
3. The Lower Tester responds to the IUT with an FHS packet.
4. The IUT sends an HCI_Inquiry_Result event with Num_Responses set to 0x01, BD_ADDR, Page_Scan_Repetition_Mode set to 0x01 (R1), Reserved, Class_Of_Device, and Clock_Offset and a successful HCI_Inquiry_Complete event to the IUT.
5. The Upper Tester sends an HCI_Create_Connection command to the IUT with the BD_ADDR, Packet_Type set to DM1, Page_Scan_Repetition_Mode set to 0x01 (R1), Reserved set to 0x00, Clock_Offset, and Allow_Role_Switch and receives a successful HCI_Command_Status event in response.
6. The IUT sends a Page packet to the Lower Tester.
7. The Lower Tester responds to the IUT with a Page Response packet.

8. The IUT sends an FHS packet to the Lower Tester.
9. The Lower Tester responds to the IUT with a Page Response packet.
10. The IUT sends a POLL packet to the Lower Tester.
11. The Lower Tester responds to the IUT with a NULL packet.
12. The IUT sends an LMP_HOST_CONNECTION_REQ PDU to the Lower Tester.
13. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_HOST_CONNECTION_REQ Opcode and Error_Code set to 0x1F (Unspecified Error).
14. The IUT sends an HCI_Connection_Complete event to the Upper Tester with Status set to 0x1F (Unspecified Error), Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode.
15. The IUT sends an LMP_DETACH PDU to the Lower Tester with Error_Code set to 0x1F (Unspecified Error).

- Expected Outcome

  Pass verdict

  In Step 15, the IUT sends the LMP_DETACH PDU to the Lower Tester upon reception of the LMP_NOT_ACCEPTED PDU from the Lower Tester.

4.8.6 Hold mode – Peripheral

Verify that the ACL connection between two Bluetooth devices can be placed in hold mode for a specified Hold_Time. The IUT is the Peripheral.

LMP/LIH/BV-06-C [Hold Mode, Peripheral]

- Test Purpose

  Verify that the IUT enters and exits Hold Mode after the Hold interval, first upon request from the Lower Tester and then upon force from the Lower Tester. Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.1.1

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service first by requesting and then by force.
  - The IUT’s minimum acceptable hold interval is defined by TSPX_hold_mode_min_interval, and the IUT’s maximum acceptable hold interval is defined by TSPX_hold_mode_max_interval.

- Test Procedure

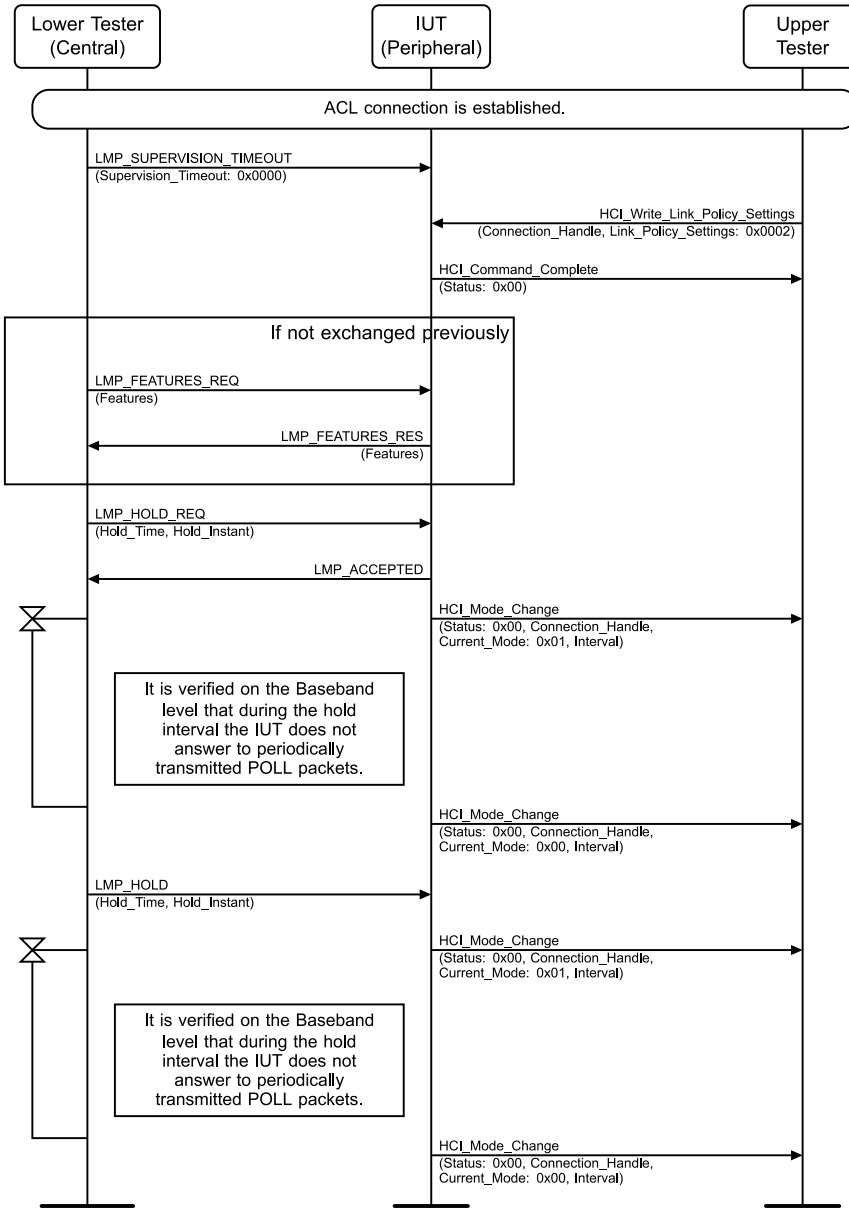


Figure 4.8-14: LMP/LIH/BV-06-C [Hold Mode, Peripheral] MSC

1. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU to the IUT with the Supervision_Timeout set to 0x0000.
2. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0002 (Hold Mode) and receives a successful HCI_Command_Complete event in response.
3. If the IUT and the Lower Tester have not exchanged LMP Features, the Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter and receives an LMP_FEATURES_RES PDU with the Features parameter in response.
4. The Lower Tester sends an LMP_HOLD_REQ PDU to the IUT with Hold_Time set between TSPX_hold_mode_min_interval and TSPX_hold_mode_max_interval and Hold_Instant.
5. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode.

6. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
7. While the hold interval is active, the Lower Tester sends POLL packets periodically to the IUT.
8. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
9. The Lower Tester sends an LMP_HOLD PDU to the IUT with Hold_Time set between TSPX_hold_mode_min_interval and TSPX_hold_mode_max_interval and Hold_Instant.
10. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
11. While the hold interval is active, the Lower Tester sends POLL packets periodically to the IUT.
12. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.

- Expected Outcome

  Pass verdict

  In Step 5, the IUT accepts the LMP_HOLD_REQ PDU sent by the Lower Tester by sending the LMP_ACCEPTED PDU.

  After Step 9, the IUT accepts the LMP_HOLD PDU sent by the Lower Tester.

  In Steps 7 and 11, the IUT does not respond to POLL packets sent by the Lower Tester during the hold interval.

- Notes

  The hold interval has an even value. The Hold_Instant is at an even slot.

LMP/LIH/BV-09-C [Hold Mode Request, Peripheral]

- Test Purpose

  Verify that the IUT can request or force the Lower Tester to enter Hold Mode during the Hold interval. Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.1

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Peripheral and initiates the service by requesting or forcing. The Lower Tester is the Central.
  - The IUT’s minimum acceptable hold interval is defined by TSPX_hold_mode_min_interval, and the IUT’s maximum acceptable hold interval is defined by TSPX_hold_mode_max_interval.

- Test Procedure

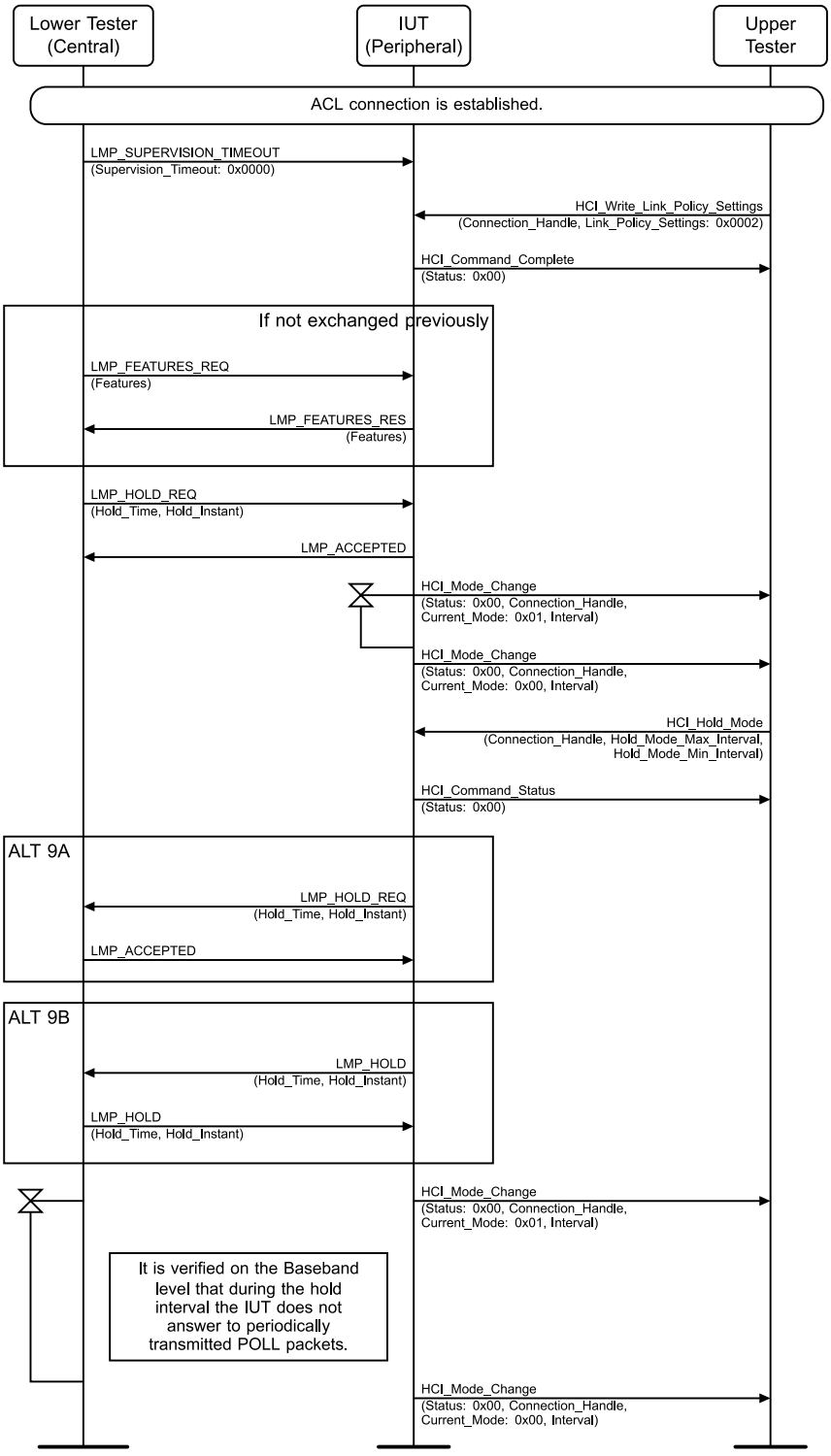


Figure 4.8-15: LMP/LIH/BV-09-C [Hold Mode Request, Peripheral] MSC

1. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU to the IUT with the Supervision_Timeout set to 0x0000.
2. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0002 (Hold Mode) and receives a successful HCI_Command_Complete event in response.

3. If the IUT and the Lower Tester have not exchanged LMP Features, the Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter and receives an LMP_FEATURES_RES PDU with the Features parameter in response.
4. The Lower Tester sends an LMP_HOLD_REQ PDU to the IUT with Hold_Time set between TSPX_hold_mode_min_interval and TSPX_hold_mode_max_interval and Hold_Instant.
5. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode.
6. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
7. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
8. The Upper Tester sends an HCI_Hold_Mode command to the IUT with the Connection_Handle, Hold_Mode_Max_Interval, and Hold_Mode_Min_Interval and receives a successful HCI_Command_Status event in response.
9. Perform either alternative 9A or 9B depending on the IUT's response.
   Alternative 9A (The IUT sends the LMP_HOLD_REQ PDU to the Lower Tester):
   - 9A.1. The IUT sends an LMP_HOLD_REQ PDU to the IUT with the Hold_Time and Hold_Instant.
   - 9A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode.

   Alternative 9B (The IUT sends the LMP_HOLD PDU to the Lower Tester):
   - 9B.1. The IUT sends an LMP_HOLD PDU to the IUT with the Hold_Time and Hold_Instant.
   - 9B.2. The Lower Tester responds to the IUT with an LMP_HOLD PDU with the Hold_Time and Hold_Instant.
10. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
11. While the hold interval is active, the Lower Tester sends POLL packets periodically to the IUT.
12. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.

- Expected Outcome

  Pass verdict

  In alternative 9A, the IUT sends the LMP_HOLD_REQ PDU to the Lower Tester.

  In alternative 9B, the IUT sends the LMP_HOLD PDU to the Lower Tester.

  In Step 11, the IUT does not respond to POLL packets during the hold interval.

- Notes

  There is no special HCI command for the LMP_HOLD PDU, as it is the same as for the LMP_HOLD_REQ PDU. It is therefore not possible to know if the IUT is going to force or request the Lower Tester to go into HOLD mode.

  The hold interval has an even value. The Hold_Instant is at an even slot. Hold_Mode_Min_Interval and Hold_Mode_Max_Interval sent from the Upper Tester will have the same value to exactly define the hold interval.

4.8.7 Hold mode – Central

Verify that the ACL connection between two Bluetooth devices can be placed in hold mode for a specified Hold_Time. The IUT is the Central.

LMP/LIH/BV-10-C [Hold Mode, Central]

- Test Purpose

  Verify that the IUT can request or force the ACL link into Hold mode after a previous successful request. Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.1

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Central and initiates the service. The Lower Tester is the Peripheral.
  - The IUT must page the Lower Tester to become the Central of the piconet.
  - The IUT’s minimum acceptable hold interval is defined by TSPX_hold_mode_min_interval, and the IUT’s maximum acceptable hold interval is defined by TSPX_hold_mode_max_interval.

- Test Procedure

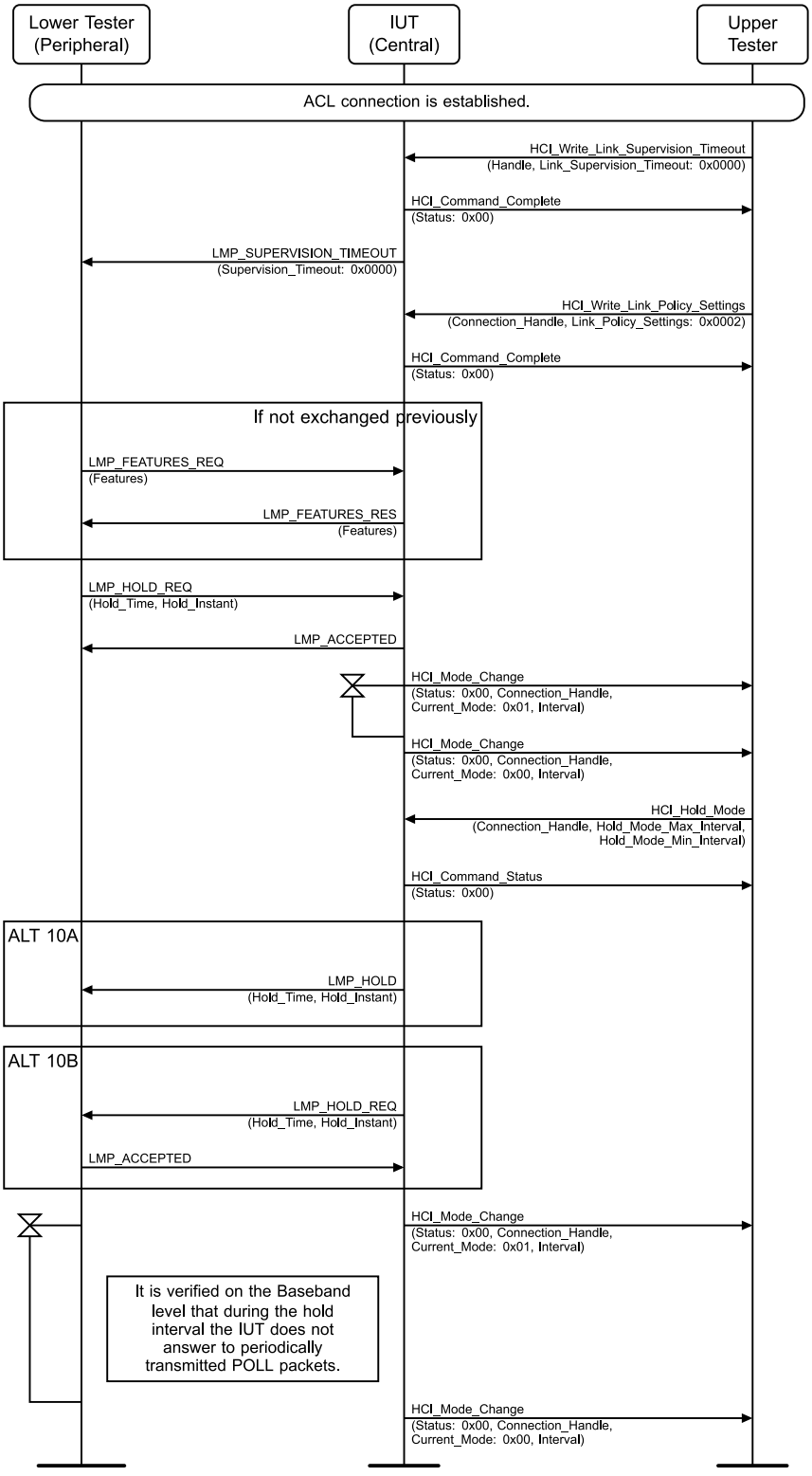


Figure 4.8-16: LMP/LIH/BV-10-C [Hold Mode, Central] MSC

1. The Upper Tester sends an HCI_Write_Link_Supervision_Timeout command to the IUT with the Handle and Link_Supervision_Timeout set to 0x0000 and receives a successful HCI_Command_Complete event in response.
2. The IUT sends an LMP_SUPERVISION_TIMEOUT PDU to the Lower Tester with the Supervision_Timeout set to 0x0000.
3. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0002 (Hold Mode) and receives a successful HCI_Command_Complete event in response.
4. If the IUT and the Lower Tester have not exchanged LMP Features, the Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter and receives an LMP_FEATURES_RES PDU with the Features parameter in response.
5. The Lower Tester sends an LMP_HOLD_REQ PDU to the IUT with Hold_Time set between TSPX_hold_mode_min_interval and TSPX_hold_mode_max_interval and Hold_Instant.
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode.
7. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
8. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
9. The Upper Tester sends an HCI_Hold_Mode command to the IUT with the Connection_Handle, Hold_Mode_Max_Interval, and Hold_Mode_Min_Interval and receives a successful HCI_Command_Status event in response.
10. Perform either alternative 10A or 10B depending on the IUT's response.
    Alternative 10A (The IUT sends the LMP_HOLD PDU to the Lower Tester):
        10A.1. The IUT sends an LMP_HOLD PDU to the IUT with the Hold_Time and Hold_Instant.
    Alternative 10B (The IUT sends the LMP_HOLD_REQ PDU to the Lower Tester):
        10B.1. The IUT sends an LMP_HOLD_REQ PDU to the IUT with the Hold_Time and Hold_Instant.
        10B.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode.
11. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
12. While the hold interval is active, the Lower Tester sends POLL packets periodically to the IUT.
13. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.

- Expected Outcome

  Pass verdict

  In Step 10, the IUT sends the LMP_HOLD or LMP_HOLD_REQ PDU to the Lower Tester.

  In Step 12, the IUT does not address the Lower Tester during the Hold interval.

- Notes

  There is no special HCI command for the LMP_HOLD PDU, as it is the same as for the LMP_HOLD_REQ PDU. It is therefore not possible to know if the IUT is going to force or request the Lower Tester to go into HOLD mode.

  The hold interval has an even value. The Hold_Instant is at an even slot. Hold_Mode_Min_Interval and Hold_Mode_Max_Interval sent from the Upper Tester will have the same value to exactly define the hold interval.

LMP/LIH/BV-11-C [Hold Mode Request, Central]

- Test Purpose

  Verify that the IUT accepts that the Lower Tester forces Hold mode. Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.1.2

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Central. The Lower Tester is the Peripheral and initiates the service by force.
  - The IUT must page the Lower Tester to become the Central of the piconet.
  - The IUT’s minimum acceptable hold interval is defined by TSPX_hold_mode_min_interval and the IUT’s maximum acceptable hold interval is defined by TSPX_hold_mode_max_interval.

- Test Procedure

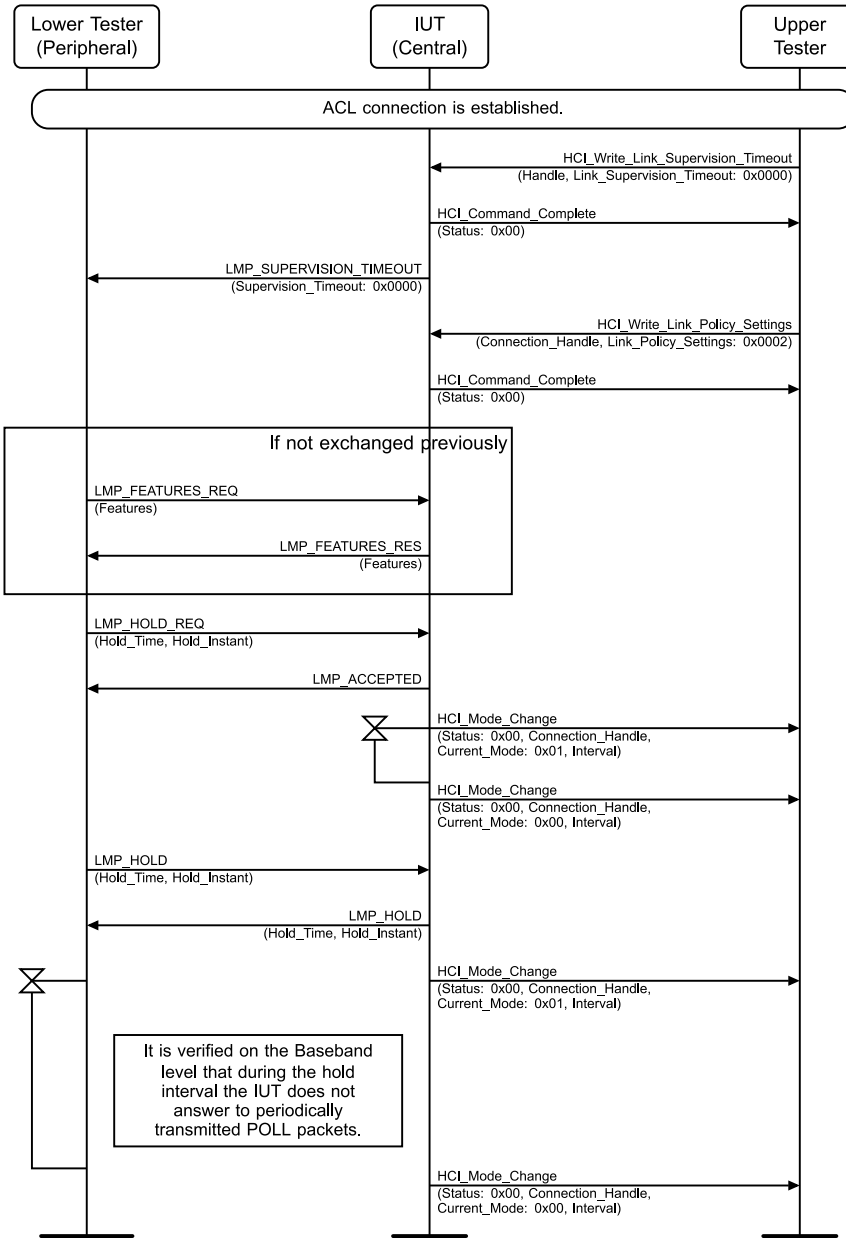


Figure 4.8-17: LMP/LIH/BV-11-C [Hold Mode Request, Central] MSC

1. The Upper Tester sends an HCI_Write_Link_Supervision_Timeout command to the IUT with the Handle and Link_Supervision_Timeout set to 0x0000 and receives a successful HCI_Command_Complete event in response.
2. The IUT sends an LMP_SUPERVISION_TIMEOUT PDU to the Lower Tester with the Supervision_Timeout set to 0x0000.
3. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0002 (Hold Mode) and receives a successful HCI_Command_Complete event in response.
4. If the IUT and the Lower Tester have not exchanged LMP Features, the Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter and receives an LMP_FEATURES_RES PDU with the Features parameter in response.

5. The Lower Tester sends an LMP_HOLD_REQ PDU to the IUT with Hold_Time set between TSPX_hold_mode_min_interval and TSPX_hold_mode_max_interval and Hold_Instant.
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode.
7. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
8. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
9. The Lower Tester sends an LMP_HOLD PDU to the IUT with Hold_Time set between TSPX_hold_mode_min_interval and TSPX_hold_mode_max_interval and Hold_Instant.
10. The IUT responds to the IUT with an LMP_HOLD PDU with the Hold_Time and Hold_Instant.
11. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval.
12. While the hold interval is active, the Lower Tester sends POLL packets periodically to the IUT.
13. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.

- Expected Outcome

  Pass verdict

  In Step 10, the IUT sends the LMP_HOLD PDU to the Lower Tester upon reception of the LMP_HOLD PDU from the Lower Tester.

  In Step 12, the IUT does not address the Lower Tester during the Hold interval.

- Notes

  The hold interval has an even value. The Hold_Instant is at an even slot.

4.8.8 Hold mode – Both Central and Peripheral

Verify that the IUT declines the Hold mode in a correct manner. The role of the IUT is of no importance.

LMP/LIH/BV-12-C [Hold Mode Unsupported]

- Test Purpose

  Verify that the IUT responds that it does not support Hold mode upon a Hold request from the Lower Tester.

- Reference

  [1] 4.5.1.3

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

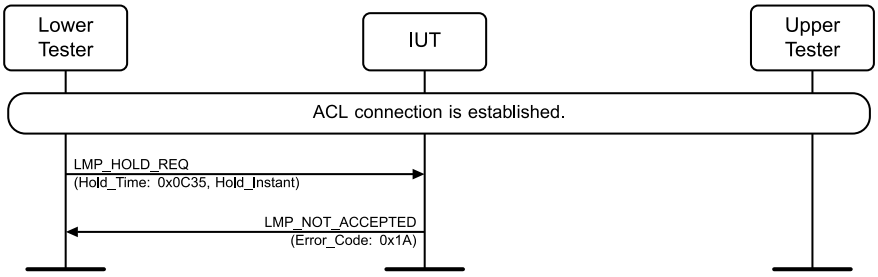


Figure 4.8-18: LMP/LIH/BV-12-C [Hold Mode Unsupported] MSC

1. The Lower Tester sends an LMP_HOLD_REQ PDU to the IUT with Hold_Time set 0x0C35 and the Hold_Instant.
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode and Error_Code set to 0x1A (Unsupported LMP Feature).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1A (Unsupported LMP Feature) upon reception of the LMP_HOLD_REQ PDU from the Lower Tester.

4.8.9 Sniff mode – Peripheral

Verify that the ACL connection between two Bluetooth devices can be placed in Sniff mode. The IUT is the Peripheral.

LMP/LIH/BV-14-C [Enter Sniff Mode]

- Test Purpose

  Verify that the IUT enters Sniff mode upon request from the Lower Tester, interprets the Sniff_Attempt and Sniff_Timeout correctly, and ignores timing control flags bits 0 and 2 in the LMP_SNIFF_REQ PDU sent by the Lower Tester.

  Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.3.1

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service by requesting.

- Test Procedure

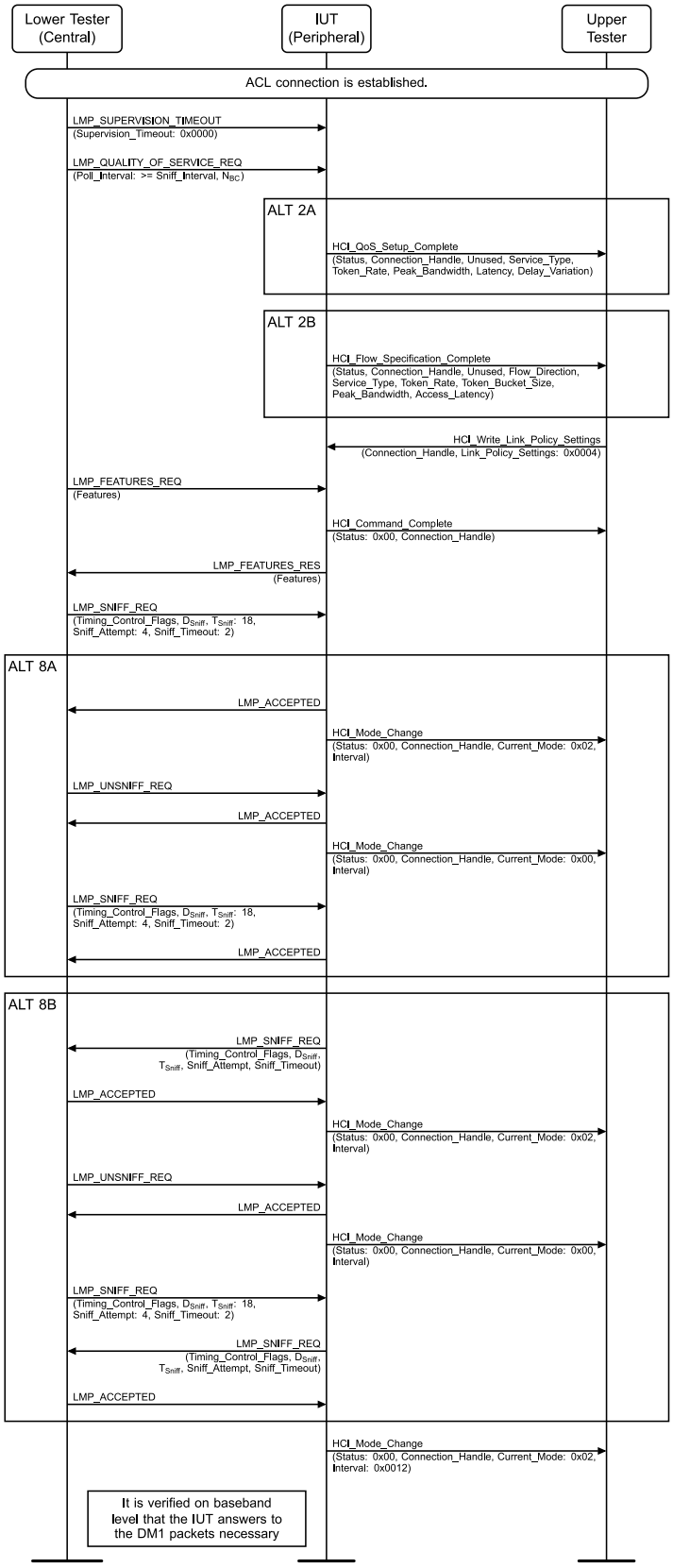


Figure 4.8-19: LMP/LIH/BV-14-C [Enter Sniff Mode] MSC

1. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU with the Supervision_Timeout set to 0x0000 and an LMP_QUALITY_OF_SERVICE_REQ PDU with the Poll_Interval greater than or equal to the Sniff_Interval and N_{BC}.
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   - 2A.1. The IUT sends an HCI_QoS_Setup_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

   Alternative 2B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):
   - 2B.1. The IUT sends an HCI_Flow_Specification_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.
3. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0004 (Sniff Mode).
4. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
5. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
6. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester with the Features parameter.
7. The Lower Tester sends an LMP_SNIFF_REQ PDU to the IUT with Timing_Control_Flags having bits 0 and 2 set to zero, D_{Sniff}, T_{Sniff} set to 18, Sniff_Attempt set to 4, and Sniff_Timeout set to 2.
8. Perform either alternative 8A or 8B depending on the IUT's response.
   Alternative 8A (The IUT sends the LMP_ACCEPTED PDU to the Lower Tester):
   - 8A.1. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
   - 8A.2. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval.
   - 8A.3. The Lower Tester sends an LMP_UNSNIFF_REQ PDU to the IUT.
   - 8A.4. The IUT responds with an LMP_ACCEPTED PDU to the Lower Tester with the LMP_UNSNIFF_REQ PDU Opcode.
   - 8A.5. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
   - 8A.6. The Lower Tester sends an LMP_SNIFF_REQ PDU to the IUT with Timing_Control_Flags having bits 0 and 2 set to one, D_{Sniff}, T_{Sniff} set to 18, Sniff_Attempt set to 4, and Sniff_Timeout set to 2.
   - 8A.7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.

   Alternative 8B (The IUT sends the LMP_SNIFF_REQ PDU to the Lower Tester):
   - 8B.1. The IUT responds to the Lower Tester with an LMP_SNIFF_REQ PDU with Timing_Control_Flags, D_{Sniff}, T_{Sniff}, Sniff_Attempt, and Sniff_Timeout.
   - 8B.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
   - 8B.3. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval.
   - 8B.4. The Lower Tester sends an LMP_UNSNIFF_REQ PDU to the IUT.
   - 8B.5. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_UNSNIFF_REQ PDU Opcode.
   - 8B.6. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
   - 8B.7. The Lower Tester sends an LMP_SNIFF_REQ PDU to the IUT with Timing_Control_Flags having bits 0 and 2 set to one, D_{Sniff}, T_{Sniff} set to 18, Sniff_Attempt set to 4, and Sniff_Timeout set to 2.

8B.8. The IUT responds to the Lower Tester with an LMP_SNIFF_REQ PDU with Timing_Control_Flags, D_{Sniff}, T_{Sniff}, Sniff_Attempt, and Sniff_Timeout.

8B.9. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.

9. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval set to 0x0012.
10. The Lower Tester sends POLL packets to the IUT according to Figure 4.8-20.

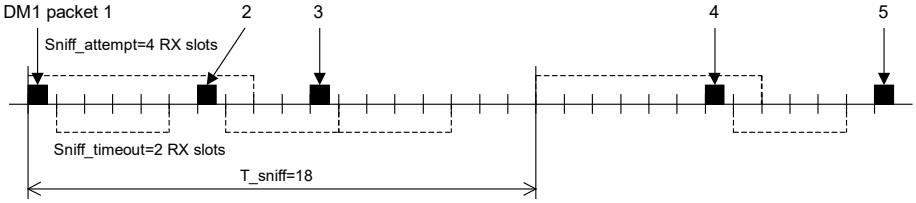


Figure 4.8-20: LMP/LIH/BV-14-C, Polling

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU or LMP_SNIFF_REQ PDU to the Lower Tester upon reception of the LMP_SNIFF_REQ PDU from the Lower Tester.

  The IUT enters Sniff mode and acknowledges DM1 packets 1–4 and does not acknowledge DM1 packet 5 for a period of 20*T_{Sniff} slots.

- Notes

  Timing_Control_Flags and D_{Sniff} are determined by CLK_{27} of the Central.

LMP/LIH/BV-15-C [Initiate Sniff Mode, Peripheral]

- Test Purpose

  Verify that the IUT can request the Lower Tester to enter Sniff mode and that the IUT interprets the Sniff_Attempt and Sniff_Timeout correctly.

  Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.3.1

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Peripheral and initiates the service by requesting. The Lower Tester is the Central and accepts the first request.

- Test Procedure

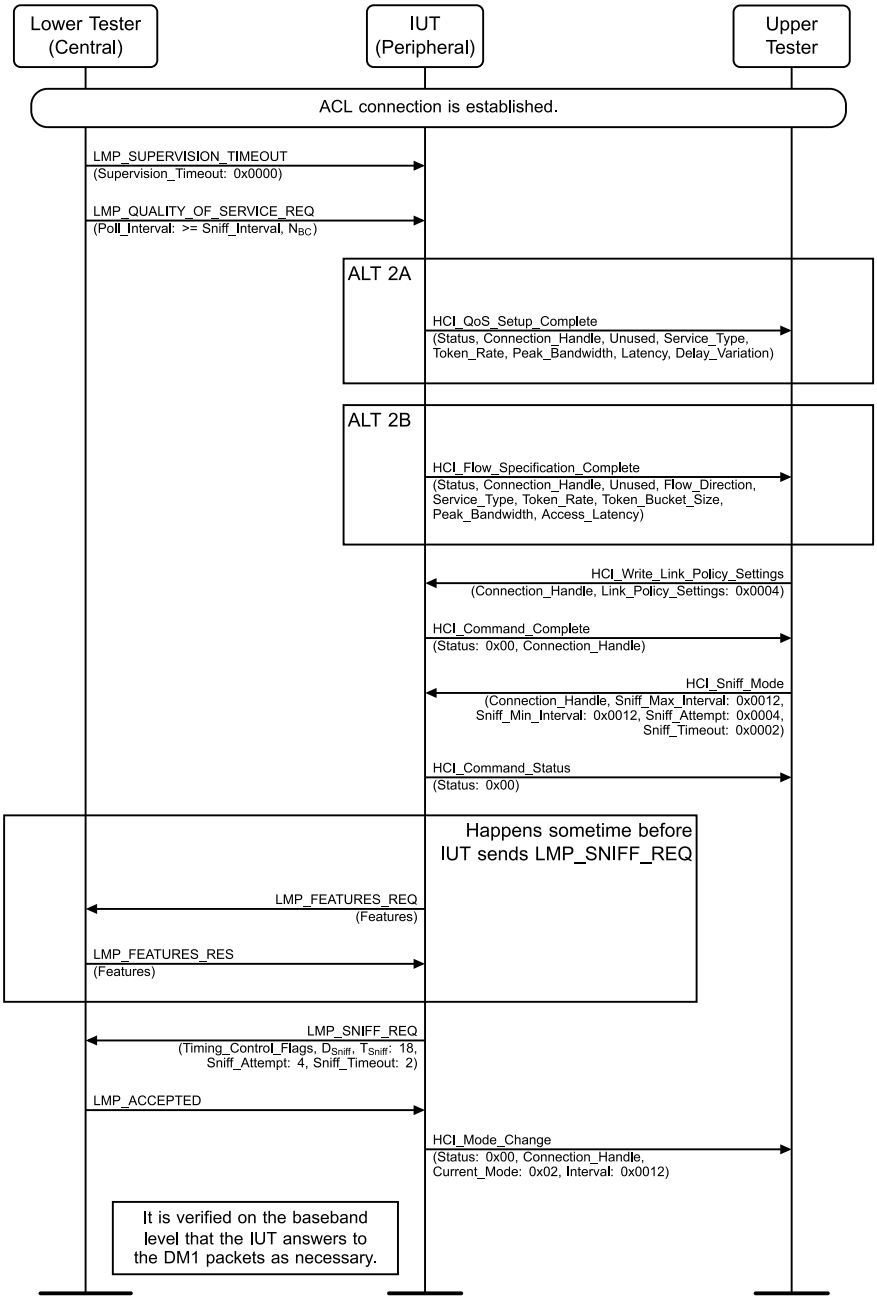


Figure 4.8-21: LMP/LIH/BV-15-C [Initiate Sniff Mode, Peripheral] MSC

1. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU with the Supervision_Timeout set to 0x0000 and an LMP_QUALITY_OF_SERVICE_REQ PDU with the Poll_Interval greater than or equal to the Sniff_Interval and N_{BC}.
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   2A.1. The IUT sends an HCI_QoS_Setup_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

Alternative 2B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):

2B.1. The IUT sends an HCI_Flow_Specification_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.

3. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0004 (Sniff Mode) and receives a successful HCI_Command_Complete event in response.
4. The Upper Tester sends an HCI_Sniff_Mode command to the IUT with the Connection_Handle, Sniff_Max_Interval set to 0x0012, Sniff_Min_Interval set to 0x0012, Sniff_Attempt set to 0x0004, and Sniff_Timeout set to 0x0002 and receives a successful HCI_Command_Status event in response.
5. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
6. The Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with the Features parameter.
7. The IUT sends an LMP_SNIFF_REQ PDU to the Lower Tester with Timing_Control_Flags, D_{Sniff}, T_{Sniff} set to 18, Sniff_Attempt set to 4, and Sniff_Timeout set to 2.
8. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
9. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval set to 0x0012.
10. The Lower Tester sends POLL packets to the IUT according to Figure 4.8-22.

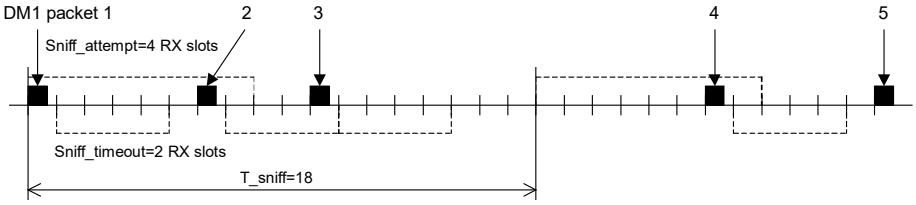


Figure 4.8-22: LMP/LIH/BV-15-C, Polling

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SNIFF_REQ PDU to the Lower Tester.

  The IUT enters Sniff mode and acknowledges DM1 packets 1–4 and does not acknowledge DM1 packet 5 for a period of 20*T_{Sniff} slots.

- Notes

  Timing_Control_Flags and D_{Sniff} are determined by CLK_{27} of the Central.

LMP/LIH/BV-16-C [Exit Sniff Mode]

- Test Purpose

  Verify that the IUT exits Sniff mode upon request from the Lower Tester.

  Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.3.2

- Initial Condition
  - See the “Default settings” section.
  - The Lower Tester is the Central and initiates the service. The IUT is the Peripheral.
  - The IUT is already in Sniff mode by request of the Lower Tester.
  - The Supervision Timeout is disabled.
- Test Procedure

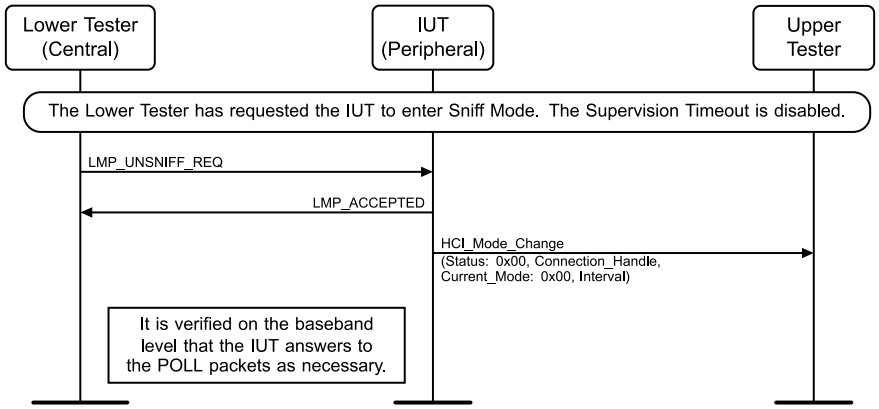


Figure 4.8-23: LMP/LIH/BV-16-C [Exit Sniff Mode] MSC

1. The Lower Tester sends an LMP_UNSNIFF_REQ PDU to the IUT.
2. The IUT responds with an LMP_ACCEPTED PDU with the LMP_UNSNIFF_REQ PDU Opcode.
3. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
4. The Lower Tester sends POLL packets to the IUT in all Central slots.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_UNSNIFF_REQ PDU from the Lower Tester.

  The IUT exits Sniff mode and acknowledges all POLL packets sent by the Lower Tester during a period of 200 slots.

LMP/LIH/BV-17-C [Accept Sniff Reject]

- Test Purpose

  Verify that the IUT accepts that the Lower Tester declines the Sniff mode request and remains in Active mode.

  Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.3.1

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester is the Central. The IUT is the Peripheral and initiates the service by requesting.
- Test Procedure

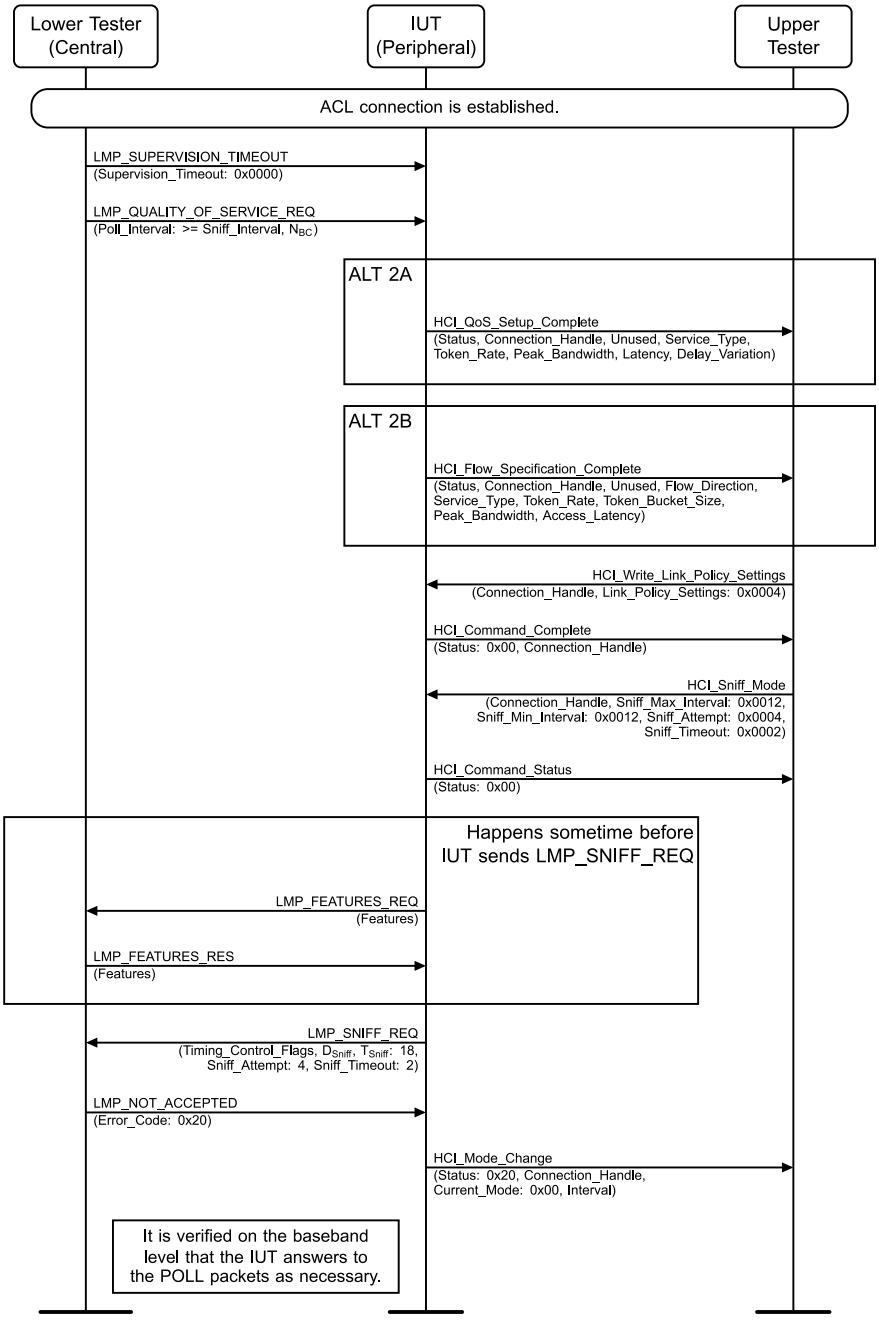


Figure 4.8-24: LMP/LIH/BV-17-C [Accept Sniff Reject] MSC

1. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU with the Supervision_Timeout set to 0x0000 and an LMP_QUALITY_OF_SERVICE_REQ PDU with the Poll_Interval greater than or equal to the Sniff_Interval and N_{BC}.

2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   2A.1. The IUT sends an HCI_QoS_Setup_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.
   Alternative 2B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):
   2B.1. The IUT sends an HCI_Flow_Specification_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.
3. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0004 (Sniff Mode) and receives a successful HCI_Command_Complete event in response.
4. The Upper Tester sends an HCI_Sniff_Mode command to the IUT with the Connection_Handle, Sniff_Max_Interval set to 0x0012, Sniff_Min_Interval set to 0x0012, Sniff_Attempt set to 0x0004, and Sniff_Timeout set to 0x0002 and receives a successful HCI_Command_Status event in response.
5. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
6. The Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with the Features parameter.
7. The IUT sends an LMP_SNIFF_REQ PDU to the Lower Tester with Timing_Control_Flags, D_{Sniff}, T_{Sniff} set to 18, Sniff_Attempt set to 4, and Sniff_Timeout set to 2.
8. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
9. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
10. The Lower Tester sends POLL packets to the IUT in all Central slots.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SNIFF_REQ PDU to the Lower Tester.

  The IUT does not enter Sniff mode after the Lower Tester sends the LMP_NOT_ACCEPTED PDU to the IUT. The IUT acknowledges all POLL packets sent by the Lower Tester during a period of 200 slots.

- Notes

  Timing_Control_Flags and D_{Sniff} are determined by CLK_{27} of the Central.

4.8.10 Sniff mode – Central

Verify that the ACL connection between two Bluetooth devices can be placed in Sniff mode. The IUT is the Central.

LMP/LIH/BV-18-C [Initiate Sniff Mode, Central]

- Test Purpose

  Verify that the IUT can request that the Lower Tester enter into Sniff mode and does not address the Lower Tester with its LT_ADDR outside the Sniff slots.

  Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.3.1

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Central and initiates the service by requesting. The Lower Tester is the Peripheral.
  - The IUT must page the Lower Tester to become the Central of the piconet.

- Test Procedure

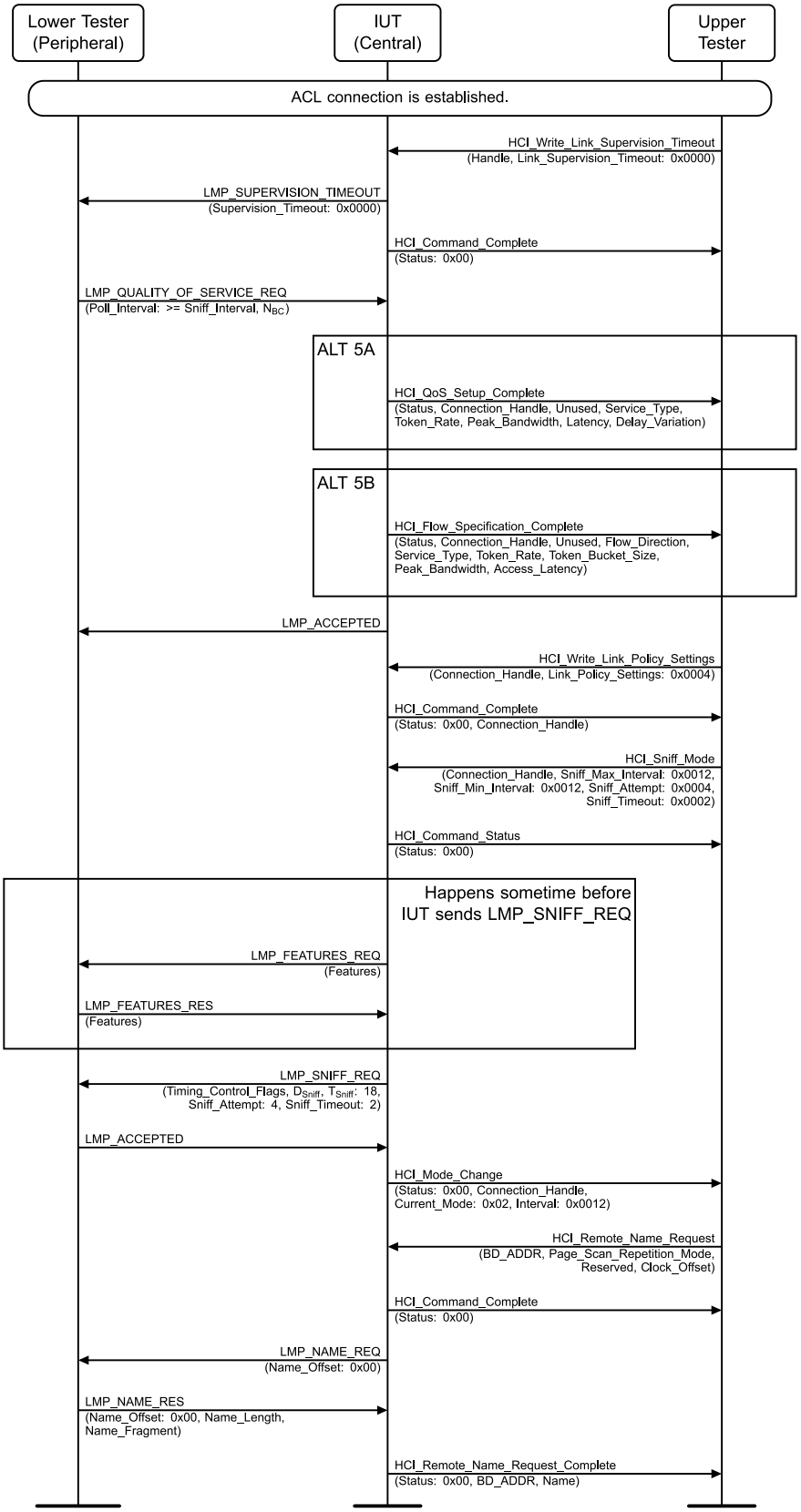


Figure 4.8-25: LMP/LIH/BV-18-C [Initiate Sniff Mode, Central] MSC

1. The Upper Tester sends an HCI_Write_Link_Supervision_Timeout command to the IUT with the Handle and Link_Supervision_Timeout set to 0x0000.
2. The IUT sends an LMP_SUPERVISION_TIMEOUT PDU to the Lower Tester with the Supervision_Timeout set to 0x0000.
3. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
4. The Lower Tester sends an LMP_QUALITY_OF_SERVICE_REQ PDU with the Poll_Interval greater than or equal to the Sniff_Interval and N_{BC}.
5. Perform either alternative 5A or 5B depending on the IUT's response.
   Alternative 5A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   - 5A.1. The IUT sends an HCI_QoS_Setup_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

   Alternative 5B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):
   - 5B.1. The IUT sends an HCI_Flow_Specification_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.
6. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_QUALITY_OF_SERVICE_REQ PDU Opcode.
7. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0004 (Sniff Mode) and receives a successful HCI_Command_Complete event in response.
8. The Upper Tester sends an HCI_Sniff_Mode command to the IUT with the Connection_Handle, Sniff_Max_Interval set to 0x0012, Sniff_Min_Interval set to 0x0012, Sniff_Attempt set to 0x0004, and Sniff_Timeout set to 0x0002 and receives a successful HCI_Command_Status event in response.
9. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
10. The Lower Tester responds with an LMP_FEATURES_RES PDU to the IUT with the Features parameter.
11. The IUT sends an LMP_SNIFF_REQ PDU to the Lower Tester with Timing_Control_Flags, D_{Sniff}, T_{Sniff} set to 18, Sniff_Attempt set to 4, and Sniff_Timeout set to 2.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
13. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval set to 0x0012.
14. The Upper Tester sends an HCI_Remote_Name_Request command to the IUT with the BD_ADDR, Page_Scan_Repetition_Mode, Reserved, and Clock_Offset and receives a successful HCI_Command_Status event in response.
15. The IUT sends an LMP_NAME_REQ PDU to the Lower Tester with Name_Offset set to 0x00.
16. The Lower Tester responds with an LMP_NAME_RES PDU with Name_Offset set to 0x00, Name_Length, and Name_Fragment.
17. The IUT sends a successful HCI_Remote_Name_Request_Complete event to the Upper Tester with the BD_ADDR and Remote_Name of the Lower Tester.

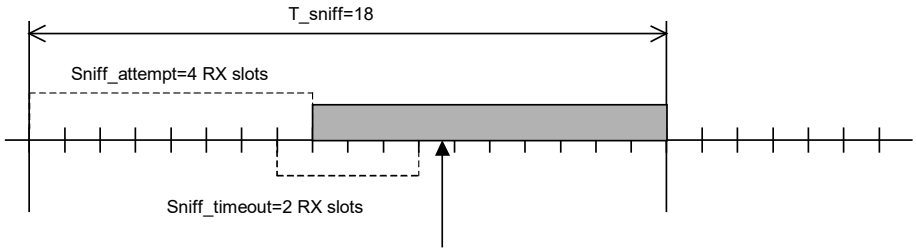


Figure 4.8-26: LMP/LIH/BV-18-C, Verifying

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SNIFF_REQ PDU to the Lower Tester.

  The ACL link enters Sniff mode, and no polling is done outside the sniff interval.

  The IUT does not address the Lower Tester with the LT_ADDR in the grey zone shown in Figure 4.8-26, unless a packet follows in the Sniff_Timeout frame after a packet received in the Sniff_Attempt frame, for a period of $20*T_{Sniff}$ slots.

- Notes

  Timing_Control_Flags and D_{Sniff} are determined by CLK_{27} of the Central.

LMP/LIH/BV-19-C [Request Sniff Mode Exit]

- Test Purpose

  Verify that the IUT can request the Lower Tester to exit Sniff mode.

  Baseband functionality is tested in the test case.

- Reference

  [1] 4.5.3.2

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Central and initiates the service. The Lower Tester is the Peripheral.
  - The Lower Tester configures the QoS to have the IUT poll the Lower Tester every six slots.
  - The IUT has requested the Lower Tester into Sniff mode.
  - The Supervision Timeout is disabled.

- Test Procedure

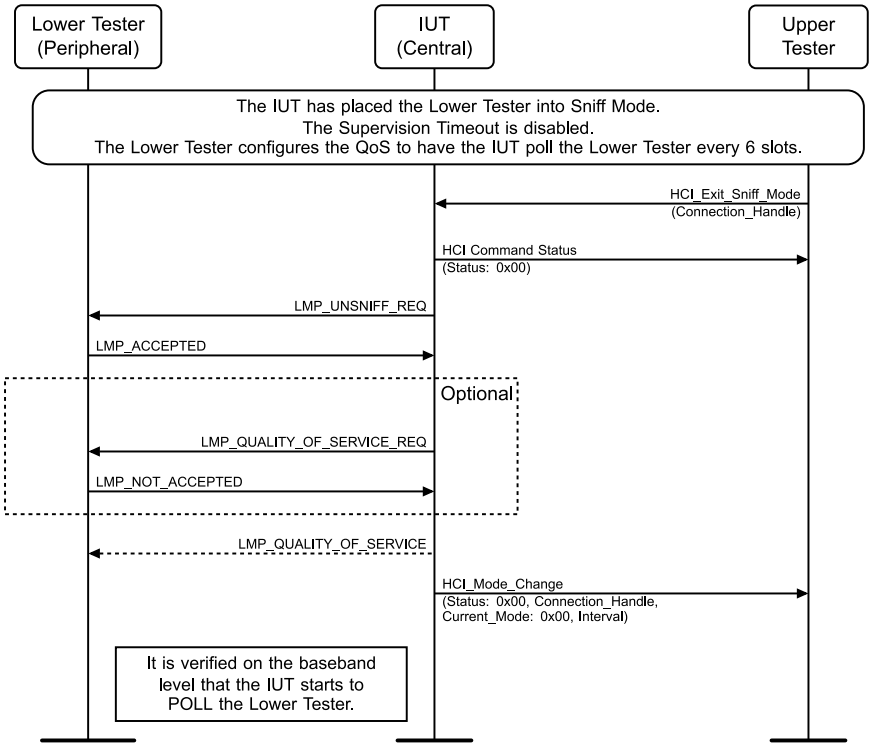


Figure 4.8-27: LMP/LIH/BV-19-C [Request Sniff Mode Exit] MSC

1. The Upper Tester sends an HCI_Exit_Sniff_Mode command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_UNSNIFF_REQ PDU to the Lower Tester.
3. The Lower Tester responds with an LMP_ACCEPTED PDU with the LMP_UNSNIFF_REQ PDU Opcode.
4. Optionally, the IUT sends an LMP_QUALITY_OF_SERVICE_REQ PDU to the Lower Tester with the Poll_Interval and N_{BC} and receives an LMP_NOT_ACCEPTED PDU with the LMP_QUALITY_OF_SERVICE_REQ PDU Opcode in response.
5. Optionally, the IUT sends an LMP_QUALITY_OF_SERVICE PDU to the Lower Tester with the Poll_Interval and N_{BC}.
6. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
7. The IUT sends POLL packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_UNSNIFF_REQ PDU to the Lower Tester.

  The ACL link exits Sniff mode, and the IUT sends POLL packets to the Lower Tester during a period of 1600 slots. The time between the IUT's POLL packets in Active mode is less than or equal to T_{poll} at least 95% of the time. T_{poll} is 6 unless the IUT changes it by sending an LMP_QUALITY_OF_SERVICE PDU to the Lower Tester.

4.8.11 Sniff mode – Both Central and Peripheral

Verify that the IUT declines the Sniff mode request correctly. The role of the IUT is of no importance.

LMP/LIH/BV-20-C [Sniff Mode Reject]

- Test Purpose

  Verify that the IUT responds that it does not support Sniff mode upon a request from the Lower Tester.

- Reference

  [1] 4.5.3.1

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.

- Test Procedure

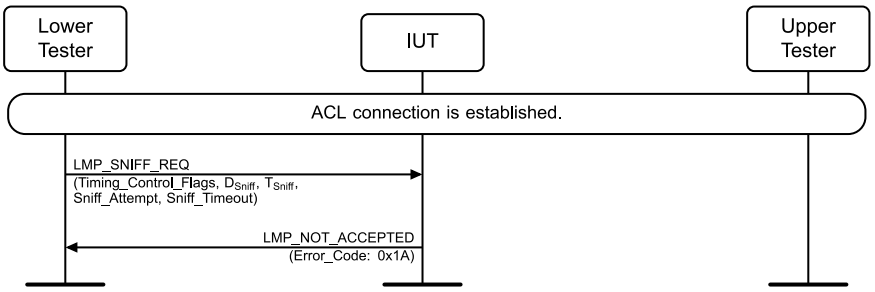


Figure 4.8-28: LMP/LIH/BV-20-C [Sniff Mode Reject] MSC

1. The Lower Tester sends an LMP_SNIFF_REQ PDU to the IUT with with Timing_Control_Flags, D_{Sniff}, T_{Sniff}, Sniff_Attempt, and Sniff_Timeout.
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode and Error_Code set to 0x1A (Unsupported LMP Feature).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1A (Unsupported LMP Feature) upon reception of the LMP_SNIFF_REQ PDU from the Lower Tester.

4.8.12 Power control – Both Central and Peripheral

Verify that a unit can request a change of another unit’s TX power. The role of the IUT is of no importance.

LMP/LIH/BV-35-C [Lowest Power Report]

- Test Purpose

  Verify that the IUT reports that it transmits at the lowest power upon requests from the Lower Tester to decrease the power.

- Reference

  [1] 4.1.3

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.
  - The time it takes the IUT to increase or decrease its output power one step is defined by TSPX_power_control_step_rate.

- Test Procedure

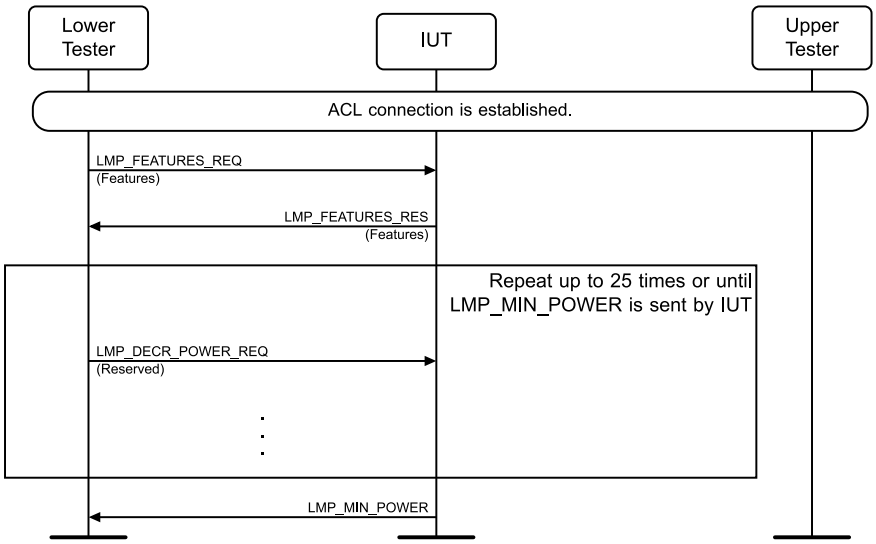


Figure 4.8-29: LMP/LIH/BV-35-C [Lowest Power Report] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. Execute Step 3 up to 25 times or until Step 4 occurs. The time between LMP_DECR_POWER_REQ PDUs sent by the Lower Tester is greater than 5 seconds or greater than TSPX_power_control_step_rate, if defined.
4. The Lower Tester sends an LMP_DECR_POWER_REQ PDU to the IUT with Reserved.
5. The IUT sends an LMP_MIN_POWER PDU to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_MIN_POWER PDU to the Lower Tester.

LMP/LIH/BV-36-C [Highest Power Report]

- Test Purpose

  Verify that the IUT reports that it transmits at the highest power upon requests from the Lower Tester to increase the power.

- Reference

  [1] 4.1.3

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.
  - The time it takes the IUT to increase or decrease its output power one step is defined by TSPX_power_control_step_rate.

- Test Procedure

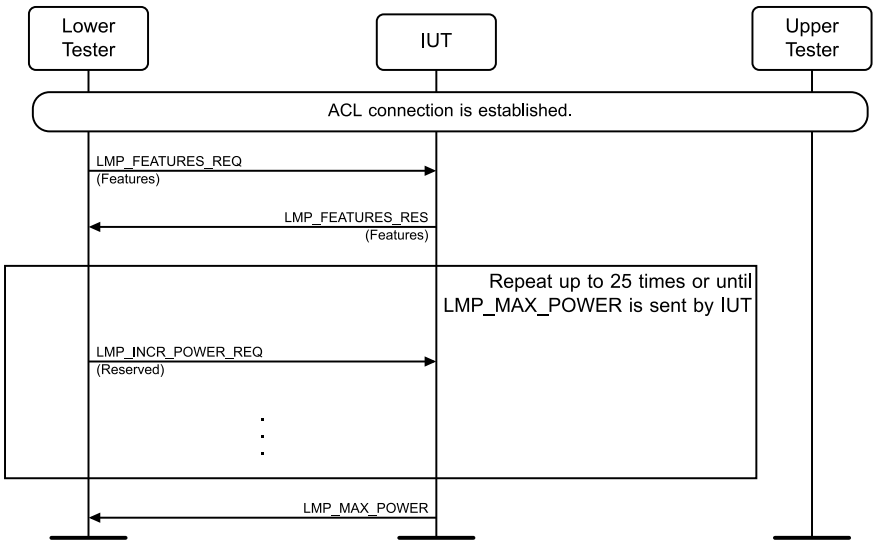


Figure 4.8-30: LMP/LIH/BV-36-C [Highest Power Report] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

Execute Step 3 up to 25 times or until Step 4 occurs. The time between LMP_INCR_POWER_REQ PDUs sent by the Lower Tester is greater than 5 seconds or greater than TSPX_power_control_step_rate, if defined.

3. The Lower Tester sends an LMP_INCR_POWER_REQ PDU to the IUT with Reserved.
4. The IUT sends an LMP_MAX_POWER PDU to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_MAX_POWER PDU to the Lower Tester.

LMP/LIH/BV-76-C [Request Decreased Power]

- Test Purpose

  Verify that the IUT sends the LMP_DECR_POWER_REQ PDU when the Lower Tester sends POLL packets at power levels around the upper threshold of the Golden Receive Power Range (measured at the IUT's antenna connector).

- Reference

  [1] 4.5.3

  [6] 4.6

- Initial Condition

  - See the "Default settings" section.
  - The Lower Tester is the Central. The IUT is the Peripheral.
  - An ACL connection has been established.
  - The time it takes the IUT to increase or decrease its output power one step is defined by TSPX_power_control_step_rate.
  - The upper threshold of the Golden Receive Power Range of the IUT is defined by TSPX_receive_power_golden_range_upper_limit.

- Test Procedure

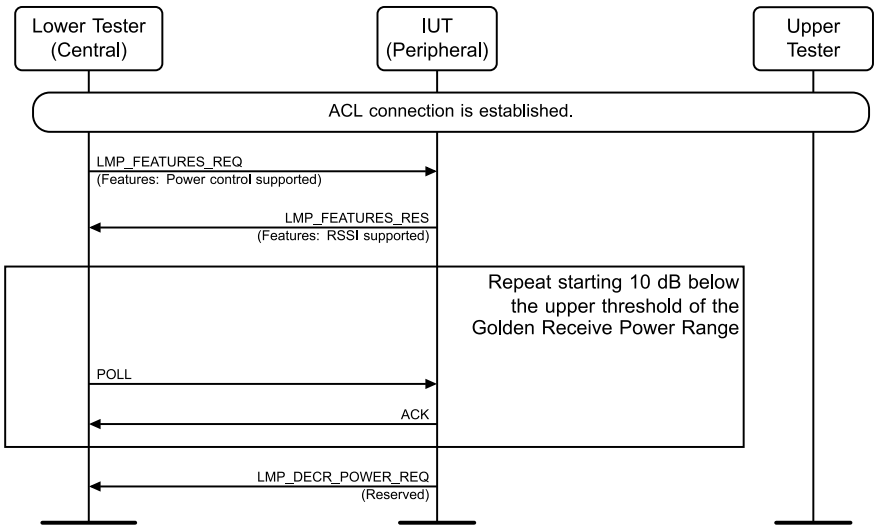


Figure 4.8-31: LMP/LIH/BV-76-C [Request Decreased Power] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter indicating "Power control requests" feature support.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

Execute Steps 3–4 continuously with the Lower Tester transmitting with its power level 10 dB below TSPX_receive_power_golden_range_upper_limit. The Lower Tester increases its power at steps of 2 dB, at a rate of not more than one step per 5-second period or per TSPX_power_control_step_rate period, if defined. The Lower Tester does not increase the power level more than 7 dB above TSPX_receive_power_golden_range_upper_limit.

3. The Lower Tester sends a POLL packet to the IUT.
4. The IUT responds to the Lower Tester with an ACK.
5. The IUT sends an LMP_DECR_POWER_REQ PDU to the Lower Tester with Reserved.

- Test Condition

  Nominal Test Conditions, see [6] Section 5.1.

  The Lower Tester and the IUT are connected with a cable and RF attenuator to give sufficient measurement accuracy. TSPX_RF_Attenuation is used to give the value for the RF attenuator calculated from the IUT's Golden Receive Power Range, the cable loss, and the Lower Tester's TX power range.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_DECR_POWER_REQ PDU to the Lower Tester.

- Notes

  The initial power level is 10 dB below the declared upper threshold to give a margin. The Lower Tester has an accuracy not worse than ±3 dB in transmitted power level. Until dedicated Bluetooth test systems are available, it is allowed to use other values for step size and accuracy in the transmitted power level. Also, the POLL packet can be replaced with other packet types.

LMP/LIH/BV-77-C [Request Increased Power]

- Test Purpose

  Verify that the IUT sends the LMP_INCR_POWER_REQ PDU when the Lower Tester sends POLL packets at power levels around the lower threshold of the Golden Receive Power Range (measured at the IUT's antenna connector).

- Reference

  [1] 4.5.3

  [6] 4.6

- Initial Condition

  - See the "Default settings" section.
  - The Lower Tester is the Central. The IUT is the Peripheral.
  - An ACL connection has been established.
  - The time it takes the IUT to increase or decrease its output power one step is defined by TSPX_power_control_step_rate.
  - The lower threshold of the Golden Receive Power Range of the IUT is defined by TSPX_receive_power_golden_range_lower_limit.

- Test Procedure

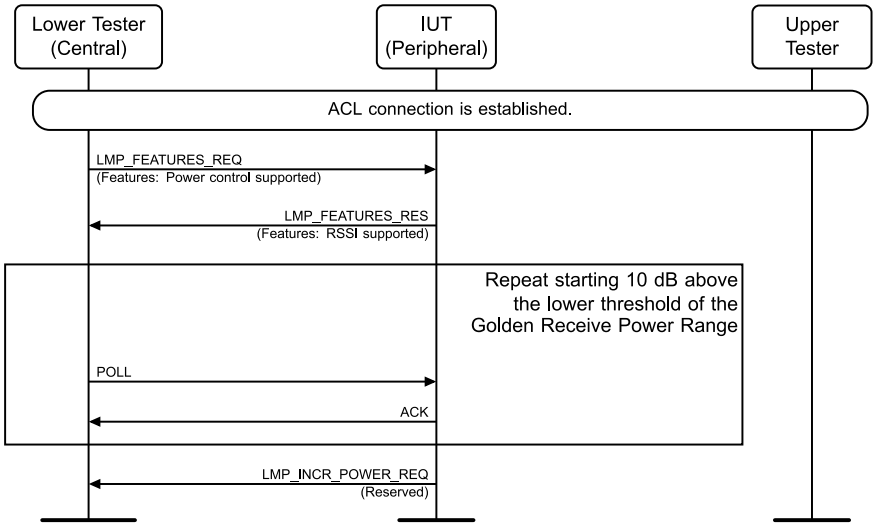


Figure 4.8-32: LMP/LIH/BV-77-C [Request Increased Power] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter indicating “Power control requests” feature support.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

Execute Steps 3–4 continuously with the Lower Tester transmitting with its power level 10 dB above TSPX_receive_power_golden_range_lower_limit. The Lower Tester decreases its power at steps of 2 dB, at a rate of not more than one step per 5-second period or per TSPX_power_control_step_rate period, if defined. The Lower Tester does not increase the power level more than 6 dB below TSPX_receive_power_golden_range_lower_limit.

3. The Lower Tester sends a POLL packet to the IUT.
4. The IUT responds to the Lower Tester with an ACK.
5. The IUT sends an LMP_INCR_POWER_REQ PDU to the Lower Tester with Reserved.

- Test Condition

  Nominal Test Conditions, see [6] Section 5.1.

  The Lower Tester and the IUT are connected with a cable and RF attenuator to give sufficient measurement accuracy. TSPX_RF_Attenuation is used to give the value for the RF attenuator calculated from the IUT’s Golden Receive Power Range, the cable loss, and the Lower Tester’s TX power range.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_INCR_POWER_REQ PDU to the Lower Tester.

- Notes

  The initial power level is 10 dB above the declared lower threshold to give a margin. The Lower Tester has an accuracy not worse than ±3 dB in transmitted power level. Until dedicated Bluetooth test systems are available, it is allowed to use other values for step size and accuracy in the transmitted power level. Also, the POLL packet can be replaced with other packet types.

LMP/LIH/BV-127-C [Respond to EPC Increment Request]

- Test Purpose

  Verify that the IUT will respond correctly to an Enhanced Power Control increment single step request and the HCI_Read_Enhanced_Transmit_Power_Level command.

- Reference

  [1] 4.1.3

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

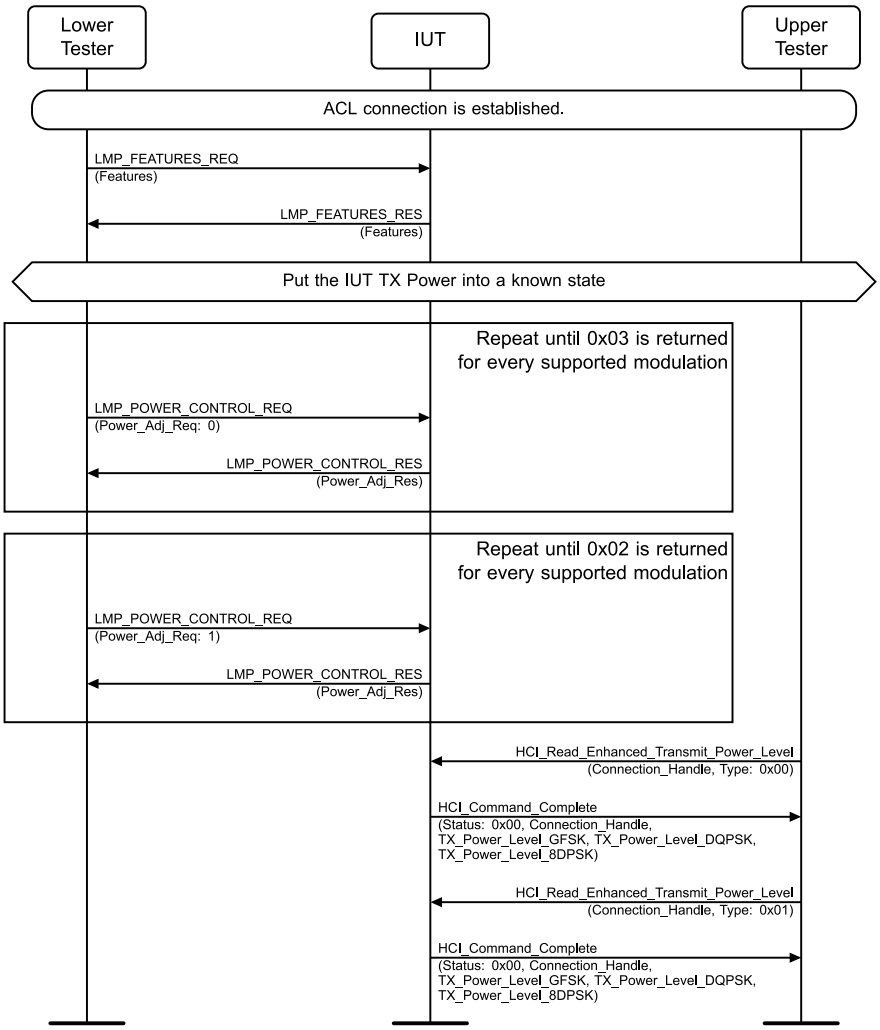


Figure 4.8-33: LMP/LIH/BV-127-C [Respond to EPC Increment Request] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The IUT’s TX Power is placed into a known state.

Execute Steps 4–5 until 0x03 (Min Power) is returned for every supported modulation.

4. The Lower Tester sends an LMP_POWER_CONTROL_REQ PDU to the IUT with Power_Adj_Req set to 0.
5. The IUT responds to the Lower Tester with an LMP_POWER_CONTROL_RES PDU with Power_Adj_Res.

Execute Steps 6–7 until 0x02 (Max Power) is returned for every supported modulation.

6. The Lower Tester sends an LMP_POWER_CONTROL_REQ PDU to the IUT with Power_Adj_Req set to 1.
7. The IUT responds to the Lower Tester with an LMP_POWER_CONTROL_RES PDU with Power_Adj_Res.
8. The Upper Tester sends an HCI_Read_Enhanced_Transmit_Power_Level command to the IUT with the Connection_Handle and Type set to 0x00 (Read Current Transmit Power Level).
9. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with the Connection_Handle, TX_Power_Level_GFSK, TX_Power_Level_DQPSK, and TX_Power_Level_8DPSK.
10. The Upper Tester sends an HCI_Read_Enhanced_Transmit_Power_Level command to the IUT with the Connection_Handle and Type set to 0x01 (Read Maximum Transmit Power Level).
11. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with the Connection_Handle, TX_Power_Level_GFSK, TX_Power_Level_DQPSK, and TX_Power_Level_8DPSK.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends an LMP_FEATURES_RES PDU to the Lower Tester indicating support for power control, power control requests, and Enhanced Power Control.

  In Step 7 during the looping:

  - The IUT sends the LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x01 (Changed one step) for at least one supported modulation.
  - The IUT sends the LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x01 (Changed one step) for at least one supported modulation that is not at the maximum.
  - The IUT sends the LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x02 (Max Power) for all supported modulation.

  The IUT responds to the HCI_Read_Enhanced_Transmit_Power_Level commands from the Upper Tester with an HCI Command Complete event with the current and maximum power levels equal when all supported modulations report that they are at maximum.

LMP/LIH/BV-128-C [Respond to EPC Decrement Request]

- Test Purpose

  Verify that the IUT will respond correctly to an Enhanced Power Control decrement single step request and the HCI_Read_Enhanced_Transmit_Power_Level command.

- Reference

  [1] 4.1.3

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
- Test Procedure

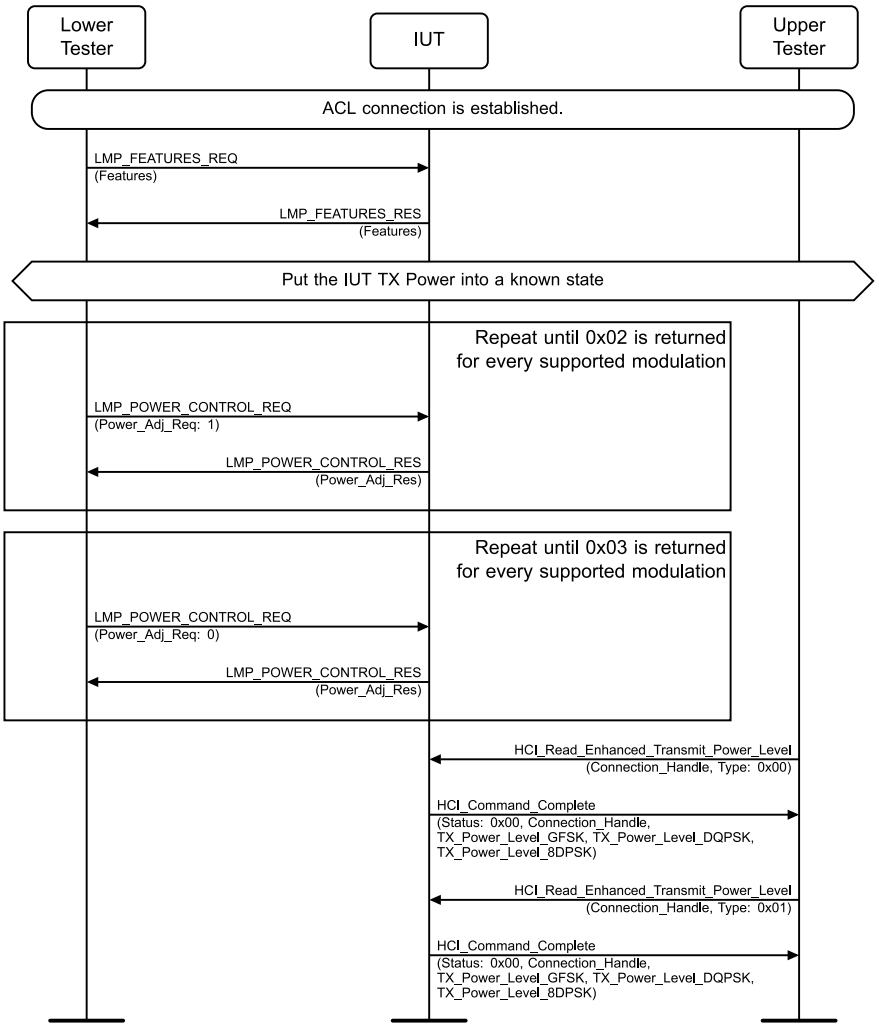


Figure 4.8-34: LMP/LIH/BV-128-C [Respond to EPC Decrement Request] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The IUT’s TX Power is placed into a known state.

Execute Steps 4–5 until 0x02 (Max Power) is returned for every supported modulation.

4. The Lower Tester sends an LMP_POWER_CONTROL_REQ PDU to the IUT with Power_Adj_Req set to 1.
5. The IUT responds to the Lower Tester with an LMP_POWER_CONTROL_RES PDU with Power_Adj_Res.

Execute Steps 6–7 until 0x03 (Min Power) is returned for every supported modulation.

6. The Lower Tester sends an LMP_POWER_CONTROL_REQ PDU to the IUT with Power_Adj_Req set to 0.
7. The IUT responds to the Lower Tester with an LMP_POWER_CONTROL_RES PDU with Power_Adj_Res.
8. The Upper Tester sends an HCI_Read_Enhanced_Transmit_Power_Level command to the IUT with the Connection_Handle and Type set to 0x00 (Read Current Transmit Power Level).
9. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with the Connection_Handle, TX_Power_Level_GFSK, TX_Power_Level_DQPSK, and TX_Power_Level_8DPSK.
10. The Upper Tester sends an HCI_Read_Enhanced_Transmit_Power_Level command to the IUT with the Connection_Handle and Type set to 0x01 (Read Maximum Transmit Power Level).
11. The IUT sends a successful HCI_Command_Complete event to the Upper Tester with the Connection_Handle, TX_Power_Level_GFSK, TX_Power_Level_DQPSK, and TX_Power_Level_8DPSK.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends an LMP_FEATURES_RES PDU to the Lower Tester indicating support for power control, power control requests, and Enhanced Power Control.

  In Step 7 during the looping:

  - The IUT sends the LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x01 (Changed one step) for at least one supported modulation.
  - The IUT sends the LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x01 (Changed one step) for at least one supported modulation that is not at the minimum.
  - The IUT sends the LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x03 (Min Power) for all supported modulation.

  The IUT responds to the HCI_Read_Enhanced_Transmit_Power_Level commands from the Upper Tester with an HCI Command Complete event with the current and maximum power levels not equal when all supported modulations report that they are at minimum.

LMP/LIH/BV-129-C [Respond to EPC go to Maximum Power Level]

- Test Purpose

  Verify that the IUT can be requested to go to maximum power level using the Enhanced Power Control go to maximum request.

- Reference

  [1] 4.1.3

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

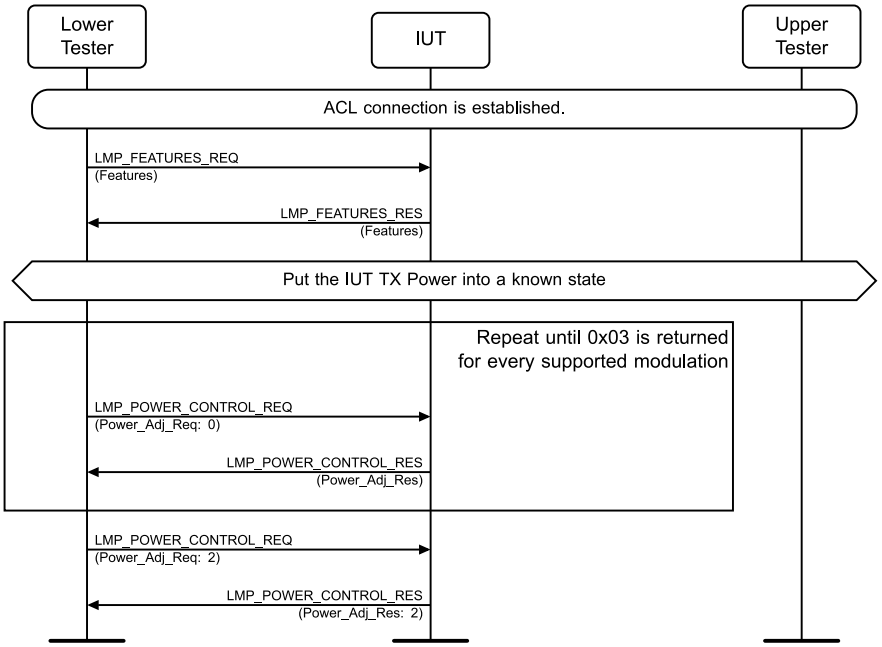


Figure 4.8-35: LMP/LIH/BV-129-C [Respond to EPC go to Maximum Power Level] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The IUT's TX Power is placed into a known state.

Execute Steps 4–5 until 0x03 (Min Power) is returned for every supported modulation.

4. The Lower Tester sends an LMP_POWER_CONTROL_REQ PDU to the IUT with Power_Adj_Req set to 0.
5. The IUT responds to the Lower Tester with an LMP_POWER_CONTROL_RES PDU with Power_Adj_Res.
6. The Lower Tester sends an LMP_POWER_CONTROL_REQ PDU to the IUT with Power_Adj_Req set to 0x02.
7. The IUT responds to the Lower Tester with an LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x02 for all supported modulations.

- Expected Outcome

Pass verdict

In Step 2, the IUT sends an LMP_FEATURES_RES PDU to the Lower Tester indicating support for power control, power control requests, and Enhanced Power Control.

After the Lower Tester sends the LMP_POWER_CONTROL_REQ PDU with Power_Adj_Req set to 0x02, the IUT sends the LMP_POWER_CONTROL_RES PDU with Power_Adj_Res set to 0x02 for all supported modulations.

LMP/LIH/BV-130-C [Request an EPC Increment]

- Test Purpose

  Verify that the IUT will request an Enhanced Power Control increment single step.

- Reference

  [6] 4.6

  [1] 4.5.3

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The lower threshold of the Golden Receive Power Range of the IUT is defined by TSPX_receive_power_golden_range_lower_limit.

- Test Procedure

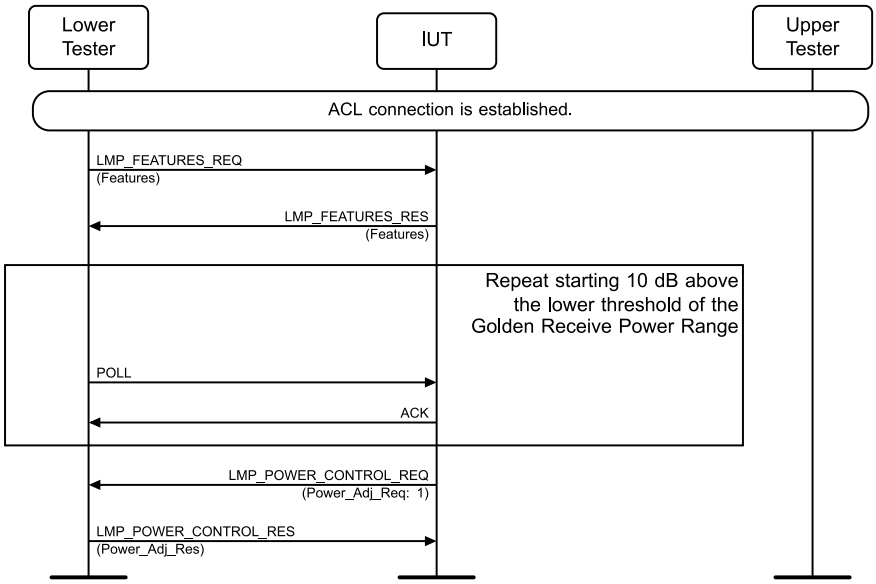


Figure 4.8-36: LMP/LIH/BV-130-C [Request an EPC Increment] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

Execute Steps 3–4 continuously with the Lower Tester transmitting with its power level 10 dB above TSPX_receive_power_golden_range_lower_limit. The Lower Tester decreases its power at steps of 2 dB. The Lower Tester does not increase the power level more than 6 dB below TSPX_receive_power_golden_range_lower_limit.

3. The Lower Tester sends a POLL packet to the IUT.
4. The IUT responds to the Lower Tester with a NULL packet.
5. The IUT sends an LMP_POWER_CONTROL_REQ PDU to the Lower Tester with Power_Adj_Req set to 0x01 (Increment power one step).
6. The Lower Tester responds to the IUT with an LMP_POWER_CONTROL_RES PDU with the Power_Adj_Res.

- Test Condition

  Nominal Test Conditions; see [6] Section 5.1.

  The Lower Tester and the IUT must be connected with cable and RF attenuator to give sufficient measurement accuracy.

  TSPX_RF_Attenuation is used to give the value for the RF attenuator calculated from the IUT's Receive Power Range, the cable loss, and the Lower Tester's TX power range.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends an LMP_FEATURES_RES PDU to the Lower Tester indicating support for power control, power control requests, and Enhanced Power Control.

  In Step 5, the IUT sends the LMP_POWER_CONTROL_REQ PDU with Power_Adj_Req set to 0x01 (Increment power one step).

- Notes

  The initial power level is 10 dB above the declared lower threshold to give a margin. The Lower Tester has an accuracy not worse than ±3 dB in transmitted power level. Until dedicated Bluetooth test systems are available, it is allowed to use other values for step size and accuracy in the transmitted power level. The POLL packet can be replaced with other packet types.

LMP/LIH/BV-131-C [Request an EPC Decrement]

- Test Purpose

  Verify that the IUT will request an Enhanced Power Control decrement single step.

- Reference

  [6] 4.6

  [1] 4.5.3

- Initial Condition

  - See the "Default settings" section.
  - An ACL connection has been established.
  - The upper threshold of the Golden Receive Power Range of the IUT is defined by TSPX_receive_power_golden_range_upper_limit.

- Test Procedure

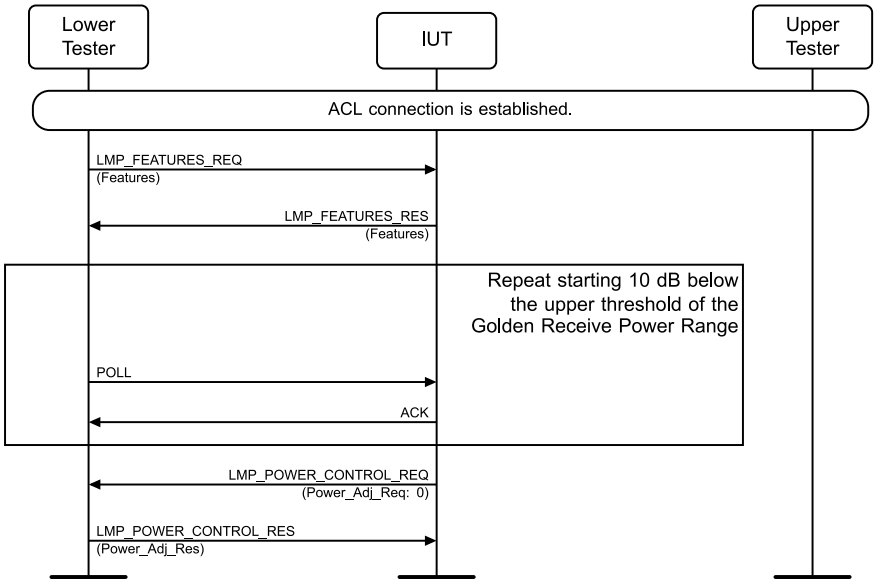


Figure 4.8-37: LMP/LIH/BV-131-C [Request an EPC Decrement] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

Execute Steps 3–4 continuously with the Lower Tester transmitting with its power level 10 dB below TSPX_receive_power_golden_range_upper_limit. The Lower Tester increases its power at steps of 2 dB. The Lower Tester does not increase the power level more than 7 dB above TSPX_receive_power_golden_range_lower_limit.

3. The Lower Tester sends a POLL packet to the IUT.
4. The IUT responds to the Lower Tester with a NULL packet.
5. The IUT sends an LMP_POWER_CONTROL_REQ PDU to the Lower Tester with Power_Adj_Req set to 0x00 (Decrement power one step).
6. The Lower Tester responds to the IUT with an LMP_POWER_CONTROL_RES PDU with the Power_Adj_Res.

- Test Condition

  Nominal Test Conditions; see [6] Section 5.1.

  The Lower Tester and the IUT must be connected with cable and RF attenuator to give sufficient measurement accuracy.

  TSPX_RF_Attenuation is used to give the value for the RF attenuator calculated from the IUT's Receive Power Range, the cable loss, and the Lower Tester's TX power range.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_FEATURES_RES PDU to the Lower Tester indicating support for power control, power control requests, and Enhanced Power Control.

  In Step 5, the IUT sends the LMP_POWER_CONTROL_REQ PDU with Power_Adj_Req set to 0x00 (decrement power one step).

- Notes

  The initial power level is 10 dB below the declared upper threshold to give a margin. The Lower Tester has accuracy not worse than ±3 dB in transmitted power level. Until dedicated Bluetooth test systems are available, it is allowed to use other values for step size and accuracy in the transmitted power level. Also, the POLL packet can be replaced with other packet types.

LMP/LIH/BV-133-C [Power Response Reports Unsupported Modulation Correctly]

- Test Purpose

  Verify that the IUT will respond correctly to an EPC power change request for an unsupported modulation type from a device that supports Enhanced Power Control.

- Reference

  [6] 4.6

  [1] 4.5.3

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Procedure

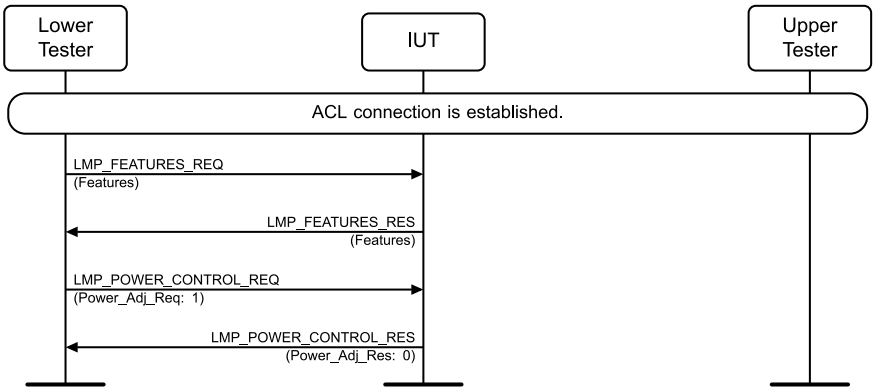


Figure 4.8-38: LMP/LIH/BV-133-C [Power Response Reports Unsupported Modulation Correctly] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The Lower Tester sends an LMP_POWER_CONTROL_REQ PDU to the IUT with Power_Adj_Req set to 0x01 (Increment power one step).
4. The IUT responds to the Lower Tester with an LMP_POWER_CONTROL_RES PDU with the Power_Adj_Res.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_FEATURES_RES PDU to the Lower Tester indicating support for power control, power control requests, and Enhanced Power Control.

  In Step 4, the IUT sends the LMP_POWER_CONTROL_RES PDU to the Lower Tester with Power_Adj_Res set to 0x00 (not supported) for any modulations that are not supported.

LMP/LIH/BV-152-C [Power Control Request – Not Supported]

- Test Purpose

  Verify that the IUT responds to an LMP_DECR_POWER_REQ and LMP_INCR_POWER_REQ correctly.

- Reference

  [1] 4.1.3

  [6] 4.6

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - Power Control Requests are not supported.

- Test Procedure

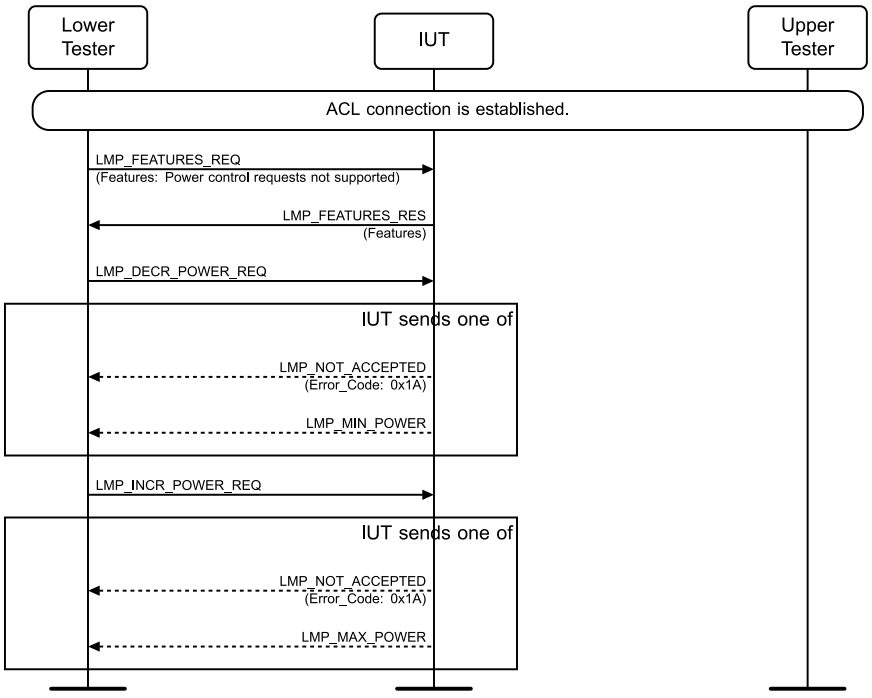


Figure 4.8-39: LMP/LIH/BV-152-C [Request Decreased Power] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester with the Features parameter indicating that Power Control Requests are not supported.
3. The Lower Tester sends an LMP_DECR_POWER_REQ PDU to the IUT.
4. The IUT sends an LMP_MIN_POWER PDU or an LMP_NOT_ACCEPTED PDU with the LMP_DECR_POWER_REQ PDU Opcode and Error_Code set to Unsupported Remote Feature (0x1A).
5. The Lower Tester sends an LMP_INCR_POWER_REQ PDU to the IUT.
6. The IUT sends an LMP_MAX_POWER PDU or an LMP_NOT_ACCEPTED PDU with the LMP_INCR_POWER_REQ PDU Opcode and Error_Code set to Unsupported Remote Feature (0x1A).

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends an LMP_MIN_POWER or LMP_NOT_ACCEPTED PDU to the Lower Tester.

  In Step 6, the IUT sends an LMP_MAX_POWER or LMP_NOT_ACCEPTED PDU to the Lower Tester.

4.8.13 Quality of Service (QoS) – Peripheral

Verify that a unit can request a change of maximum polling interval. The IUT is the Peripheral.

LMP/LIH/BV-39-C [Accept Polling Interval Notification]

- Test Purpose

  Verify that the IUT accepts the new maximum polling interval after notification from the Lower Tester.

- Reference

  [1] 4.1.8.1

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Peripheral. The Lower Tester is the Central and notifies the Peripheral.

- Test Procedure

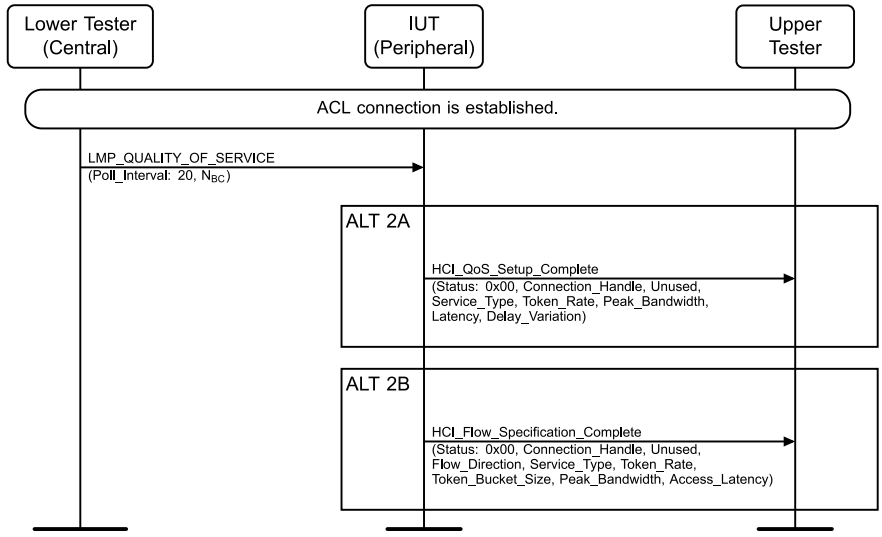


Figure 4.8-40: LMP/LIH/BV-39-C [Accept Polling Interval Notification] MSC

1. The Lower Tester sends an LMP_QUALITY_OF_SERVICE PDU to the IUT with Poll_Interval set to 20 and N_{BC}.
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   - 2A.1. The IUT sends a successful HCI_QoS_Setup_Complete event to the Upper Tester with the Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

   Alternative 2B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):
   - 2B.1. The IUT sends a successful HCI_Flow_Specification_Complete event to the Upper Tester with the Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.

- Expected Outcome

  Pass verdict

  In alternative 2A, the IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester.

  In alternative 2B, the IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester.

LMP/LIH/BV-40-C [Accept Polling Interval Request]

- Test Purpose

  Verify that the IUT accepts the new maximum polling interval after request from the Lower Tester. The maximum polling interval must be changed accordingly.

- Reference

  [1] 4.1.8.2

- Initial Condition
  - See the "Default settings" section.
  - An ACL connection has been established.
  - The IUT is the Peripheral. The Lower Tester is the Central and requests the Peripheral.

- Test Procedure

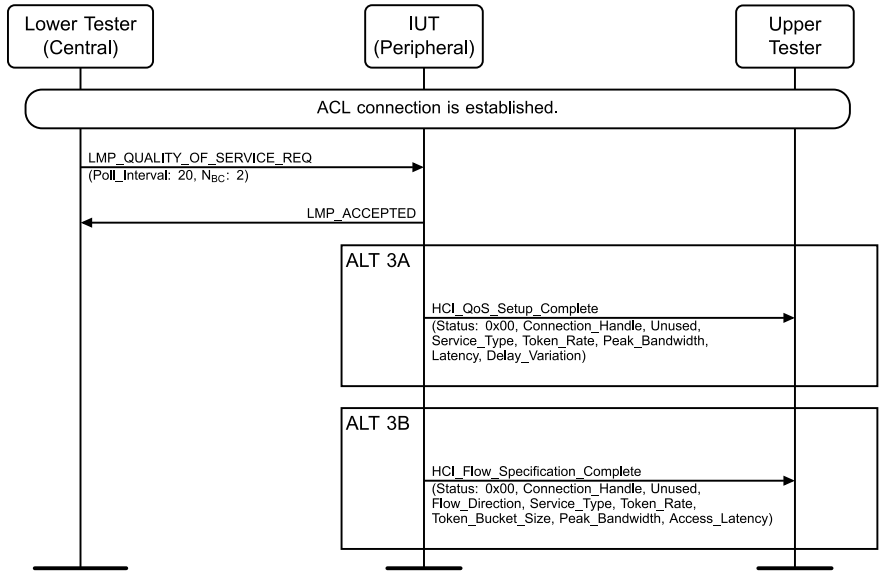


Figure 4.8-41: LMP/LIH/BV-40-C [Accept Polling Interval Request] MSC

1. The Lower Tester sends an LMP_QUALITY_OF_SERVICE_REQ PDU to the IUT with Poll_Interval set to 20 and N_{BC} set to 2.
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_QUALITY_OF_SERVICE_REQ PDU Opcode.
3. Perform either alternative 3A or 3B depending on the IUT's response:
   Alternative 3A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   - 3A.1. The IUT sends a successful HCI_QoS_Setup_Complete event to the Upper Tester with the Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

   Alternative 3B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):
   - 3B.1. The IUT sends a successful HCI_Flow_Specification_Complete event to the Upper Tester with the Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_QUALITY_OF_SERVICE_REQ PDU from the Lower Tester.

LMP/LIH/BV-41-C [Polling Interval Rejected]

- Test Purpose

  Verify that the IUT accepts a rejection of the Polling interval from the Lower Tester.

- Reference

  [1] 4.1.8.2

- Initial Condition
  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Peripheral and requests the Central. The Lower Tester is the Central.
- Test Procedure

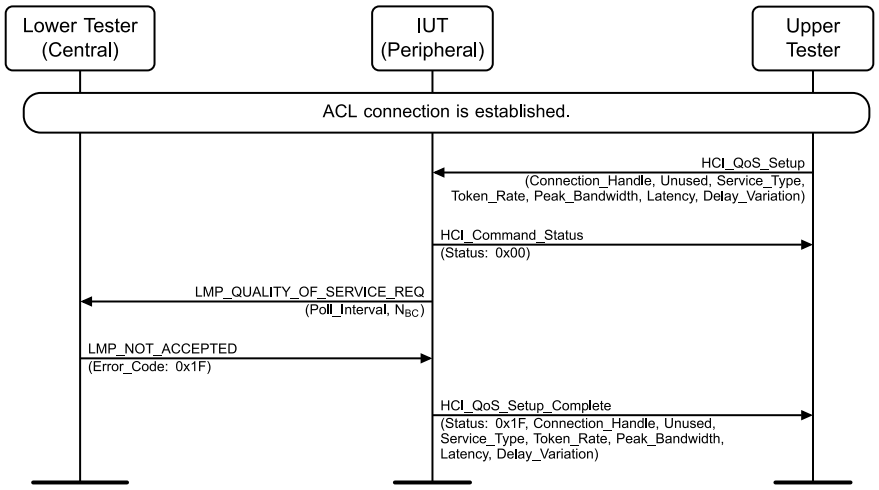


Figure 4.8-42: LMP/LIH/BV-41-C [Polling Interval Rejected] MSC

1. The Upper Tester sends an HCI_QoS_Setup command to the IUT with the Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_QUALITY_OF_SERVICE_REQ PDU to the Lower Tester with the Poll_Interval and N_{BC}.
3. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_QUALITY_OF_SERVICE_REQ PDU Opcode.
4. The IUT sends an HCI_QoS_Setup_Complete event to the Lower Tester with Status set to 0x1F (Unspecified Error), Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_QUALITY_OF_SERVICE_REQ PDU to the Lower Tester and accepts the LMP_NOT_ACCEPTED PDU sent by the Lower Tester in response.

- Notes

  There is no special HCI command for PDU LMP_QUALITY_OF_SERVICE_REQ; it is the same as for PDU LMP_QUALITY_OF_SERVICE.

4.8.14 Quality of Service (QoS) – Central

Verify that a unit can request a change of maximum polling interval. The IUT is the Central.

LMP/LIH/BV-42-C [Set Polling Interval]

- Test Purpose

  Verify that the IUT can request or notify the Lower Tester of the new polling interval. Verify on the baseband level that the time between subsequent transmissions to the Lower Tester never exceeds the POLLING interval.

- Reference

  [1] 4.1.8

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The IUT is the Central and notifies or requests the Lower Tester. The Lower Tester is the Peripheral.
  - The IUT must page the Lower Tester to become the Central of the piconet.

- Test Procedure

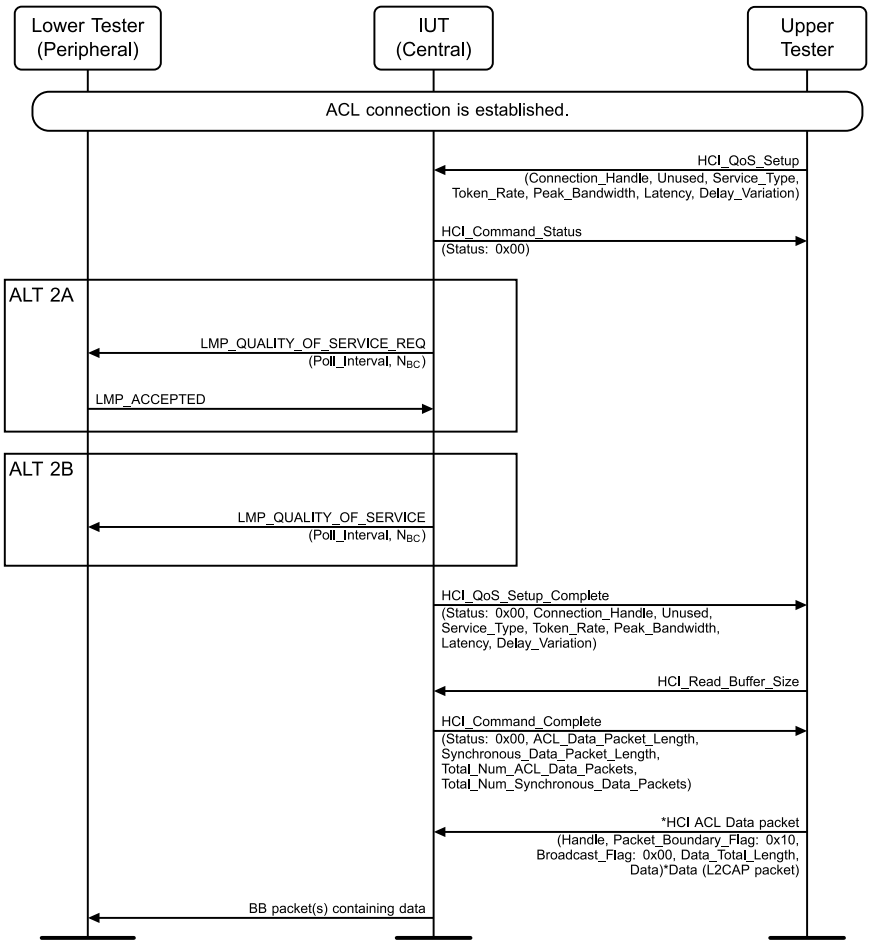


Figure 4.8-43: LMP/LIH/BV-42-C [Set Polling Interval] MSC

1. The Upper Tester sends an HCI_QoS_Setup command to the IUT with the Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation and receives a successful HCI_Command_Status event in response.
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the LMP_QUALITY_OF_SERVICE_REQ PDU to the Lower Tester):
   - 2A.1. The IUT sends an LMP_QUALITY_OF_SERVICE_REQ PDU to the Lower Tester with Poll_Interval and N_{BC}.
   - 2A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_QUALITY_OF_SERVICE_REQ PDU Opcode.

   Alternative 2B (The IUT sends the LMP_QUALITY_OF_SERVICE PDU to the Lower Tester):
   - 2B.1. The IUT sends an LMP_QUALITY_OF_SERVICE PDU to the Lower Tester with Poll_Interval and N_{BC}.
3. The IUT sends a successful HCI_QoS_Setup_Complete event to the Upper Tester with the Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.
4. The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event with ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, and Total_Num_Synchronous_Data_Packets.
5. The Upper Tester sends HCI ACL Data packets to the IUT.
6. The IUT sends BB data packets to the Lower Tester.

Verify that the IUT transmits allowed packets according to the polling interval given.

- Expected Outcome

  Pass verdict

  In alternative 2A, the IUT sends the LMP_QUALITY_OF_SERVICE_REQ PDU to the Lower Tester.

  In alternative 2B, the IUT sends the LMP_QUALITY_OF_SERVICE PDU to the Lower Tester.

  In Step 6, the IUT sends the packets according to the new maximum polling interval.

- Notes

  There is no special HCI command for the LMP_QUALITY_OF_SERVICE PDU; it is the same as the LMP_QUALITY_OF_SERVICE_REQ PDU.

4.8.15 SCO links – Peripheral

Verify that the unit can initiate and delete a SCO link. The IUT is the Peripheral.

4.8.15.1 Request or accept SCO connection

- Test Purpose

  Verify that the IUT sets up a SCO link upon request from the Lower Tester. Verify that the correct SCO setup is used.

- Reference

  [1] 4.6.1.1

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral.
  - A Features request has been carried out: see LMP/INF/BV-10-C [Supported Features Response].
- Test Case Configuration

Test Case	Packet type(s)	Initiator	T_{SCO}	SCO_Packet
LMP/LIH/BV-43-C	HV1 or DV	Lower Tester	2	0
LMP/LIH/BV-44-C	HV2	Lower Tester	4	1
LMP/LIH/BV-45-C	HV3	Lower Tester	6	2
LMP/LIH/BV-46-C	HV1 or DV	IUT	2	0

Table 4.8-3: Request or accept SCO connection – Peripheral test cases

- Test Procedure

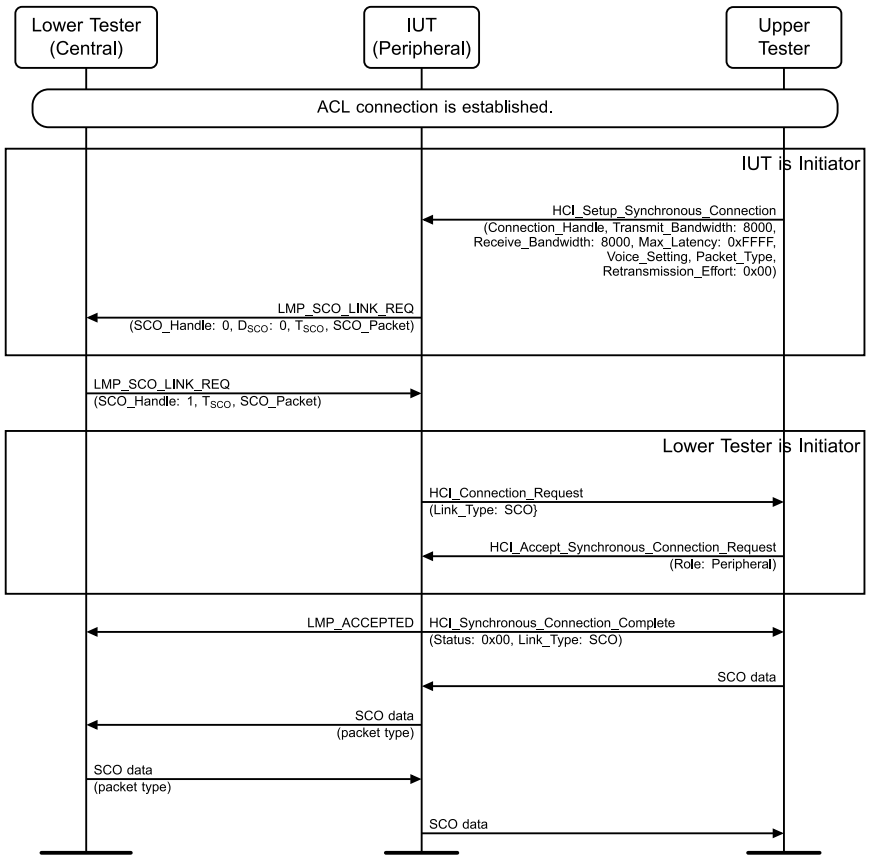


Figure 4.8-44: Request or accept SCO connection MSC

1. If the IUT is the Initiator, perform this step:
   1.1 The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with Connection_Handle set to the handle of the existing ACL connection, Transmit_Bandwidth and Receive_Bandwidth = 8000 (0x1F40), Max_Latency set to 0xFFFF, Voice_Setting set to a valid value, Retransmission_Effort = 0x00, and Packet_Type set to allow only the packet type(s) in Table 4.8-3 and receives a successful HCI_Command_Status event in reply.

1.2 The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with SCO_Handle = 0, D_{SCO} = 0, T_{SCO} and SCO_Packet set to the values in Table 4.8-3, and the other parameters set to valid values.

2. The Lower Tester sends an LMP_SCO_LINK_REQ PDU to the IUT with SCO_Handle = 1, T_{SCO} and SCO_Packet set to the values in Table 4.8-3, and the other parameters set to valid values.
3. If the Lower Tester is the Initiator, perform this step:
   3.1 The IUT sends an HCI_Connection_Request event to the Upper Tester with Link_Type set to SCO.
   3.2 The Upper Tester sends an HCI_Accept_Synchronous_Connection_Request command to the IUT with role set to Peripheral and receives a successful HCI_Command_Status event in reply.

Steps 4 and 5 may be conducted in either order.

4. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_SCO_LINK_REQ PDU Opcode.
5. The Upper Tester receives a successful HCI_Synchronous_Connection_Complete event from the IUT with Link_Type set to SCO.
6. The IUT and the Lower Tester exchange the packets specified in Table 4.8-3 containing data.

- Expected Outcome

  Pass verdict

  In Step 1.2, the IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with the specified values.

  In Step 3.1, the IUT sends an HCI_Connection_Request event to the Upper Tester with Link_Type set to SCO.

  In Step 4, the IUT sends a DM1 packet containing an LMP_ACCEPTED PDU to the Lower Tester with Opcode set to LMP_SCO_LINK_REQ.

  In Step 6, the IUT and the Lower Tester exchange the specified packets.

4.8.15.2 Accept SCO Change

- Test Purpose

  Verify that the IUT changes SCO interval and SCO_Packet type (from HV1 to the Packet Type in Table 4.8-4) upon request from the Lower Tester.

- Reference

  [1] 4.6.1.3

- Initial Condition

  - See LMP/LIH/BV-43-C, with the exception that HCI_Accept_Synchronous_Connection_Request uses a packet type of all HV packets (0x03FF).
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - A SCO link using HV1 packets has been established.
  - HCI Buffers have been checked.
  - Whether the IUT requires SCO data to be provided over HCI to cause SCO packets to be transmitted is defined by TSPX_hci_sco_data_packets_needed.

- Test Case Configuration

Test Case	Packet Type	T_{SCO}	SCO_Packet
LMP/LIH/BV-47-C [Accept Change to HV2 as Peripheral]	HV2	4	1
LMP/LIH/BV-48-C [Accept Change to HV3]	HV3	6	2

Table 4.8-4: Accept SCO Change test cases

- Test Procedure

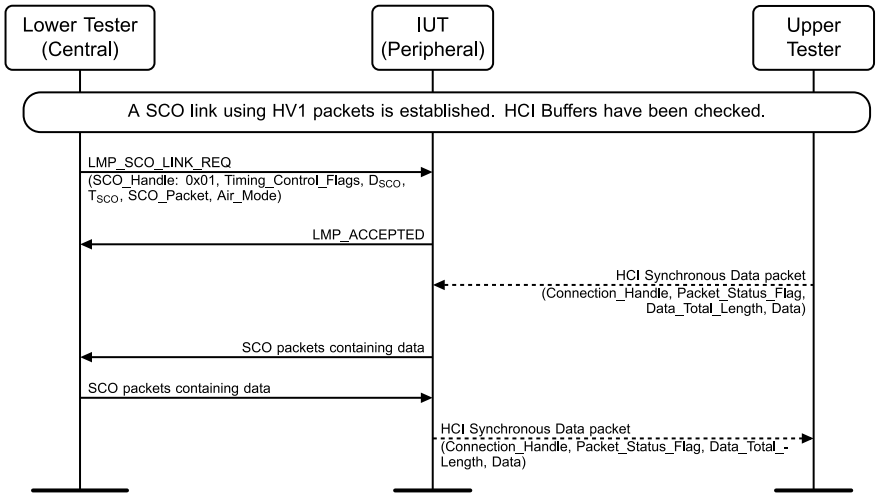


Figure 4.8-45: Accept SCO Change MSC

1. The Lower Tester sends an LMP_SCO_LINK_REQ PDU to the IUT with SCO_Handle set to 0x01, Timing_Control_Flags, D_{SCO}, T_{SCO} and SCO_Packet set to the value in Table 4.8-4, and Air_Mode.
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SCO_LINK_REQ PDU Opcode.
3. If TSPX_hci_sco_data_packets_needed is TRUE, the Upper Tester sends an HCI Synchronous Data packet to the IUT with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.
4. The IUT and the Lower Tester exchange the packets specified in Table 4.8-4 containing data.
5. Optionally, the IUT sends an HCI Synchronous Data packet to the Upper Tester with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SCO_LINK_REQ PDU from the Lower Tester.

  The SCO interval, T_{SCO}, and packet type is changed accordingly, and the packet specified in Table 4.8-4 is sent every T_{SCO} time slots.

LMP/LIH/BV-49-C [Request Change to HV2]

- Test Purpose

  Verify that the IUT can request that the Lower Tester change the SCO interval and packet type (from HV1 to HV2).

- Reference

  [1] 4.6.1.4

- Initial Condition

  - See LMP/LIH/BV-43-C.
  - The IUT is the Peripheral and initiates the service. The Lower Tester is the Central.
  - A SCO link using HV1 packets has been established.
  - HCI Buffers have been checked.
  - Whether the IUT requires SCO data to be provided over HCI to cause SCO packets to be transmitted is defined by TSPX_hci_sco_data_packets_needed.

- Test Procedure

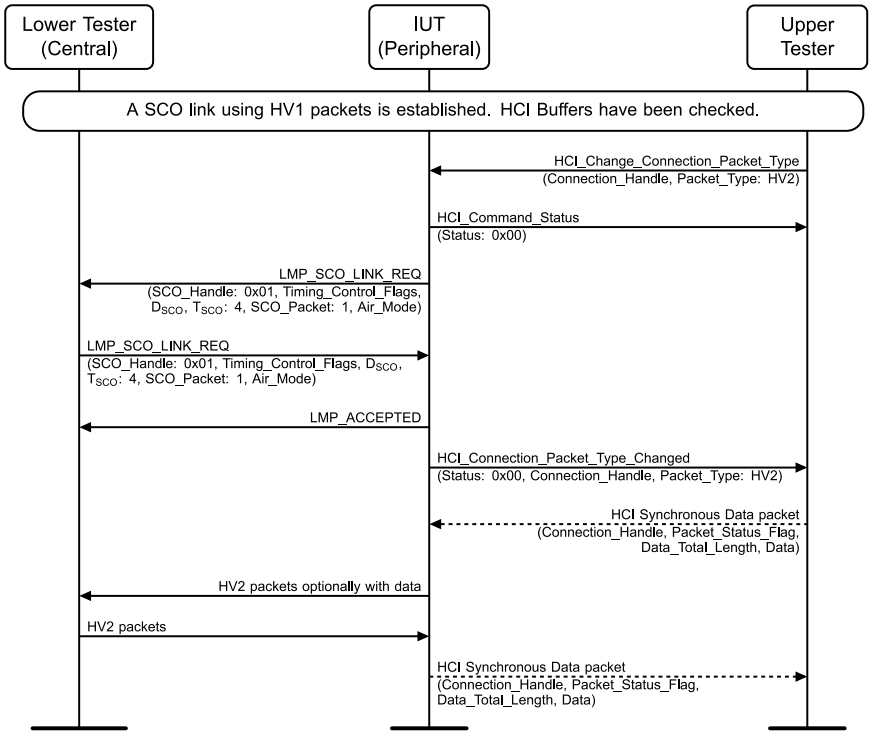


Figure 4.8-46: LMP/LIH/BV-49-C [Request Change to HV2] MSC

1. The Upper Tester sends an HCI_Change_Connection_Packet_Type command to the IUT with the Connection_Handle and Packet_Type set to HV2 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with SCO_Handle set to 0x01, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.
3. The Lower Tester responds to the IUT with an LMP_SCO_LINK_REQ PDU with SCO_Handle set to 0x01, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.

4. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_SCO_LINK_REQ PDU Opcode.
5. The IUT sends a successful HCI_Connection_Packet_Type_Changed event to the Upper Tester with the Connection_Handle and Packet_Type set to HV2.
6. If TSPX_hci_sco_data_packets_needed is TRUE, the Upper Tester sends an HCI Synchronous Data packet to the IUT with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.
7. The IUT and the Lower Tester exchange HV2 packets with or without data.
8. Optionally, the IUT sends an HCI Synchronous Data packet to the Upper Tester with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SCO_LINK_REQ PDU to the Lower Tester and the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_SCO_LINK_REQ PDU from the Lower Tester.

  The packet type used is HV2, and an HV2 packet is sent every four time slots.

LMP/LIH/BV-50-C [HV2 Request Rejected by Central]

- Test Purpose

  Verify that the IUT can request a change of the SCO interval and packet type (from HV1 to HV2) and that the IUT accepts when the Lower Tester rejects the request.

- Reference

  [1] 4.6.1.4

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Peripheral and initiates the service. The Lower Tester is the Central.
  - SCO link using HV1 packets has been established.
  - HCI Buffers have been checked.
  - Whether the IUT requires SCO data to be provided over HCI to cause SCO packets to be transmitted is defined by TSPX_hci_sco_data_packets_needed.

- Test Procedure

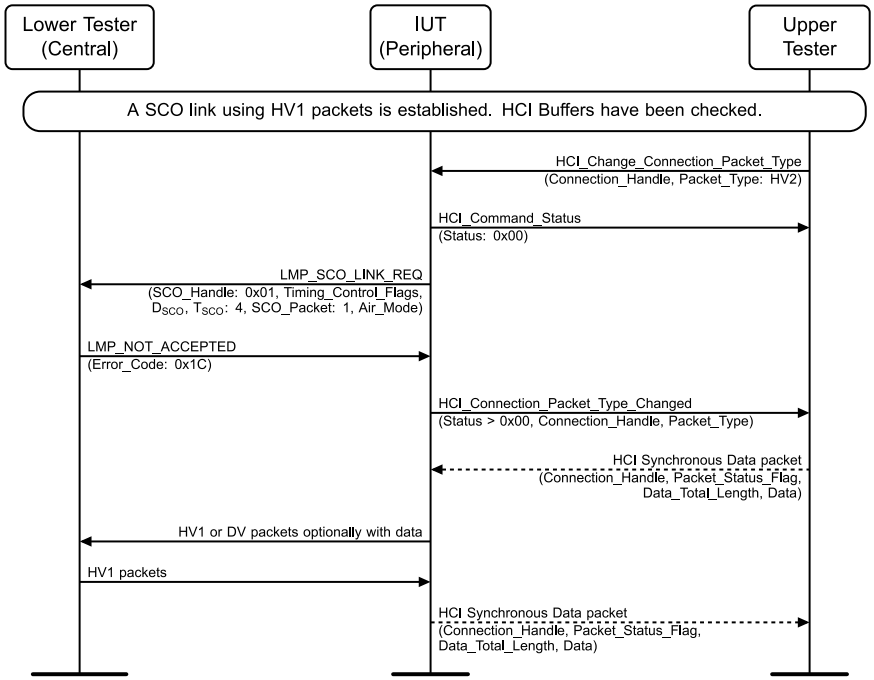


Figure 4.8-47: LMP/LIH/BV-50-C [HV2 Request Rejected by Central] MSC

1. The Upper Tester sends an HCI_Change_Connection_Packet_Type command to the IUT with the Connection_Handle and Packet_Type set to HV2 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with SCO_Handle set to 0x01, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.
3. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SCO_LINK_REQ PDU Opcode and Error_Code set to 0x1C (SCO Interval Rejected).
4. The IUT sends an HCI_Connection_Packet_Type_Changed event to the Upper Tester with Status set to a valid error code, the Connection_Handle, and Packet_Type.
5. If TSPX_hci_sco_data_packets_needed is TRUE, the Upper Tester sends an HCI Synchronous Data packet to the IUT with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.
6. The IUT and the Lower Tester exchange HV1 or DV packets with or without data.
7. Optionally, the IUT sends an HCI Synchronous Data packet to the Upper Tester with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SCO_LINK_REQ PDU to the Lower Tester and accepts the LMP_NOT_ACCEPTED PDU from the Lower Tester.

  The packet type remains HV1 or DV, and an HV1 or DV packet is sent every two time slots.

4.8.16 SCO links – Central

Verify that the unit can initiate and delete a SCO link. The IUT is the Central.

4.8.16.1 Accept or establish SCO

- Test Purpose

  Verify that the IUT can establish a SCO link.

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Central. The Lower Tester is the Peripheral.
  - A Features request has to be carried out: see LMP/INF/BV-10-C [Supported Features Response].
  - The IUT must page the Lower Tester to become the Central of the piconet.

- Test Case Configuration

Test Case	Reference	Initiator
LMP/LIH/BV-53-C	[1] 4.6.1.1	IUT
LMP/LIH/BV-54-C	[1] 4.6.1.2	Lower Tester

Table 4.8-5: Accept or establish SCO test cases

- Test Procedure

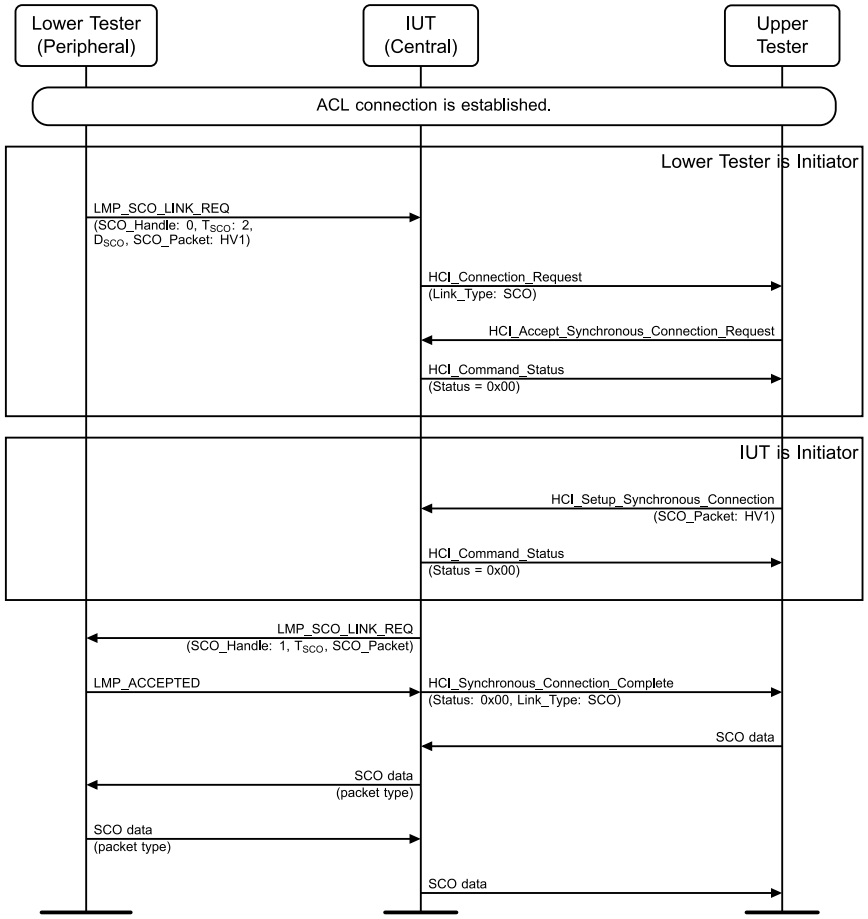


Figure 4.8-48: Accept or establish SCO MSC

1. If the Lower Tester is the Initiator, perform this step:
   1.1 The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with SCO_Handle = 0, T_{SCO} set to 2, D_{SCO}, and SCO_Packet set to 0 (HV1), and the other parameters set to valid values.
   1.2 The IUT sends an HCI_Connection_Request event to the Upper Tester with Link_Type set to SCO.
   1.3 The Upper Tester sends an HCI_Accept_Synchronous_Connection_Request to the IUT and receives a successful HCI_Command_Status in response.
2. If the IUT is the Initiator, perform this step:
   2.1 The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with Packet_Type set to HV1 and receives a successful HCI_Command_Status event in response.
3. The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with SCO_handle = 1, T_{SCO} set to 2, and SCO_Packet set to 0 (HV1), and the other parameters set to valid values.

Steps 4 and 5 may be conducted in either order.

4. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_SCO_LINK_REQ PDU Opcode.
5. The Upper Tester receives a successful HCI_Synchronous_Connection_Complete event from the IUT with Link_Type set to SCO.

6. The IUT and the Lower Tester exchange the HV1 or DV packets containing data. The Lower Tester only transmits on the slots identified by T_{SCO} and D_{SCO} in Step 3, and the IUT transmits on the following slots.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT sends an LMP_ACCEPTED PDU to the Lower Tester with Opcode set to LMP_SCO_LINK_REQ.

  In Step 6, the IUT and the Lower Tester exchange the specified HV1 packets.

LMP/LIH/BV-55-C [Request change to HV3]

- Test Purpose

  Verify that the IUT can request a change of the SCO parameters (packet type HV1 to HV3).

- Reference

  [1] 4.6.1.3

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Central. The Lower Tester is the Peripheral and initiates the service.
  - A SCO link using HV1 packets has been established.

- Test Procedure

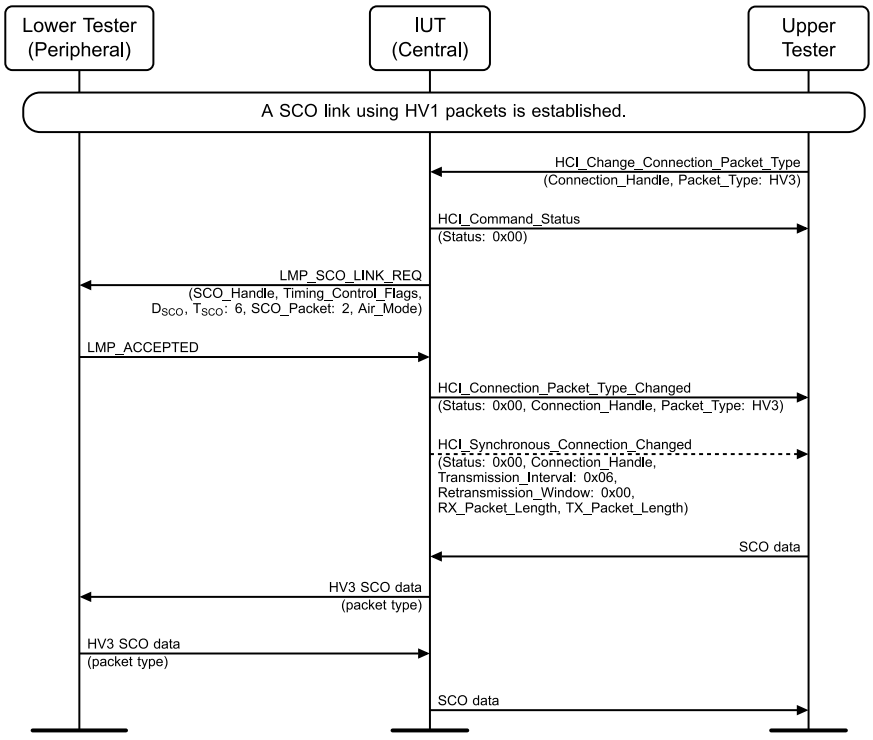


Figure 4.8-49: LMP/LIH/BV-55-C [Request change to HV3] MSC

1. The Upper Tester sends an HCI_Change_Connection_Packet_Type command to the IUT with the Connection_Handle and Packet_Type set to HV3 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with the SCO_Handle, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 6, SCO_Packet set to 2 (HV3), and Air_Mode.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SCO_LINK_REQ PDU Opcode.
4. The IUT sends a successful HCI_Connection_Packet_Type_Changed event to the Upper Tester with the Connection_Handle and Packet_Type set to HV3.
5. Optionally, the IUT sends a successful HCI_Synchronous_Connection_Changed event to the Upper Tester with the Connection_Handle, Transmission_Interval set to 0x06, Retransmission_Window set to 0x00, RX_Packet_Length, and TX_Packet_Length.
6. The Upper Tester sends SCO data to the IUT.
7. The IUT and the Lower Tester exchange HV3 data packets.
8. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SCO_LINK_REQ PDU to the Lower Tester and accepts reception of the LMP_ACCEPTED PDU.

  The SCO parameters are changed.

  The IUT and the Lower Tester exchange the specified packets. An HV3 packet is sent every six time slots.

LMP/LIH/BV-56-C [HV3 request rejected by Peripheral]

- Test Purpose

  Verify that the IUT can request a change of the SCO parameters (packet type HV1 to HV3) and that the IUT accepts a rejection from the Lower Tester.

- Reference

  [1] 4.6.1.4

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Central. The Lower Tester is the Peripheral and initiates the service.
  - A SCO link using HV1 packets has been established.

- Test Procedure

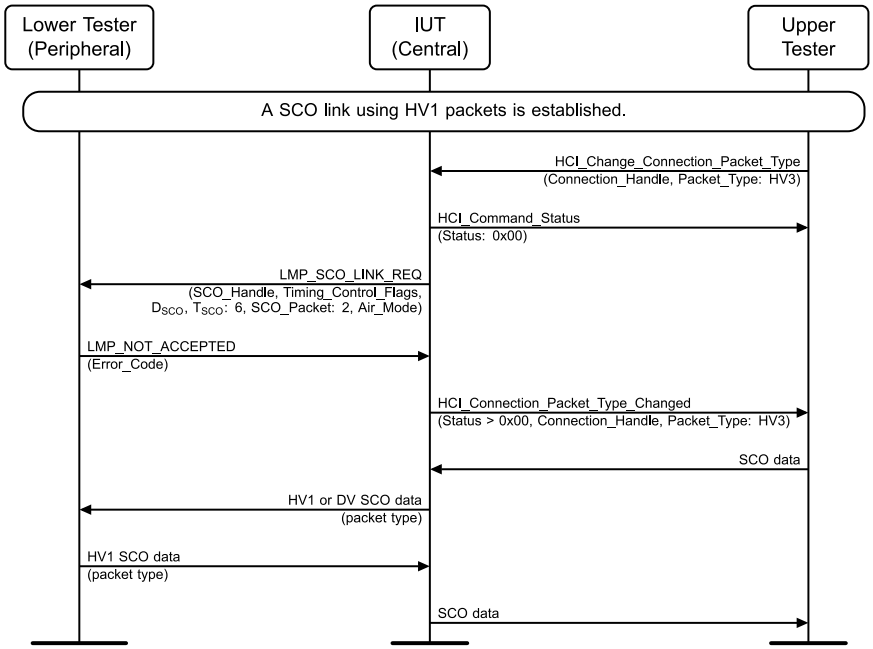


Figure 4.8-50: LMP/LIH/BV-56-C [HV3 request rejected by Peripheral] MSC

1. The Upper Tester sends an HCI_Change_Connection_Packet_Type command to the IUT with the Connection_Handle and Packet_Type set to HV3 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with the SCO_Handle, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 6, SCO_Packet set to 2 (HV3), and Air_Mode.
3. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SCO_LINK_REQ PDU Opcode and a valid Error_Code.
4. The IUT sends an HCI_Connection_Packet_Type_Changed event to the Upper Tester with Status set to a valid error code, the Connection_Handle, and Packet_Type set to HV3.
5. The Upper Tester sends SCO data to the IUT.
6. The IUT and the Lower Tester exchange HV1 or DV data packets.
7. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  The SCO link is not changed.

  The IUT and the Lower Tester exchange the specified HV1 or DV packets. An HV1 or DV packet is sent every two time slots.

LMP/LIH/BV-57-C [Accept change to HV2 as Central]

- Test Purpose

  Verify that the IUT accepts a request from the Lower Tester to change the SCO parameters (packet type HV1 to HV2). Also verify that the timing control flags bits 0 and 2 in the LMP_SCO_LINK_REQ PDU sent by the Lower Tester are ignored by the IUT.

- Reference

  [1] 4.6.1.4

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Central. The Lower Tester is the Peripheral and initiates the service.
  - A SCO link using HV1 packets has been established.
  - Flow Control is off.
- Test Procedure

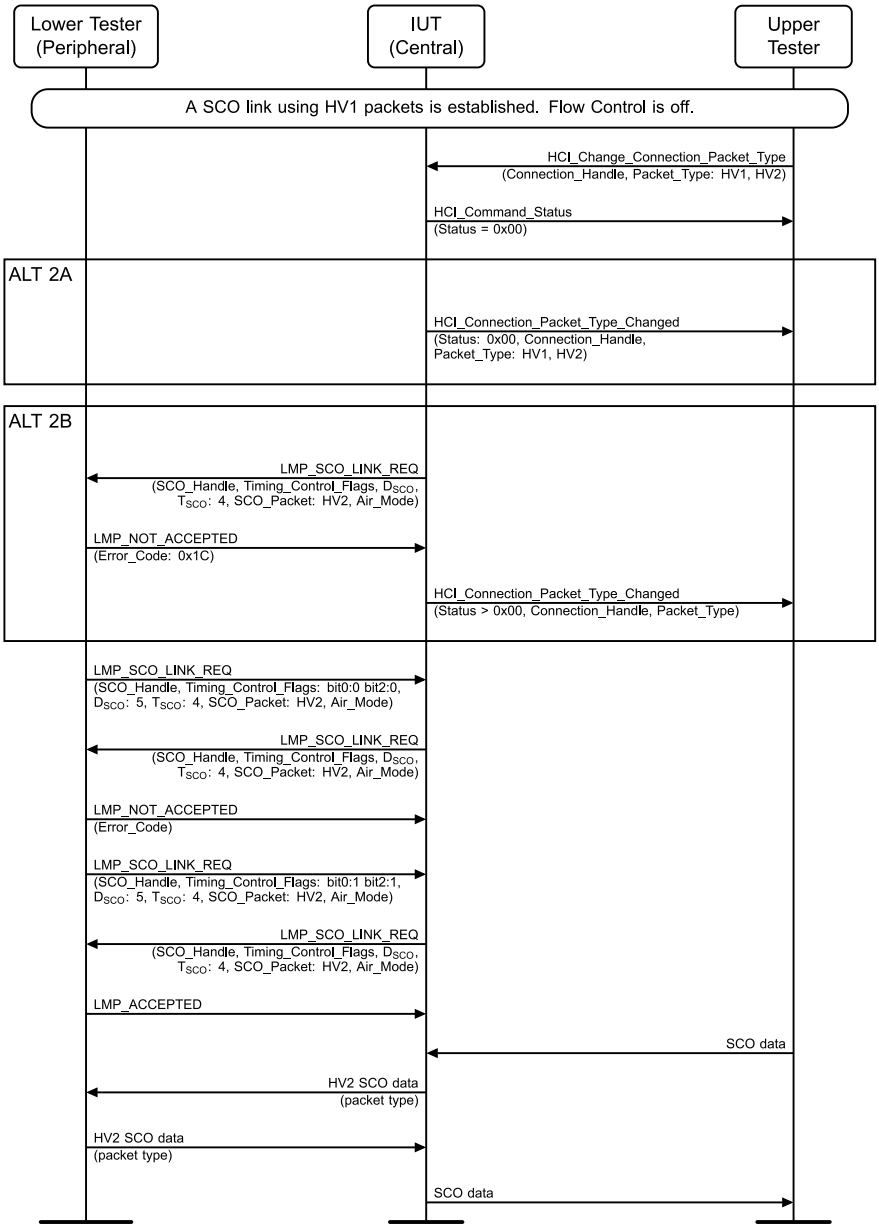


Figure 4.8-51: LMP/LIH/BV-57-C [Accept change to HV2 as Central] MSC

1. The Upper Tester sends an HCI_Change_Connection_Packet_Type command to the IUT with the Connection_Handle and Packet_Type set to HV1 and HV2 and receives a successful HCI_Command_Status event in response.

2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the HCI_Connection_Packet_Type_Changed event to the Upper Tester):
   - 2A.1. The IUT sends a successful HCI_Connection_Packet_Type_Changed event to the Upper Tester with the Connection_Handle and Packet_Type set to HV1 and HV2.

   Alternative 2B (The IUT sends the LMP_SCO_LINK_REQ PDU to the Lower Tester):
   - 2B.1. The IUT sends an LMP_SCO_LINK_REQ PDU to the Lower Tester with the SCO_Handle, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.
   - 2B.2. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SCO_LINK_REQ PDU Opcode and Error_Code set to 0x1C (SCO Interval Rejected).
   - 2B.3. The IUT sends an HCI_Connection_Packet_Type_Changed event to the Upper Tester with Status set to a valid error code, the Connection_Handle, and Packet_Type.
3. The Lower Tester sends an LMP_SCO_LINK_REQ PDU to the IUT with the SCO_Handle, Timing_Control_Flags with bit 0 and bit 2 set to 0, D_{SCO} set to 5, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.
4. The IUT responds to the Lower Tester with an LMP_SCO_LINK_REQ PDU with the SCO_Handle, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.
5. The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT with the LMP_SCO_LINK_REQ PDU Opcode and a valid Error_Code.
6. The Lower Tester sends an LMP_SCO_LINK_REQ PDU to the IUT with the SCO_Handle, Timing_Control_Flags with bit 0 and bit 2 set to 1, D_{SCO} set to 5, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.
7. The IUT responds to the Lower Tester with an LMP_SCO_LINK_REQ PDU with the SCO_Handle, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 4, SCO_Packet set to 1 (HV2), and Air_Mode.
8. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_SCO_LINK_REQ PDU Opcode.
9. The Upper Tester sends SCO data to the IUT.
10. The IUT and the Lower Tester exchange HV2 data packets.
11. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_SCO_LINK_REQ PDU to the Lower Tester upon reception of the LMP_SCO_LINK_REQ PDU from the Lower Tester and accepts reception of the LMP_ACCEPTED PDU from the Lower Tester.

  The SCO interval is changed.

  The IUT and the Lower Tester exchange the specified HV2 packets, and an HV2 packet is sent every four time slots.

4.8.17 SCO links – Both Central and Peripheral

Verify that the IUT declines the SCO link request in a correct manner. The role of the IUT is of no importance.

LMP/LIH/BV-60-C [Reject SCO Request]

- Test Purpose

  Verify that the IUT responds that it does not support SCO links upon request from the Lower Tester.

- Reference

  [1] 4.6.1.1

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.
  - The Lower Tester initiates the service.

- Test Procedure

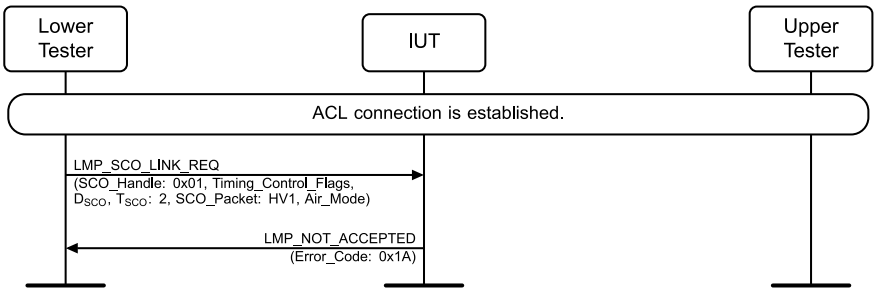


Figure 4.8-52: LMP/LIH/BV-60-C [Reject SCO Request] MSC

1. The Lower Tester sends an LMP_SCO_LINK_REQ PDU to the IUT with SCO_Handle set to 0x01, Timing_Control_Flags, D_{SCO}, T_{SCO} set to 2, SCO_Packet set to 0 (HV1), and Air_Mode.
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SCO_LINK_REQ PDU Opcode and Error_Code set to 0x1A (Unsupported Remote Feature).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1A (Unsupported Remote Feature) upon reception of the LMP_SCO_LINK_REQ PDU from the Lower Tester.

4.8.17.1 Rejecting SCO Connection request when AES-CCM encryption is enabled

- Test Purpose

  Verify that if AES-CCM encryption has been enabled on an ACL connection, SCO connection requests from the Lower Tester will be rejected by the IUT with error code 0x0E (Connection Rejected Due to Security Reasons).

- Reference

  [1] 4.6.1

- Initial Condition
  - The IUT is in the role specified in Table 4.8-6.
  - An AES-CCM encrypted point-to-point connection has been established between the IUT and the Lower Tester.
- Test Case Configuration

Test Case	Role
LMP/LIH/BI-01-C [Rejecting SCO Connection request when AES-CCM encryption is enabled, Peripheral]	Peripheral
LMP/LIH/BI-02-C [Rejecting SCO Connection request when AES-CCM encryption is enabled, Central]	Central

Table 4.8-6: Rejecting SCO Connection request when AES-CCM encryption is enabled test cases

- Test Procedure

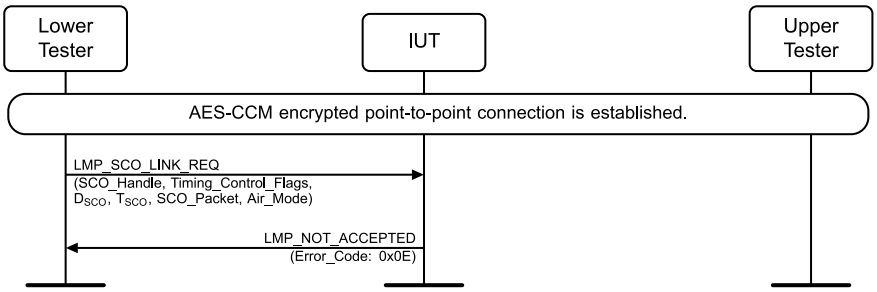


Figure 4.8-53: Rejecting SCO Connection request when AES-CCM encryption is enabled MSC

1. The Lower Tester sends an LMP_SCO_LINK_REQ PDU to the IUT with valid values.
2. The IUT responds with an LMP_NOT_ACCEPTED PDU with the LMP_SCO_LINK_REQ PDU Opcode and Error_Code set to 0x0E (Connection Rejected Due to Security Reasons).

- Expected Outcome

  Pass verdict

  The IUT responds to the LMP_SCO_LINK_REQ PDU from the Lower Tester with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x0E (Connection Rejected Due to Security Reasons).

4.8.17.2 Accept SCO Closure

- Test Purpose

  Verify that the IUT accepts a request from the Lower Tester to remove the SCO link.

- Reference

  [1] 4.6.1.5

- Initial Condition
  - See the “Default settings” section.
  - The IUT is in the role described in Table 4.8-7.
  - A SCO link using HV1 packets has been established.

- Test Case Configuration

Test Case	Role
LMP/LIH/BV-51-C [Accept SCO Closure as Peripheral]	Peripheral
LMP/LIH/BV-59-C [Accept SCO Closure as Central]	Central

Table 4.8-7: Accept SCO Closure test cases

- Test Procedure

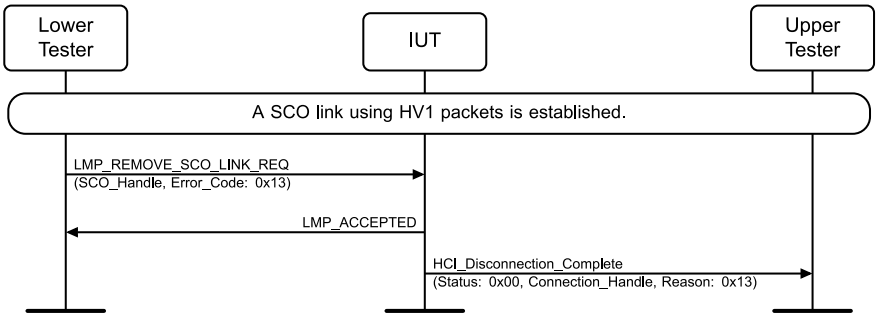


Figure 4.8-54: Accept SCO Closure MSC

1. The Lower Tester sends an LMP_REMOVE_SCO_LINK_REQ PDU to the IUT with the SCO_Handle and Error_Code set to 0x13 (Remote User Terminated Connection).
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_REMOVE_SCO_LINK_REQ PDU Opcode.
3. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_REMOVE_SCO_LINK_REQ PDU from the Lower Tester.

  The SCO link is removed.

4.8.17.3 Request SCO Closure

- Test Purpose

  Verify that the IUT can request that the Lower Tester remove the SCO link.

- Reference

  [1] 4.6.1.5

- Initial Condition

  - See the “Default settings” section.
  - The IUT is in the role described in Table 4.8-8.
  - A SCO link using HV1 packets has been established.

- Test Case Configuration

Test Case	Role
LMP/LIH/BV-52-C [Request SCO Closure as Peripheral]	Peripheral
LMP/LIH/BV-58-C [Request SCO Closure as Central]	Central

Table 4.8-8: Request SCO Closure test cases

- Test Procedure

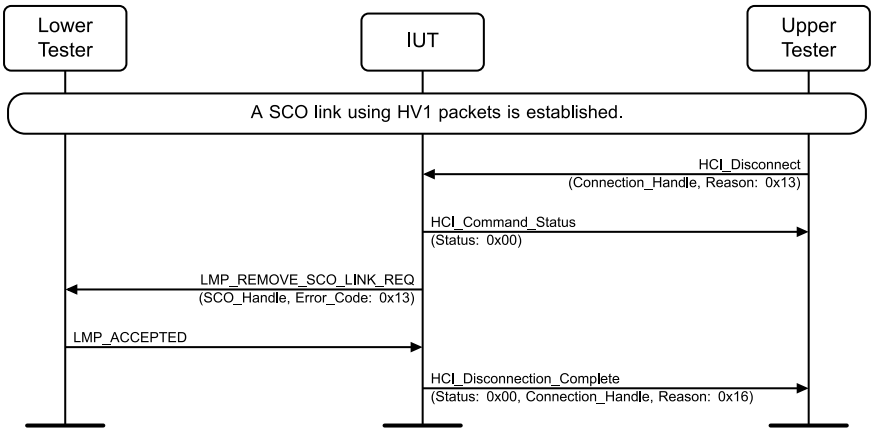


Figure 4.8-55: Request SCO Closure MSC

1. The Upper Tester sends an HCI_Disconnect command to the IUT with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection) and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_REMOVE_SCO_LINK_REQ PDU to the Lower Tester with the SCO_Handle and Error_Code set to 0x13 (Remote User Terminated Connection).
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_REMOVE_SCO_LINK_REQ PDU Opcode.
4. The IUT sends an HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x16 (Connection Terminated by Local Host).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_REMOVE_SCO_LINK_REQ PDU to the Lower Tester and accepts reception of the LMP_ACCEPTED PDU from the Lower Tester.

  The SCO link is removed.

4.8.18 eSCO links – Peripheral

4.8.18.1 Accept eSCO request

- Test Purpose

  Verify that the IUT sets up an eSCO link upon request from the Lower Tester. Verify that the correct eSCO setup is used. The EV type specified in Table 4.8-9 with data or NULL packets is used.

- Reference

  [1] 4.6.2.1

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - An ACL connection has been established.
  - A Features request has to be carried out.
- Test Case Configuration

Test Case	Reference	EV	T_{eSCO}	W_{eSCO}	Packet Length	Air Mode
LMP/LIH/BV-100-C [Accept EV3 eSCO request]	[1] 4.6.2.1	EV3	6	2	30	Any supported
LMP/LIH/BV-101-C [Accept EV4 eSCO request]	[1] 4.6.2.2	EV4	16	6	80	Transparent
LMP/LIH/BV-102-C [Accept EV5 eSCO request]	[1] 4.6.2.1	EV5	16	6	80	Transparent

Table 4.8-9: Accept eSCO request test cases

- Test Procedure

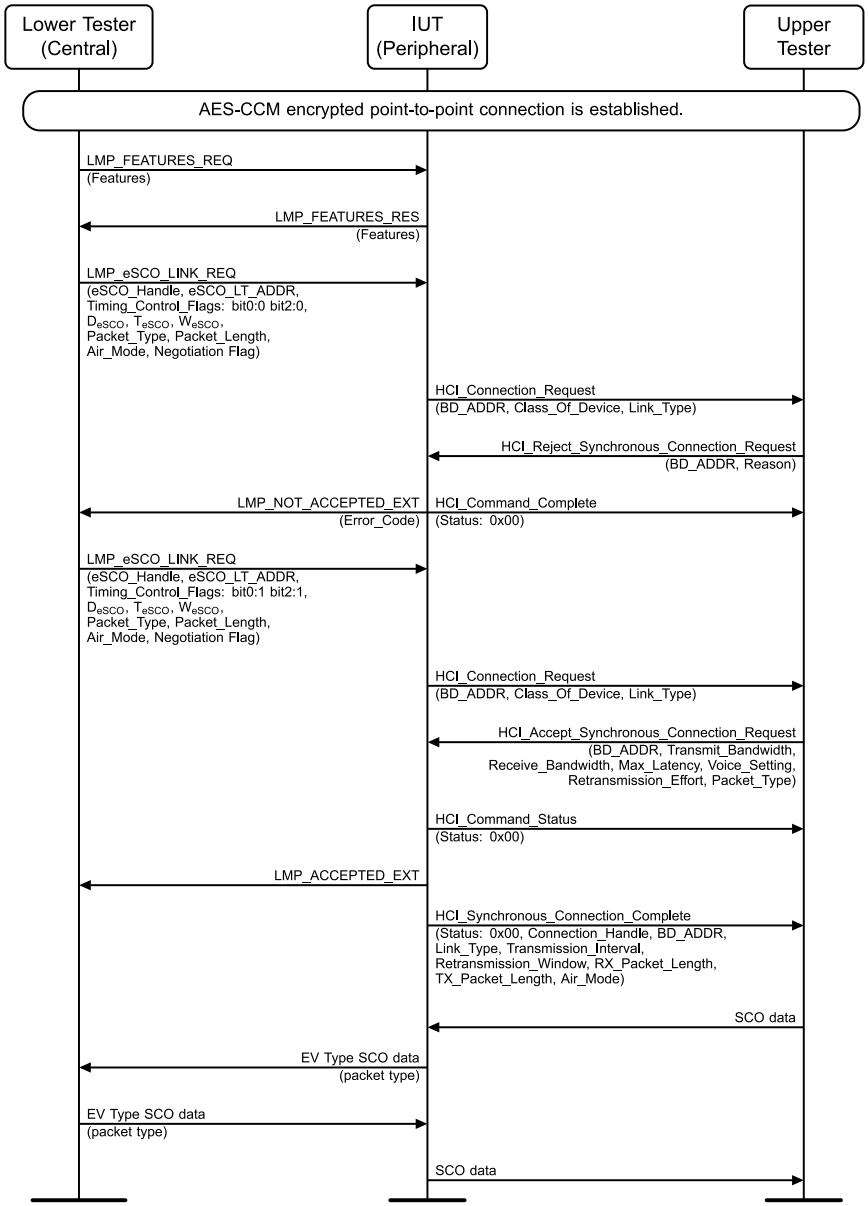


Figure 4.8-56: Accept eSCO request MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The Lower Tester sends an LMP_eSCO_LINK_REQ PDU to the IUT with:
   - eSCO_Handle and eSCO_LT_ADDR set to any valid number
   - Timing_Control_Flags derived from the Central's clock with bit 0 and bit 2 set to 0
   - D_{eSCO} set to any number in the range [0, T_{eSCO} – 2]
   - T_{eSCO}, W_{eSCO}, eSCO_Packet_Type C→P, eSCO_Packet_Type P→C, Packet_Length C→P, Packet_Length P→C, and Air_Mode set to the values specified in Table 4.8-9
   - Negotiation_State set to 0 (Initiate negotiation)
4. The IUT sends an HCI_Connection_Request event to the Upper Tester with the BD_ADDR, Class_Of_Device, and Link_Type.

5. The Upper Tester responds to the IUT with an HCI_Reject_Synchronous_Connection_Request command with the BD_ADDR and Reason.

Steps 6 and 7 can occur in either order.

6. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
7. The IUT sends an LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with the LMP_eSCO_LINK_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code.
8. The Lower Tester sends an LMP_eSCO_LINK_REQ PDU to the IUT with:
   - eSCO_Handle and eSCO_LT_ADDR set to any valid number
   - Timing_Control_Flags derived from the Central's clock with bit 0 and bit 2 set to 1
   - D_{eSCO} set to any number in the range [0, T_{eSCO} – 2]
   - T_{eSCO}, W_{eSCO}, eSCO_Packet_Type C→P, eSCO_Packet_Type P→C, Packet_Length C→P, Packet_Length P→C, and Air_Mode set to the values specified in Table 4.8-9
   - Negotiation_State set to 0 (Initiate negotiation)
9. The IUT sends an HCI_Connection_Request event to the Upper Tester with the BD_ADDR, Class_Of_Device, and Link_Type.
10. The Upper Tester responds to the IUT with an HCI_Accept_Synchronous_Connection_Request command with the BD_ADDR, Transmit_Bandwidth, Receive_Bandwidth, Max_Latency, Voice_Setting, Retransmission_Effort, and Packet_Type and receives a successful HCI_Command_Status event in response.
11. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
12. The IUT sends a successful HCI_Synchronous_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
13. The Upper Tester sends SCO data to the IUT.
14. The IUT and the Lower Tester exchange EV Type SCO data packets.
15. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester upon reception of the LMP_eSCO_LINK_REQ PDU from the Lower Tester. An eSCO link is established accordingly.

  The IUT and the Lower Tester exchange the specified EV type in Table 4.8-9 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

- Notes

  The IUT may negotiate the eSCO parameters.

LMP/LIH/BV-103-C [Request eSCO as Peripheral]

- Test Purpose

  Verify that the IUT can request that the Lower Tester set up an eSCO link and that the correct eSCO setup is used.

- Reference

  [1] 4.6.2.2

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral and initiates the service. The Lower Tester is the Central.
  - An ACL connection has been established.
  - A Features request has to be carried out.
- Test Procedure

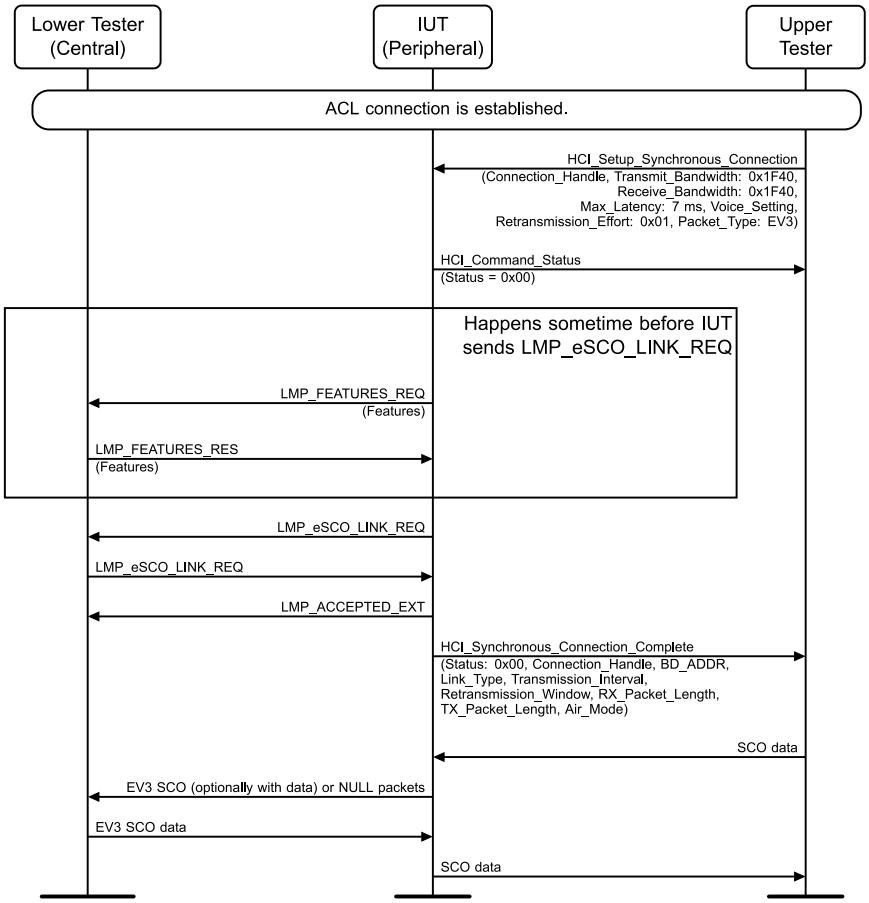


Figure 4.8-57: LMP/LIH/BV-103-C [Request eSCO as Peripheral] MSC

1. The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with the Connection_Handle, Transmit_Bandwidth and Receive_Bandwidth set to 0x1F40 (8000), Max_Latency set to 7 ms, Voice_Setting, Retransmission_Effort set to 0x01, and Packet_Type set to EV3 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
3. The Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with the Features parameter.
4. The IUT sends an LMP_eSCO_LINK_REQ PDU to the Lower Tester with valid values.
5. The Lower Tester responds to the IUT with an LMP_eSCO_LINK_REQ PDU with valid values.
6. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.

7. The IUT sends a successful HCI_Synchronous_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
8. The Upper Tester sends SCO data to the IUT.
9. The IUT and the Lower Tester exchange EV3 SCO or NULL packets.
10. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_eSCO_LINK_REQ PDU to the Lower Tester with parameters that satisfy the bandwidth and latency requirements.

  An eSCO link is established accordingly.

  In Step 9, the IUT and the Lower Tester exchange the specified EV3 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

- Notes

  The choice of packet type and Packet_Length is up to the IUT.

4.8.18.2 Accept change from EV3 to higher length packet

- Test Purpose

  Verify that the IUT changes eSCO interval and eSCO_Packet_Type from an EV3 packet to a higher payload length packet upon request from the Lower Tester.

- Reference

  [1] 4.6.2.3

- Initial Condition

  - See Accept eSCO request, with the exception that the HCI_Accept_Synchronous_Connection_Request uses a packet type of all EV packets (0x38).
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - A SCO link using EV3 packets has been established.

- Test Case Configuration

Test Case	Packet type
LMP/LIH/BV-104-C [Accept change from EV3 to higher length packet, EV4]	EV4
LMP/LIH/BV-105-C [Accept change from EV3 to higher length packet, EV5]	EV5

Table 4.8-10: Accept change from EV3 to higher length packet test cases

- Test Procedure

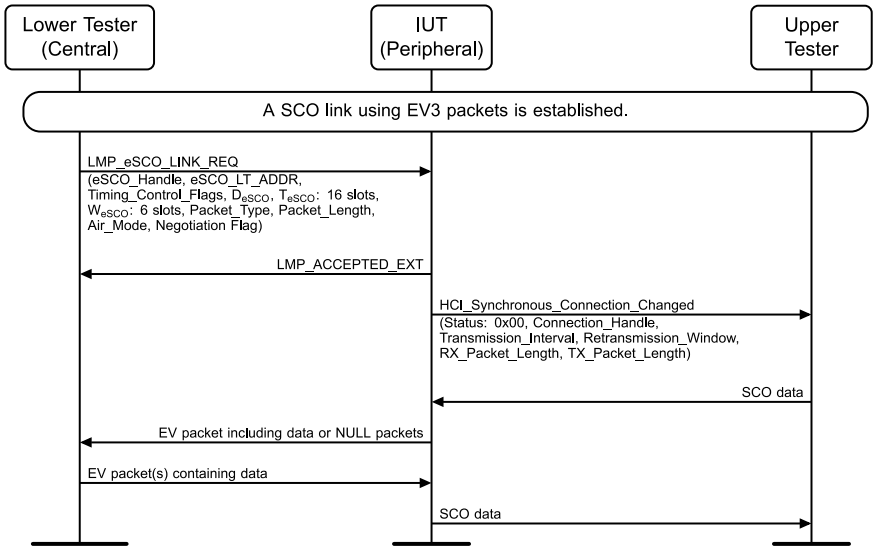


Figure 4.8-58: Accept change from EV3 to higher length packet MSC

1. The Lower Tester sends an LMP_eSCO_LINK_REQ PDU to the IUT with:
   - eSCO_Handle: The current handle of the eSCO link
   - eSCO_LT_ADDR: The current LT_ADDR of the eSCO link
   - Timing_Control_Flags: Derived from Central's clock
   - D_{eSCO}: Any number in the range [0, T_{eSCO} – 2]
   - T_{eSCO}: 16 slots
   - W_{eSCO}: 6 slots
   - Packet type C→P: Packet type specified in Table 4.8-10
   - Packet type P→C: Packet type specified in Table 4.8-10
   - Packet_Length C→P: 80 bytes
   - Packet_Length P→C: 80 bytes
   - Air_Mode: The current Air_Mode of the eSCO link
   - Negotiation Flag: Initiate Negotiation
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
3. The IUT sends a successful HCI_Synchronous_Connection_Changed event to the Upper Tester with the Connection_Handle, Transmission_Interval, Retransmission_Window, RX_Packet_Length, and TX_Packet_Length.
4. The Upper Tester sends SCO data to the IUT.
5. The IUT and the Lower Tester exchange the packets specified in Table 4.8-10. The Lower Tester only transmits on the slots identified by T_{SCO} and D_{SCO} in Step 1, and the IUT transmits on the following slots.
6. The IUT sends SCO data to the Upper Tester

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends an LMP_ACCEPTED PDU to the Lower Tester.

  In Step 5, the IUT and the Lower Tester exchange the specified packets for the EV type specified in Table 4.8-10.

- Notes

  The IUT may negotiate the packet type specified in Table 4.8-10 eSCO parameters.

LMP/LIH/BV-106-C [Request change to EV4]

- Test Purpose

  Verify that the IUT can request that the Lower Tester change the eSCO interval and packet type (from EV3 to EV4).

- Reference

  [1] 4.6.2.3

- Initial Condition

  - See Accept eSCO request, with the exception that the HCI_Accept_Synchronous_Connection_Request uses a packet type of all EV packets (0x38).
  - The IUT is the Peripheral and initiates the service. The Lower Tester is the Central.
  - A SCO link using EV3 packets has been established.

- Test Procedure

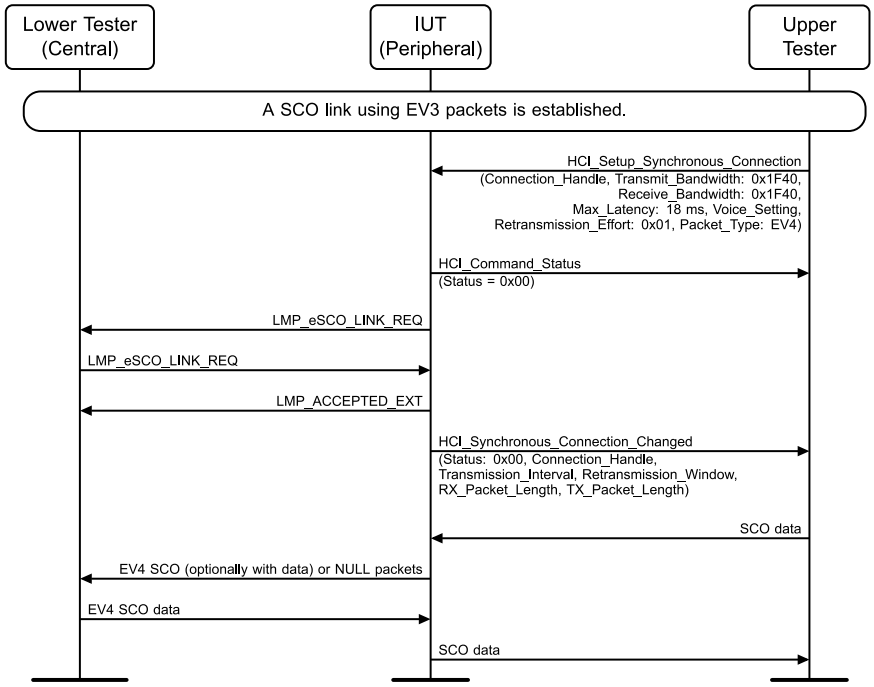


Figure 4.8-59: LMP/LIH/BV-106-C [Request change to EV4] MSC

1. The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with the Connection_Handle, Transmit_Bandwidth and Receive_Bandwidth set to 0x1F40 (8000), Max_Latency set to 18 ms, Voice_Setting, Retransmission_Effort set to 0x01, and Packet_Type set to EV4 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_eSCO_LINK_REQ PDU to the Lower Tester with valid values.
3. The Lower Tester responds to the IUT with an LMP_eSCO_LINK_REQ PDU with valid values.
4. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.

5. The IUT sends a successful HCI_Synchronous_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
6. The Upper Tester sends SCO data to the IUT.
7. The IUT and the Lower Tester exchange EV4 SCO or NULL packets.
8. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_eSCO_LINK_REQ PDU to the Lower Tester.

  After sending the LMP_ACCEPTED_EXT PDU to the Lower Tester in Step 4, the interval and packet type are changed accordingly, and data is transferred after the change.

  In Step 7, the IUT and the Lower Tester exchange the specified EV4 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

LMP/LIH/BV-107-C [EV4 request rejected]

- Test Purpose

  Verify that the IUT can request that the Lower Tester change the eSCO interval and packet type (from EV3 to EV4). Verify that the IUT accepts that the Lower Tester rejects the request.

- Reference

  [1] 4.6.2.3

- Initial Condition

  - See Accept eSCO request.
  - The IUT is the Peripheral and initiates the service. The Lower Tester is the Central.
  - A SCO link using EV3 packets has been established.

- Test Procedure

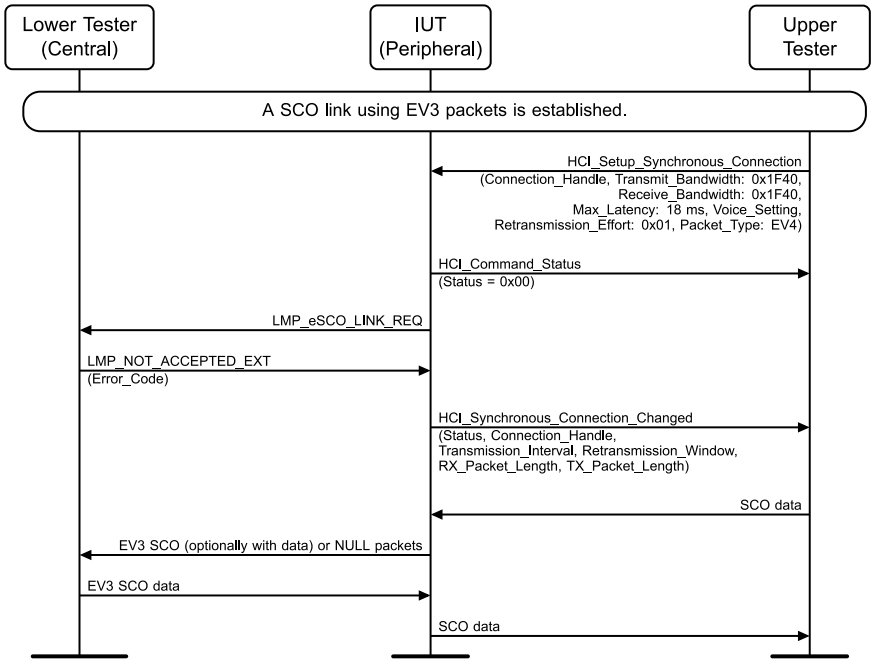


Figure 4.8-60: LMP/LIH/BV-107-C [EV4 request rejected] MSC

1. The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with the Connection_Handle, Transmit_Bandwidth and Receive_Bandwidth set to 0x1F40 (8000), Max_Latency set to 18 ms, Voice_Setting, Retransmission_Effort set to 0x01, and Packet_Type set to EV4 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_eSCO_LINK_REQ PDU to the Lower Tester with valid values.
3. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_eSCO_LINK_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code.
4. The IUT sends an HCI_Synchronous_Connection_Changed event to the Upper Tester with the Status, Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
5. The Upper Tester sends SCO data to the IUT.
6. The IUT and the Lower Tester exchange EV3 SCO or NULL packets.
7. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_eSCO_LINK_REQ PDU to the Lower Tester.

  After reception of the LMP_NOT_ACCEPTED_EXT PDU from the Lower Tester in Step 3, data is still transferred according to the previous configuration.

  In Step 6, the IUT and the Lower Tester exchange the specified EV3 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

LMP/LIH/BV-108-C [Accept eSCO Closure as Peripheral]

- Test Purpose

  Verify that the IUT accepts a request from the Lower Tester to remove the eSCO link.

- Reference

  [1] 4.6.2.4

- Initial Condition

  - See Accept eSCO request.
  - The IUT is the Peripheral. The Lower Tester is the Central.
  - A SCO link using EV3 packets has been established.

- Test Procedure

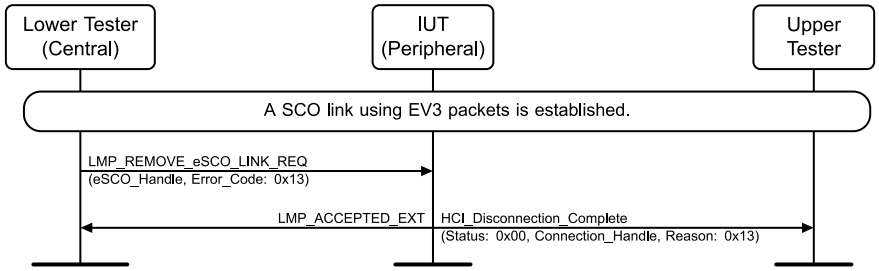


Figure 4.8-61: LMP/LIH/BV-108-C [Accept eSCO Closure as Peripheral] MSC

1. The Lower Tester sends an LMP_REMOVE_eSCO_LINK_REQ PDU to the IUT with the eSCO_Handle set to the handle of the current eSCO link and Error_Code set to 0x13 (Remote User Terminated Connection).

Steps 2 and 3 can occur in either order.

2. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection).
3. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_REMOVE_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester upon reception of the LMP_REMOVE_eSCO_LINK_REQ PDU from the Lower Tester.

  The eSCO link is removed.

LMP/LIH/BV-109-C [Request eSCO Closure as Peripheral]

- Test Purpose

  Verify that the IUT can request that the Lower Tester remove the eSCO link.

- Reference

  [1] 4.6.2.4

- Initial Condition
  - See Accept eSCO request.
  - The IUT is the Peripheral and initiates the service. The Lower Tester is the Central.
  - A SCO link using EV3 packets has been established.
- Test Procedure

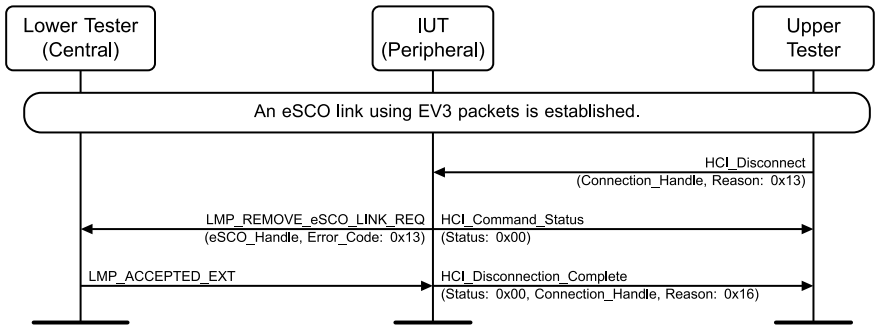


Figure 4.8-62: LMP/LIH/BV-109-C [Request eSCO Closure as Peripheral] MSC

1. The Upper Tester sends an HCI_Disconnect command to the IUT with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection).

Steps 2 and 3 can be sent in any order.

2. The IUT sends a successful HCI_Command_Status event to the Upper Tester.
3. The IUT sends an LMP_REMOVE_eSCO_LINK_REQ PDU to the Lower Tester with the eSCO_Handle set to the handle of the current eSCO link and Error_Code set to 0x13 (Remote User Terminated Connection).

Steps 4 and 5 can be sent in any order.

4. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x16 (Connection Terminated By Local Host).
5. The Lower Tester sends an LMP_ACCEPTED_EXT PDU to the IUT with the LMP_REMOVE_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_REMOVE_eSCO_LINK_REQ PDU to the Lower Tester and accepts reception of the LMP_ACCEPTED_EXT PDU from the Lower Tester.

  The eSCO link is removed.

4.8.19 eSCO links – Central

LMP/LIH/BV-110-C [Request eSCO as Central]

- Test Purpose

  Verify that the IUT can establish an eSCO link.

- Reference

  [1] 4.6.2.1

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Central and initiates the service. The Lower Tester is the Peripheral.
  - An ACL connection has been established.
- Test Procedure

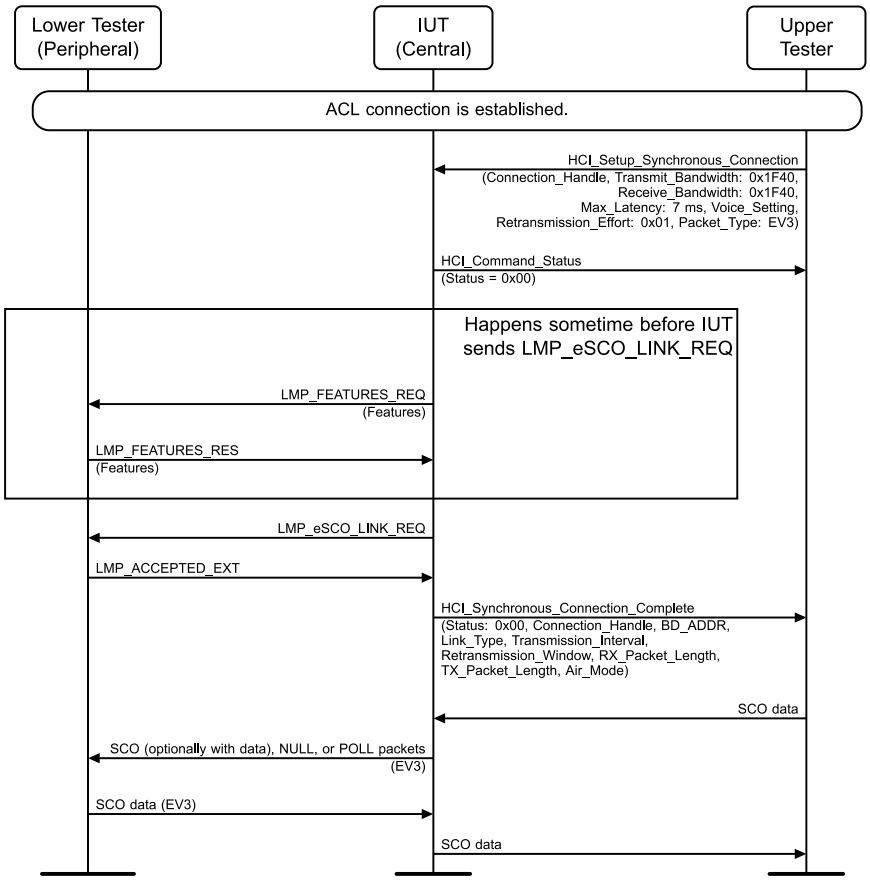


Figure 4.8-63: LMP/LIH/BV-110-C [Request eSCO as Central] MSC

1. The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with the Connection_Handle, Transmit_Bandwidth and Receive_Bandwidth set to 0x1F40 (8000), Max_Latency set to 7 ms, Voice_Setting, Retransmission_Effort set to 0x01, and Packet_Type set to EV3 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
3. The Lower Tester responds to the IUT with an LMP_FEATURES_RES PDU with the Features parameter.
4. The IUT sends an LMP_eSCO_LINK_REQ PDU to the Lower Tester with valid values.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
6. The IUT sends a successful HCI_Synchronous_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
7. The Upper Tester sends SCO data to the IUT.

8. The IUT and the Lower Tester exchange EV3 SCO, NULL, or POLL packets.
9. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends the LMP_eSCO_LINK_REQ PDU to the Lower Tester with parameters that satisfy the bandwidth and latency requirements.

  An eSCO link is established accordingly.

  In Step 8, the IUT and the Lower Tester exchange the specified EV3 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

LMP/LIH/BV-111-C [Accept eSCO request]

- Test Purpose

  Verify that the IUT accepts a request from the Lower Tester to initiate an eSCO link.

- Reference

  [1] 4.6.2.2

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Central. The Lower Tester is the Peripheral and initiates the service.
  - An ACL connection has been established.

- Test Procedure

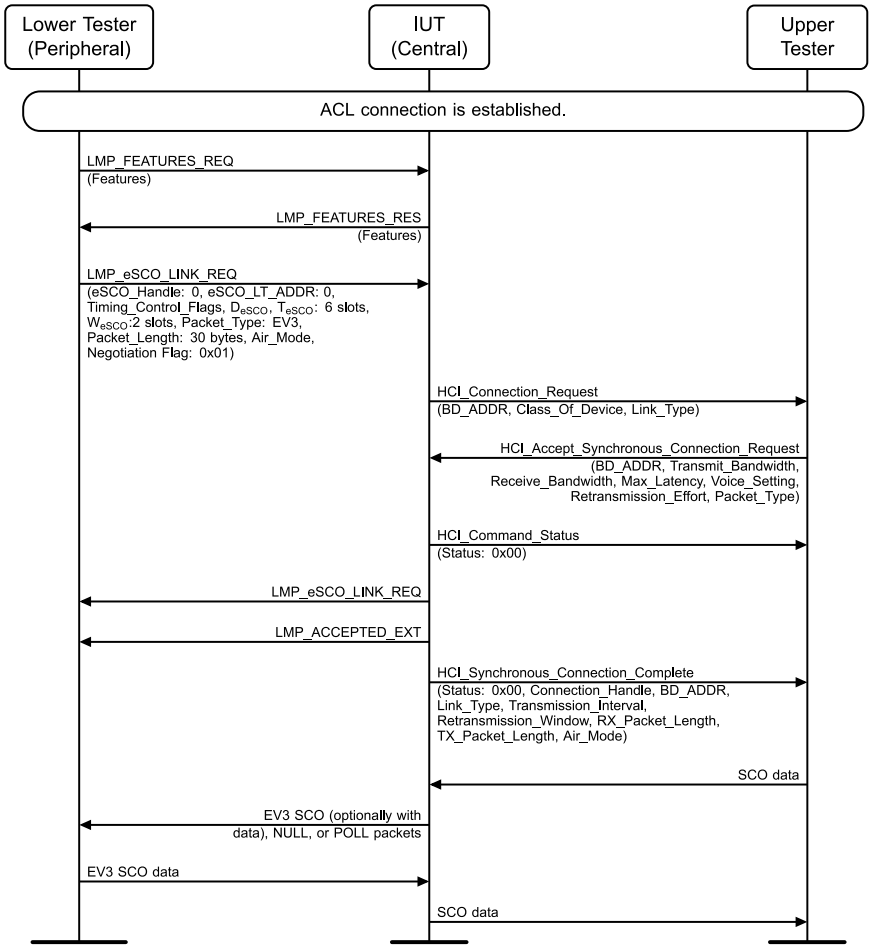


Figure 4.8-64: LMP/LIH/BV-111-C [Accept eSCO request] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The Lower Tester sends an LMP_eSCO_LINK_REQ PDU to the IUT with:
   - eSCO_Handle and eSCO_LT_ADDR set to 0
   - Timing_Control_Flags derived from IUT's Central's clock
   - D_{eSCO} set to any number in the range [0, T_{eSCO} – 2]
   - T_{eSCO} set to 6 slots
   - W_{eSCO} set to 2 slots
   - Packet type C→P and Packet type P→C set to EV3
   - Packet_Length C→P and Packet_Length P→C set to 30 bytes
   - Air_Mode set to any supported Air_Mode
   - Negotiation Flag set to 0x01 (Initiate Negotiation)
4. The IUT sends an HCI_Connection_Request event to the Upper Tester with the BD_ADDR, Class_Of_Device, and Link_Type.
5. The Upper Tester responds to the IUT with an HCI_Accept_Synchronous_Connection_Request command with the BD_ADDR, Transmit_Bandwidth, Receive_Bandwidth, Max_Latency, Voice_Setting, Retransmission_Effort, and Packet_Type and receives a successful HCI_Command_Status event in response.

6. The IUT sends an LMP_eSCO_LINK_REQ PDU to the Lower Tester with valid values.
7. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
8. The IUT sends a successful HCI_Synchronous_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
9. The Upper Tester sends SCO data to the IUT.
10. The IUT and the Lower Tester exchange EV3 SCO, NULL, or POLL packets.
11. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_eSCO_LINK_REQ PDU to the Lower Tester upon reception of the LMP_eSCO_LINK_REQ PDU from the Lower Tester.

  An eSCO link is established accordingly.

  The IUT and the Lower Tester exchange the specified EV3 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

LMP/LIH/BV-112-C [Request eSCO change]

- Test Purpose

  Verify that the IUT can request a change of the eSCO parameters (packet type EV3 to EV4 or EV5).

- Reference

  [1] 4.6.2.3

- Initial Condition

  - See LMP/LIH/BV-110-C [Request eSCO as Central].
  - The IUT is the Central and initiates the service. The Lower Tester is the Peripheral.
  - A SCO link using EV3 packets has been established.

- Test Procedure

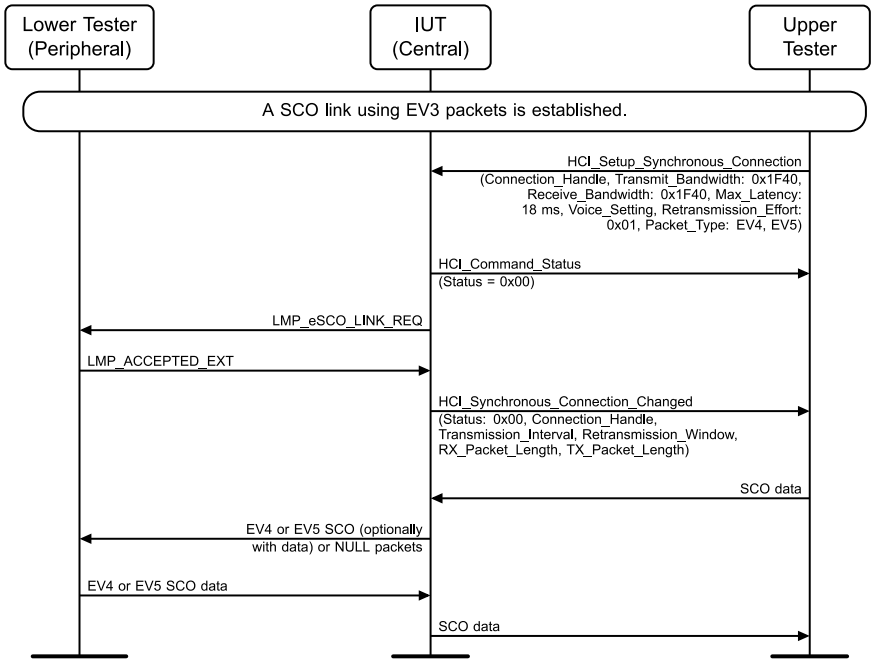


Figure 4.8-65: LMP/LIH/BV-112-C [Request eSCO change] MSC

1. The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with the Connection_Handle, Transmit_Bandwidth and Receive_Bandwidth set to 0x1F40 (8000), Max_Latency set to 18 ms, Voice_Setting, Retransmission_Effort set to 0x01, and Packet_Type set to EV4 and EV5 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_eSCO_LINK_REQ PDU to the Lower Tester with valid values.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
4. The IUT sends a successful HCI_Synchronous_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
5. The Upper Tester sends SCO data to the IUT.
6. The IUT and the Lower Tester exchange EV4 or EV5 SCO or NULL packets.
7. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_eSCO_LINK_REQ PDU to the Lower Tester. Data is transferred using the bandwidth, max latency, and Air_Mode specified in the HCI command in Step 1.

  In Step 6, the IUT and the Lower Tester exchange the specified EV4 or EV5 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

LMP/LIH/BV-113-C [eSCO change rejected]

- Test Purpose

  Verify that the IUT can request a change of the eSCO parameters (packet type EV3 to EV4 or EV5). Verify that the IUT accepts a rejection from the Lower Tester.

- Reference

  [1] 4.6.2.3

- Initial Condition

  - See LMP/LIH/BV-110-C [Request eSCO as Central].
  - The IUT is the Central and initiates the service. The Lower Tester is the Peripheral.
  - A SCO link using EV3 packets has been established.

- Test Procedure

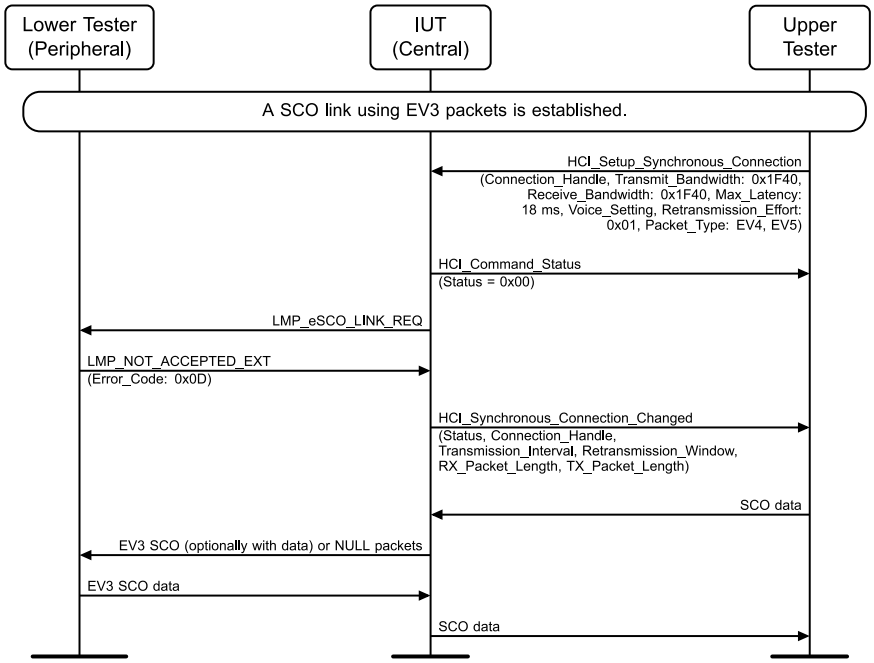


Figure 4.8-66: LMP/LIH/BV-113-C [eSCO change rejected] MSC

1. The Upper Tester sends an HCI_Setup_Synchronous_Connection command to the IUT with the Connection_Handle, Transmit_Bandwidth and Receive_Bandwidth set to 0x1F40 (8000), Max_Latency set to 18 ms, Voice_Setting, Retransmission_Effort set to 0x01, and Packet_Type set to EV4 and EV5 and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_eSCO_LINK_REQ PDU to the Lower Tester with valid values.
3. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_eSCO_LINK_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x0D (Connection Rejected due to Limited Resources).
4. The IUT sends an HCI_Synchronous_Connection_Changed event to the Upper Tester with the Status, Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
5. The Upper Tester sends SCO data to the IUT.
6. The IUT and the Lower Tester exchange EV3 SCO or NULL packets.
7. The IUT sends SCO data to the Upper Tester.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_eSCO_LINK_REQ PDU to the Lower Tester.

  After receiving the LMP_NOT_ACCEPTED_EXT PDU from the Lower Tester in Step 3, data is still transferred according to the original configuration.

  In Step 6, the IUT and the Lower Tester exchange the specified EV3 packets. The packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

LMP/LIH/BV-114-C [Request to close eSCO link]

- Test Purpose

  Verify that the IUT can request to close the eSCO link.

- Reference

  [1] 4.6.2.4

- Initial Condition
  - See LMP/LIH/BV-110-C [Request eSCO as Central].
  - The IUT is the Central and initiates the service. The Lower Tester is the Peripheral.
  - A SCO link using EV3 packets has been established.

- Test Procedure

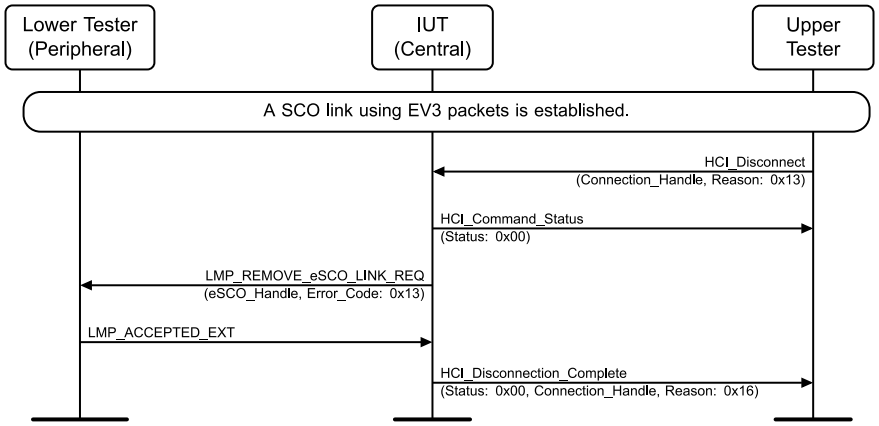


Figure 4.8-67: LMP/LIH/BV-114-C [Request to close eSCO link] MSC

1. The Upper Tester sends an HCI_Disconnect command to the IUT with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection) and receives a successful HCI_Command_Status in response.
2. The IUT sends an LMP_REMOVE_eSCO_LINK_REQ PDU to the Lower Tester with the eSCO_Handle set to the handle of the current eSCO link and Error_Code set to 0x13 (Remote User Terminated Connection).
3. The Lower Tester sends an LMP_ACCEPTED_EXT PDU to the IUT with the LMP_REMOVE_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
4. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x16 (Connection Terminated By Local Host).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_REMOVE_eSCO_LINK_REQ PDU to the Lower Tester and sends the HCI_Disconnection_Complete event to the Upper Tester.

  The eSCO link is removed.

LMP/LIH/BV-115-C [Accept eSCO closure as Central]

- Test Purpose

  Verify that the Lower Tester can request that the IUT close the eSCO link.

- Reference

  [1] 4.6.2.4

- Initial Condition

  - See LMP/LIH/BV-110-C [Request eSCO as Central].
  - The IUT is the Central. The Lower Tester is the Peripheral and initiates the service.
  - A SCO link using EV3 packets has been established.

- Test Procedure

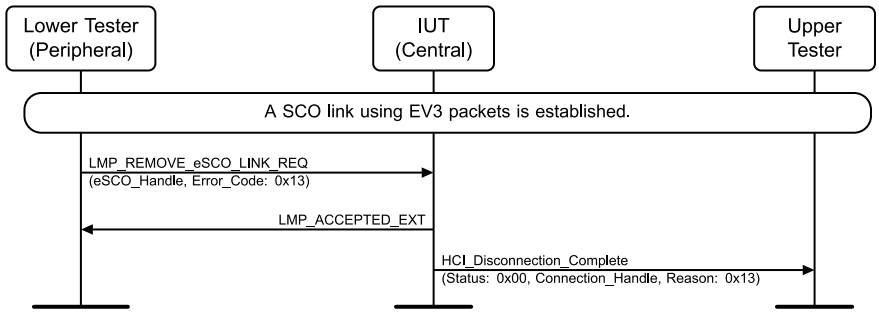


Figure 4.8-68: LMP/LIH/BV115-C [Accept eSCO Closure as Central] MSC

1. The Lower Tester sends an LMP_REMOVE_eSCO_LINK_REQ PDU to the IUT with the eSCO_Handle set to the handle of the current eSCO link and Error_Code set to 0x13 (Remote User Terminated Connection).
2. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_REMOVE_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
3. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection).

Verify that the eSCO link is removed.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester upon reception of the LMP_REMOVE_eSCO_LINK_REQ PDU from the Lower Tester.

  The eSCO link is removed.

LMP/LIH/BV-116-C [Reject eSCO Request]

- Test Purpose

  Verify that the IUT responds that it does not support eSCO links upon request from the Lower Tester.

- Reference

  [1] 4.6.2

- Initial Condition

  - See the “Default settings” section.
  - The Lower Tester initiates the service.
  - An ACL connection has been established.

- Test Procedure

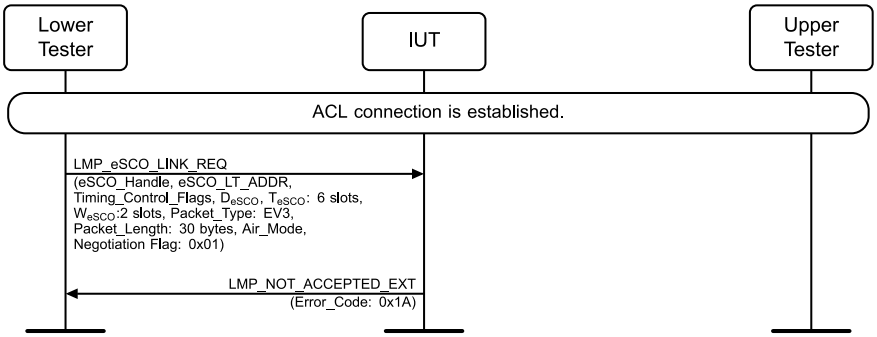


Figure 4.8-69: LMP/LIH/BV-116-C [Reject eSCO Request] MSC

1. The Lower Tester sends an LMP_eSCO_LINK_REQ PDU to the IUT with:
   - eSCO_Handle and eSCO_LT_ADDR set to any valid number if the IUT is a Peripheral; otherwise, set to 0
   - Timing_Control_Flags derived from Central’s clock
   - D_{eSCO} set to any number in the range [0, T_{eSCO} – 2]
   - T_{eSCO} set to 6 slots
   - W_{eSCO} set to 2 slots
   - Packet type C→P and Packet type P→C set to EV3
   - Packet_Length C→P and Packet_Length P→C set to 30 bytes
   - Air_Mode set to any Air_Mode
   - Negotiation Flag set to 0x01 (Initiate Negotiation)
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_eSCO_LINK_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x1A (Unsupported Remote Feature).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with Error_Code set to 0x1A (Unsupported Remote Feature) upon reception of the LMP_eSCO_LINK_REQ PDU from the Lower Tester.

4.8.20 Sniff Subrating

Verify that the IUT can imitate and reject a sniff subrating link request. See the Baseband Test Suite, Section 4.15.2.1, “Sniff Subrating Preamble” [11].

LMP/LIH/BV-117-C [LMP Feature Bits]

- Test Purpose

  Verify that a device has set the correct LMP feature bits for sniff subrate.

- Reference

  [1] 3.2, 3.3

- Initial Condition

  - The IUT is a Peripheral.
  - An ACL connection has been established between the Lower Tester and the IUT, where the IUT is Peripheral.

- Test Procedure

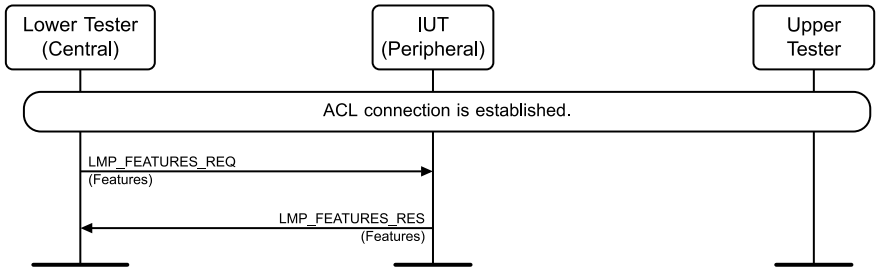


Figure 4.8-70: LMP/LIH/BV-117-C [LMP Feature Bits] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.

- Expected Outcome

  Pass verdict

  The following feature bits are set in the LMP_FEATURES_RES PDU Features parameter from the IUT:

  - Bit 7, “Sniff Mode” (Byte 0, Bit 7)
  - Bit 41, “Sniff Subrating” (Byte 5, Bit 1)

LMP/LIH/BV-118-C [Entering Sniff Subrating Mode from Sniff Mode with Lower Tester as the Initiator]

- Test Purpose

  Verify that the IUT enters Sniff Subrating mode when the Lower Tester initiates a Sniff Subrating mode request.

- Reference

  [1] 4.5.3.3

  [9] 5.1, 5.2

- Initial Condition
  - The IUT is the Peripheral.
  - The Lower Tester and the IUT have a connection in Sniff mode with the following parameters:
    - T_{Sniff} = 20 slots
    - Sniff_Attempt = 1
    - Sniff_Timeout = 0
  - No ACL_U data is currently being exchanged.

- Test Procedure

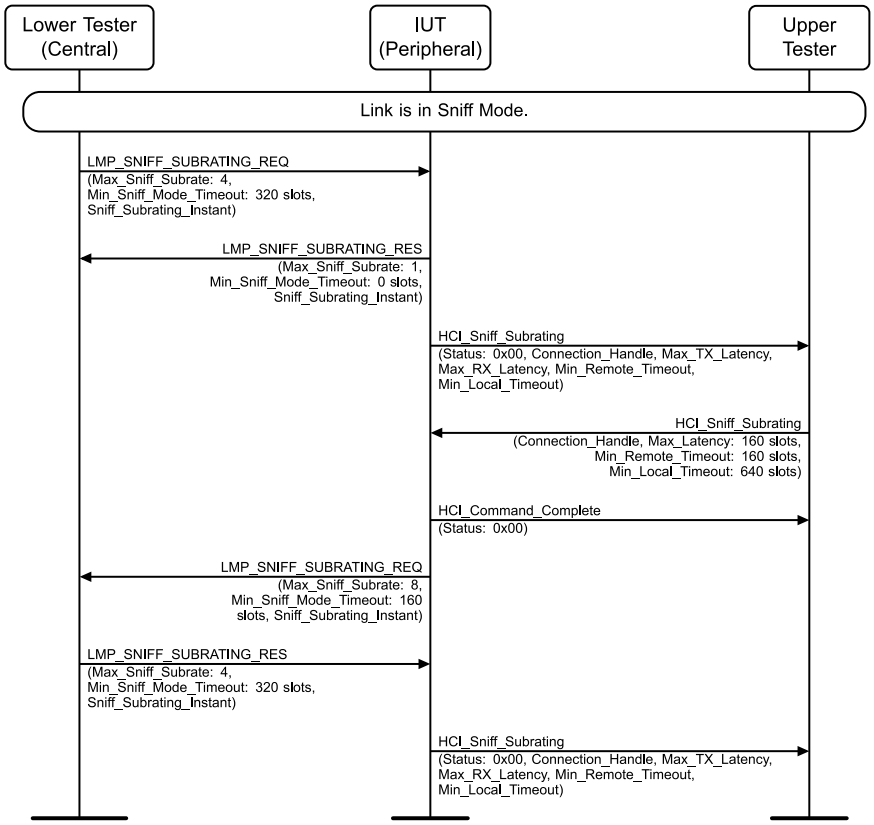


Figure 4.8-71: LMP/LIH/BV-118-C [Entering Sniff Subrating Mode from Sniff Mode With Lower Tester as The Initiator] MSC

1. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
   - Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots
2. The IUT responds to the Lower Tester with an LMP_SNIFF_SUBRATING_RES PDU with the following parameters (sniff subrating default values):
   - Max_Sniff_Subrate = 1
   - Min_Sniff_Mode_Timeout = 0 slots
3. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.
4. The Upper Tester sends an HCI_Sniff_Subrating command to the IUT with the Connection_Handle and the following parameters:
   - Max_Latency = 160 slots
   - Min_Remote_Timeout = 160 slots
   - Min_Local_Timeout = 640 slots
5. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
6. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with the following parameters:
   - Max_Sniff_Subrate = 8
   - Min_Sniff_Mode_Timeout = 160 slots
7. The Lower Tester sends an LMP_SNIFF_SUBRATING_RES PDU to the IUT with the following parameters (same parameters as the previous negotiation):
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
   - Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots
8. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

- Expected Outcome

  Pass verdict

  In Step 3, the Sniff Subrating event has been observed, and the IUT sends the HCI_Sniff_Subrating event to the Upper Tester with the following parameters:

  - Max_TX_Latency = 20 slots
  - Max_RX_Latency = 80 slots
  - Min_Remote_Timeout = 0 slots
  - Min_Local_Timeout = 320 slots

  In Step 8, the Sniff Subrating event has been observed, and the IUT sends the HCI_Sniff_Subrating event to the Upper Tester with the following parameters:

  - Max_TX_Latency = 160 slots
  - Max_RX_Latency = 80 slots
  - Min_Remote_Timeout = 160 slots
  - Min_Local_Timeout = 640 slots

LMP/LIH/BV-119-C [Entering Sniff Subrating Mode From Sniff Mode With IUT As The Initiator]

- Test Purpose

  Verify that the IUT enters Sniff Subrating mode when the IUT initiates a Sniff Subrating mode request.

- Reference

  [1] 4.5.3.3

  [9] 5.1, 5.2

- Initial Condition
  - The IUT is the Peripheral.
  - The Lower Tester and the IUT have a connection in Sniff mode with the following parameters:
    - T_{Sniff} = 20 slots
    - Sniff_Attempt = 1
    - Sniff_Timeout = 0
  - No ACL_U data is currently being exchanged.

- Test Procedure

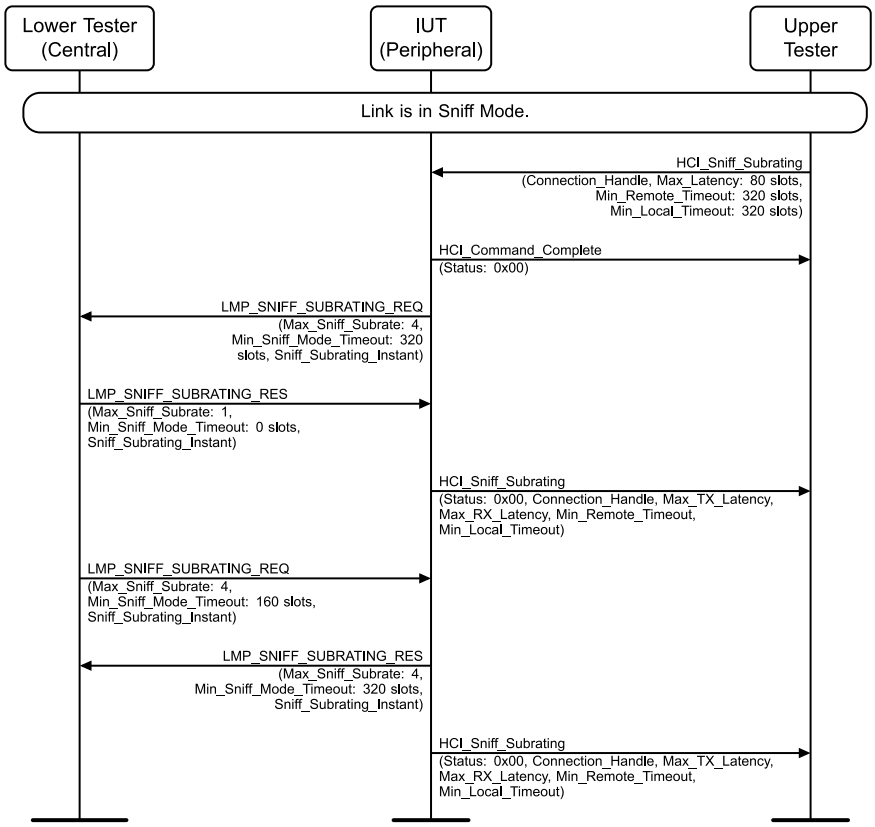


Figure 4.8-72: LMP/LIH/BV-119-C [Entering Sniff Subrating Mode From Sniff Mode With IUT As The Initiator] MSC

1. The Upper Tester sends an HCI_Sniff_Subrating command to the IUT with the following parameters:
   - Max_Latency = 80 slots
   - Min_Remote_Timeout = 320 slots
   - Min_Local_Timeout = 320 slots
2. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
3. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
4. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES PDU with the following parameters (sniff subrating default values):
   - Max_Sniff_Subrate = 1
   - Min_Sniff_Mode_Timeout = 0 slots
   - Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots
5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.
6. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 160 slots
   - Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots
7. The IUT responds to the Lower Tester with an LMP_SNIFF_SUBRATING_RES PDU with the following parameters (same parameters as the former negotiation):
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
8. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

- Expected Outcome

  Pass verdict

  In Step 5, the Sniff Subrating event has been observed, and the IUT sends the HCI_Sniff_Subrating event to the Upper Tester with the following parameters:

  - Max_TX_Latency = 80 slots
  - Max_RX_Latency = 20 slots
  - Min_Remote_Timeout = 320 slots
  - Min_Local_Timeout = 320 slots

  In Step 8, the Sniff Subrating event has been observed, and the IUT sends the HCI_Sniff_Subrating event to the Upper Tester with the following parameters:

  - Max_TX_Latency = 80 slots
  - Max_RX_Latency = 80 slots
  - Min_Remote_Timeout = 320 slots
  - Min_Local_Timeout = 320 slots

LMP/LIH/BV-120-C [IUT Rejects Sniff Subrating Request When In Active Mode]

- Test Purpose

  Verify that the IUT rejects a Sniff Subrating request correctly when the connection is still in Active mode.

- Reference

  [1] 4.5.3.3

  [9] 5.2, 5.2

- Initial Condition

  - The Lower Tester and the IUT have a connection in Active mode.
  - No Sniff mode negotiation is going on.
  - The IUT is a Peripheral or Central.

- Test Procedure

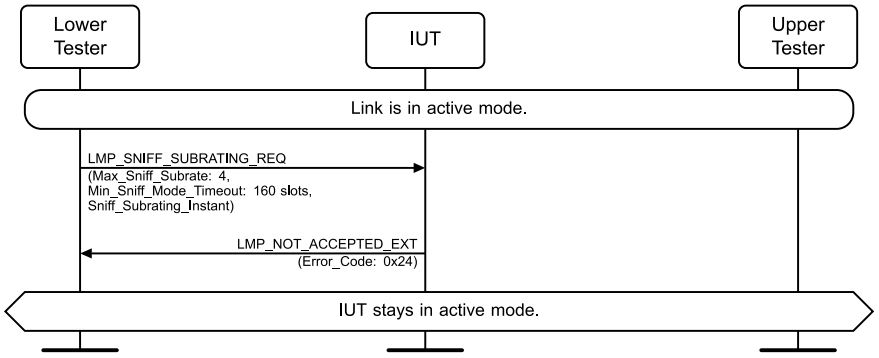


Figure 4.8-73: LMP/LIH/BV-120-C [IUT Rejects Sniff Subrating Request When In Active Mode] MSC

1. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
   - Max_Sniff_Subrate = 2
   - Min_Sniff_Mode_Timeout = 80
   - Sniff_Subrating_Instant: at least 80 slots ahead of the current piconet clock but not more than 400 slots
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with LMP_SNIFF_SUBRATING_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x24 (LMP PDU Not Allowed).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with Error_Code set to 0x24 (LMP PDU Not Allowed) in response to the LMP_SNIFF_SUBRATING_REQ PDU sent by the Lower Tester.

  No HCI_Sniff_Subrating event is observed by the Upper Tester.

LMP/LIH/BV-121-C [Entering Sniff Subrating Mode With Lower Tester Initiating Sniff Mode Request]

- Test Purpose

  Verify that the IUT enters Sniff Subrating mode when the Lower Tester and the IUT have Sniff Subrating parameters already (either just obtained from their hosts or from sniff subrating history) and the Lower Tester initiates a Sniff mode request.

- Reference

  [1] 4.5.3.3

  [9] 5.1, 5.2

- Initial Condition
  - The IUT is a Central or Peripheral.
  - The Lower Tester and the IUT have a connection in Active mode.
  - No ACL_U data is currently being exchanged.
  - The Lower Tester has the following parameters:
    - Max_Latency = 80 slots
    - Min_Remote_Timeout = 320 slots
    - Min_Local_Timeout = 320 slots

- Test Procedure

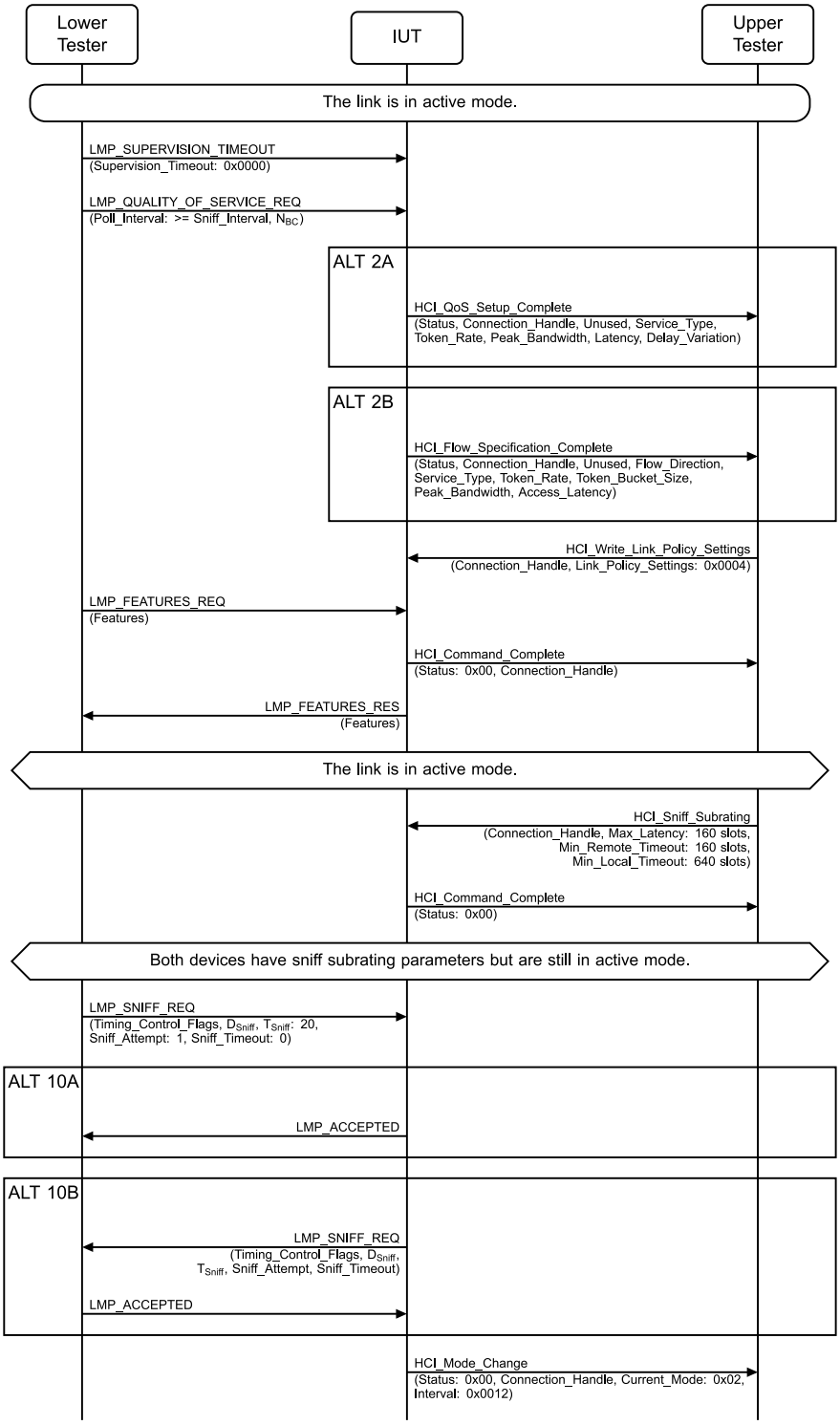


Figure 4.8-74: LMP/LIH/BV-121-C [Entering Sniff Subrating Mode With Lower Tester Initiating Sniff Mode Request] MSC – Page 1 of 2

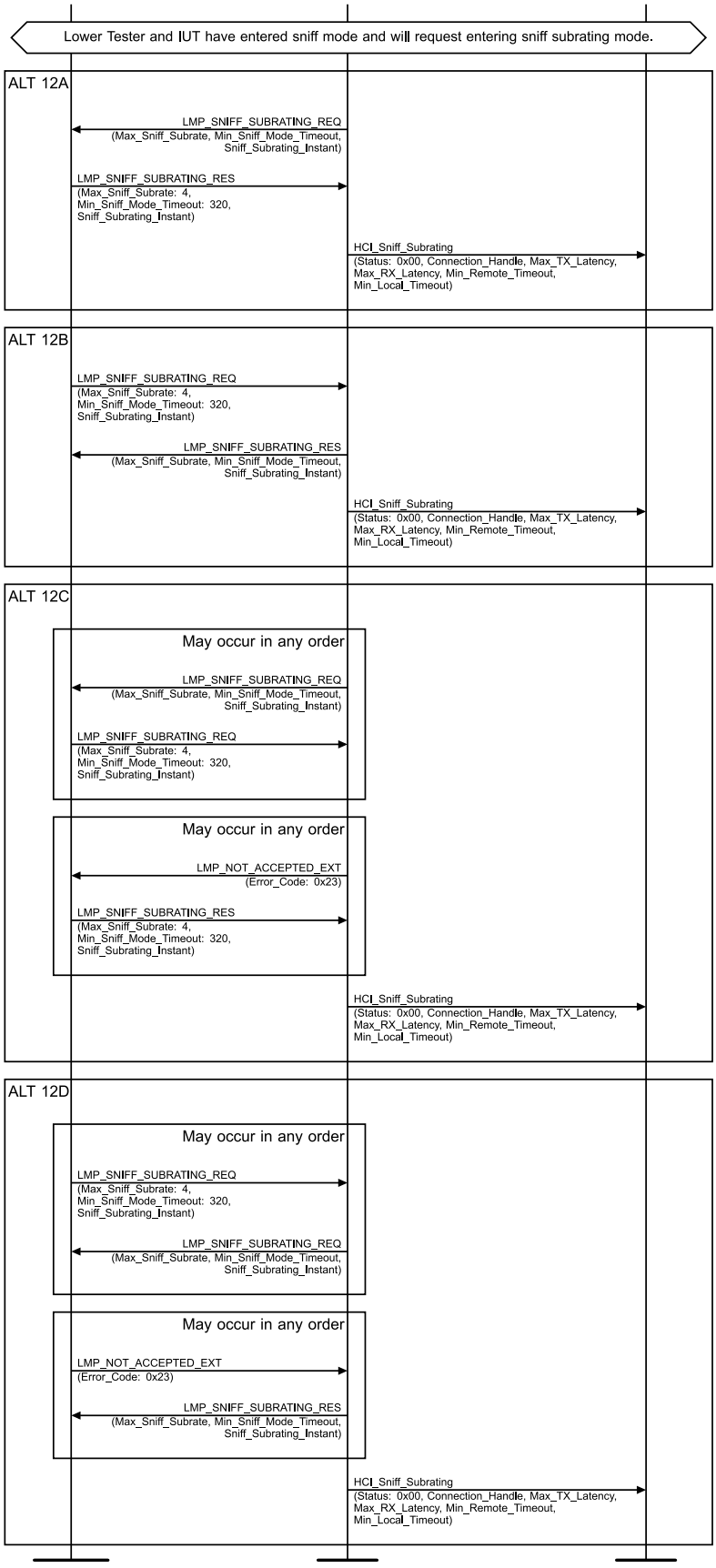


Figure 4.8-75: LMP/LIH/BV-121-C [Entering Sniff Subrating Mode With Lower Tester Initiating Sniff Mode Request] MSC – Page 2 of 2

1. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU with the Supervision_Timeout set to 0x0000 and an LMP_QUALITY_OF_SERVICE_REQ PDU with the Poll_Interval greater than or equal to the Sniff_Interval and N_{BC}.
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   - 2A.1. The IUT sends an HCI_QoS_Setup_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

   Alternative 2B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):
   - 2B.1. The IUT sends an HCI_Flow_Specification_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.
3. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0004 (Sniff Mode).
4. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
5. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
6. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester with the Features parameter.
7. The Upper Tester sends an HCI_Sniff_Subrating command to the IUT with the Connection_Handle and the following parameters:
   - Max_Latency = 160 slots
   - Min_Remote_Timeout = 160 slots
   - Min_Local_Timeout = 640 slots
8. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
9. The Lower Tester sends an LMP_SNIFF_REQ PDU to the IUT with the following Sniff parameters:
   - T_{Sniff} = 20 slots
   - Sniff_Attempt = 1
   - Sniff_Timeout = 0
10. Perform either alternative 10A or 10B depending on the IUT's response.
    Alternative 10A (The IUT sends the LMP_ACCEPTED PDU to the Lower Tester):
    - 10A.1. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.

    Alternative 10B (The IUT sends the LMP_SNIFF_REQ PDU to the Lower Tester):
    - 10B.1. The IUT responds to the Lower Tester with an LMP_SNIFF_REQ PDU with Timing_Control_Flags, D_{Sniff}, T_{Sniff}, Sniff_Attempt, and Sniff_Timeout.
    - 10B.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
11. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval set to 0x0012.
12. Perform either alternative 12A, 12B, 12C, or 12D depending on the IUT's response.
    Alternative 12A (The IUT sends the LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester):
    - 12A.1. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.
    - 12A.2. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES with the following parameters:
      - Max_Sniff_Subrate = 4
      - Min_Sniff_Mode_Timeout = 320
      - Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

12A.3. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

Alternative 12B (The Lower Tester sends the LMP_SNIFF_SUBRATING_REQ PDU to the IUT):

12B.1. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

12B.2. The IUT responds to the Lower Tester with an LMP_SNIFF_SUBRATING_RES PDU with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

12B.3. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

Alternative 12C (Negotiation collision when the IUT is the Central):

Steps 12C.1 and 12C.2 can occur in either order.

12C.1. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

12C.2. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

Steps 12C.3 and 12C.4 can occur in either order.

12C.3. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with LMP_SNIFF_SUBRATING_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x23 (LMP Error Transaction Collision / LL Procedure Collision).

12C.4. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

12C.5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

Alternative 12D (Negotiation collision when the Lower Tester is the Central):

Steps 12D.1 and 12D.2 can occur in either order.

12D.1. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

12D.2. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

Steps 12D.3 and 12D.4 can occur in either order.

12D.3. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED_EXT PDU with an LMP_SNIFF_SUBRATING_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x23 (LMP Error Transaction Collision / LL Procedure Collision).

12D.4. The IUT responds to the Lower Tester with an LMP_SNIFF_SUBRATING_RES with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

12D.5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

- Expected Outcome

  Pass verdict

  In Step 11, the IUT sends the HCI_Mode_Change event to the Upper Tester with Current_Mode set to 0x02 (Sniff Mode).

  In Step 12, the IUT sends the HCI_Sniff_Subrating Event to the Upper Tester with the following parameters:

  - Max_TX_Latency = T_{Sniff} * max_sniff_subrate_transmit slots
  - Max_RX_Latency = T_{Sniff} * max_sniff_subrate_receive slots
  - Min_Remote_Timeout = 320 slots
  - Min_Local_Timeout = 320 slots

LMP/LIH/BV-122-C [Entering Sniff Subrating Mode with IUT Initiating Sniff Mode Request]

- Test Purpose

  Verify that the IUT enters Sniff Subrating mode when the Lower Tester and the IUT already have Sniff Subrating parameters (either just obtained from their hosts or from sniff subrating history) and the IUT initiates a Sniff mode request.

- Reference

  [1] 4.5.3.3, 5.2

- Initial Condition

  - The IUT is a Central or Peripheral.
  - The Lower Tester and the IUT have a connection in Active mode.
  - No ACL_U data is currently being exchanged.
  - The Lower Tester has the following parameters:
    - Maximum_Latency = 80 slots
    - Minimum_Remote_Timeout = 320 slots
    - Minimum_Local_Timeout = 320 slots

- Test Procedure

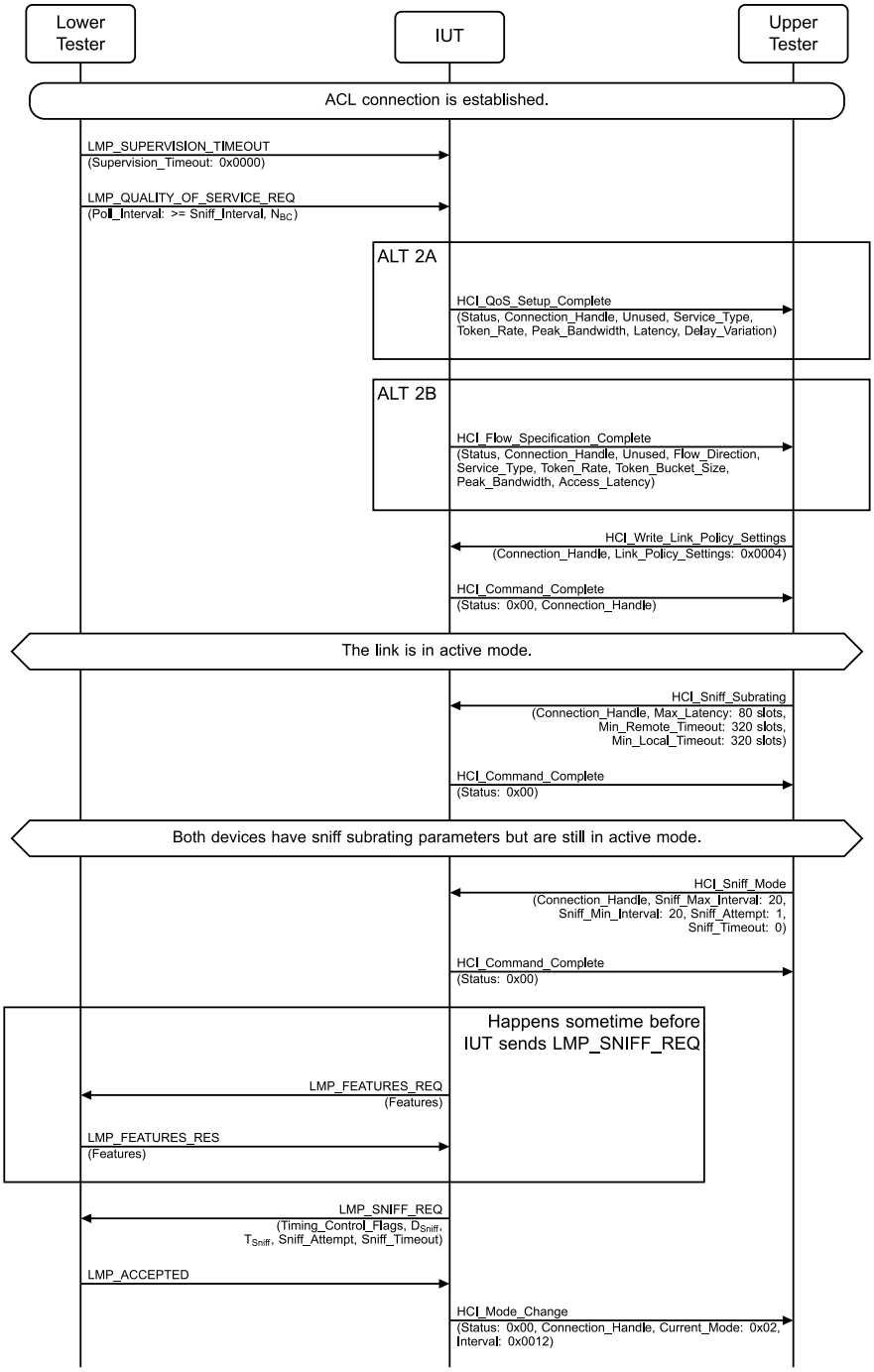


Figure 4.8-76: LMP/LIH/BV-122-C [Entering Sniff Subrating Mode with IUT Initiating Sniff Mode Request] MSC – Page 1 of 2

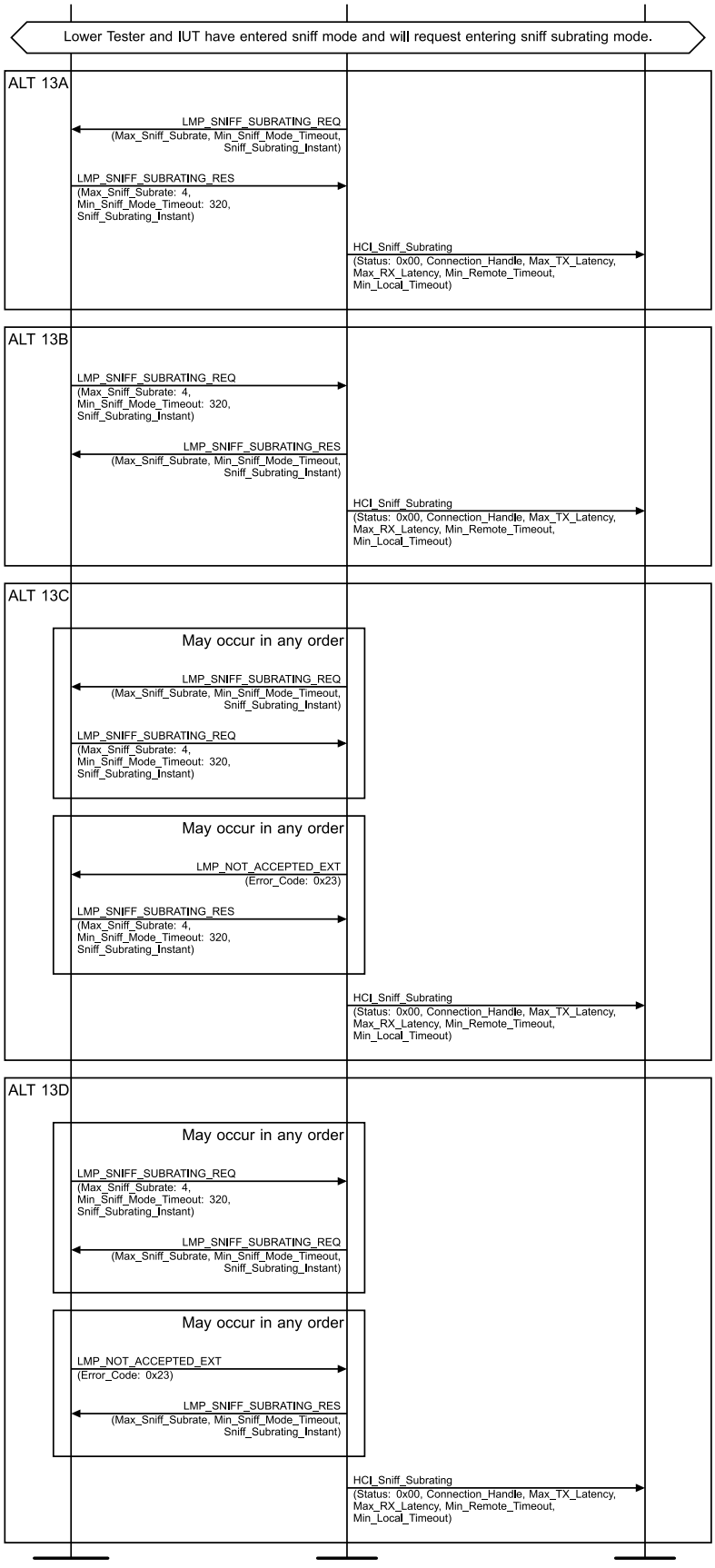


Figure 4.8-77: LMP/LIH/BV-122-C [Entering Sniff Subrating Mode with IUT Initiating Sniff Mode Request] MSC – Page 2 of 2

1. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU with the Supervision_Timeout set to 0x0000 and an LMP_QUALITY_OF_SERVICE_REQ PDU with the Poll_Interval greater than or equal to the Sniff_Interval and N_{BC}.
2. Perform either alternative 2A or 2B depending on the IUT's response.
   Alternative 2A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):
   2A.1. The IUT sends an HCI_QoS_Setup_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.
   Alternative 2B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):
   2B.1. The IUT sends an HCI_Flow_Specification_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.
3. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0004 (Sniff Mode) and receives a successful HCI_Command_Complete event in response.
4. The Upper Tester sends an HCI_Sniff_Subrating command to the IUT with the Connection_Handle and the following parameters:
   - Max_Latency = 80 slots
   - Min_Remote_Timeout = 320 slots
   - Min_Local_Timeout = 320 slots
5. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
6. The Upper Tester sends an HCI_Sniff_Mode command to the IUT with the Connection_Handle and the following parameters:
   - Sniff_Max_Interval = 20 slots
   - Sniff_Min_Interval = 20 slots
   - Sniff_Attempt = 1
   - Sniff_Timeout = 0
7. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
8. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
9. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT with the Features parameter.
10. The IUT sends an LMP_SNIFF_REQ PDU to the IUT with Timing_Control_Flags, D_{Sniff}, T_{Sniff}, Sniff_Attempt, and Sniff_Timeout.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
12. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval set to 0x0012.
13. Perform either alternative 13A, 13B, 13C, or 13D depending on the IUT's response.
    Alternative 13A (The IUT sends the LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester):
    13A.1. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.
    13A.2. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES with the following parameters:
    - Max_Sniff_Subrate = 4
    - Min_Sniff_Mode_Timeout = 320
    - Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

    13A.3. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

Alternative 13B (The Lower Tester sends the LMP_SNIFF_SUBRATING_REQ PDU to the IUT):

13B.1. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

13B.2. The IUT responds to the Lower Tester with an LMP_SNIFF_SUBRATING_RES PDU with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

13B.3. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

Alternative 13C (Negotiation collision when the IUT is the Central):

Steps 13C.1 and 13C.2 can occur in either order.

13C.1. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

13C.2. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

Steps 13C.3 and 13C.4 can occur in either order.

13C.3. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with LMP_SNIFF_SUBRATING_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x23 (LMP Error Transaction Collision / LL Procedure Collision).

13C.4. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

13C.5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

Alternative 13D (Negotiation collision when the Lower Tester is the Central):

Steps 13D.1 and 13D.2 can occur in either order.

13D.1. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
- Max_Sniff_Subrate = 4
- Min_Sniff_Mode_Timeout = 320
- Sniff_Subrating_Instant = at least 80 slots ahead of the current piconet clock but not more than 400 slots

13D.2. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

Steps 13D.3 and 13D.4 can occur in either order.

13D.3. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED_EXT PDU with an LMP_SNIFF_SUBRATING_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x23 (LMP Error Transaction Collision / LL Procedure Collision).

13D.4. The IUT responds to the Lower Tester with an LMP_SNIFF_SUBRATING_RES with Max_Sniff_Subrate, Min_Sniff_Mode_Timeout, and Sniff_Subrating_Instant.

13D.5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

- Expected Outcome

  Pass verdict

  In Step 12, the IUT sends the HCI_Mode_Change event to the Upper Tester with Current_Mode set to 0x02 (Sniff Mode).

  In Step 13, the IUT sends the HCI_Sniff_Subrating event to the Upper Tester with the following parameters:

  - Max_TX_Latency = 80 slots
  - Max_RX_Latency = 80 slots
  - Min_Remote_Timeout = 320 slots
  - Min_Local_Timeout = 320 slots

LMP/LIH/BV-123-C [IUTs Transitioning from Existing Sniff Subrating Mode to A New Set of Subrating Parameters]

- Test Purpose

  Verify that an IUT already in Sniff Subrating mode will transition to a new set of subrating parameters successfully.

- Reference

  [1] 4.5.3.3, 5.2

- Initial Condition

  - The IUT is the Peripheral.
  - The Lower Tester and the IUT have a connection in Sniff mode with the following parameters:
    - T_{Sniff} = 20 slots
    - Sniff_Attempt = 1
    - Sniff_Timeout = 0
  - No data is currently being exchanged between the IUT and the Lower Tester.

- Test Procedure

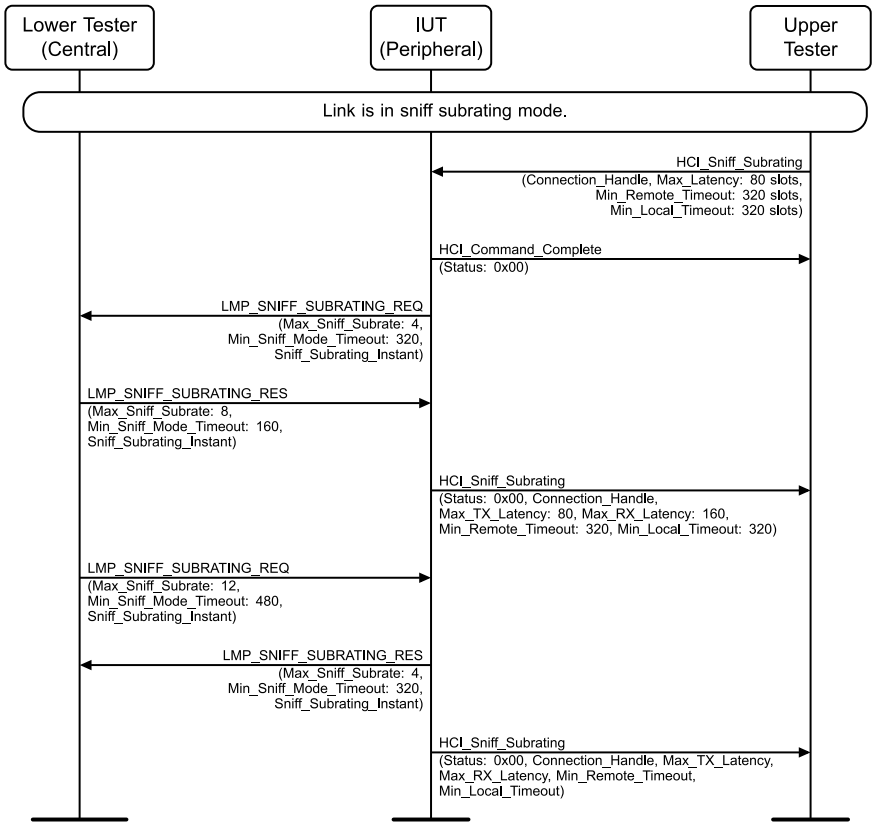


Figure 4.8-78: LMP/LIH/BV-123-C [IUTs Transitioning from Existing Sniff Subrating Mode to A New Set of Subrating Parameters] MSC

1. The Upper Tester sends an HCI_Sniff_Subrating command to the IUT with the following parameters:
   - Max_Latency = 80 slots
   - Min_Remote_Timeout = 320 slots
   - Min_Local_Timeout = 320 slots
2. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
3. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
4. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES PDU with the following parameters:
   - Max_Sniff_Subrate = 8
   - Min_Sniff_Mode_Timeout = 160 slots
   - Sniff_Subrating_Instant = at least 240 slots ahead of the current piconet clock but not more than 360 slots
5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency set to 80 slots, Max_RX_Latency set to 160 slots, Min_Remote_Timeout set to 320 slots, and Min_Local_Timeout set to 320 slots.

6. The Lower Tester sends an LMP_SNIFF_SUBRATING_REQ PDU to the IUT with the following parameters:
   - Max_Sniff_Subrate = 12
   - Min_Sniff_Mode_Timeout = 480 slots
   - Sniff_Subrating_Instant = at least 480 slots ahead of the current piconet clock but not more than 960 slots
7. The IUT responds to the Lower Tester with an LMP_SNIFF_SUBRATING_RES PDU with the following parameters (same parameters as the former negotiation):
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
8. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency, Max_RX_Latency, Min_Remote_Timeout, and Min_Local_Timeout.

- Expected Outcome

  Pass verdict

  In Step 8, the IUT sends the HCI_Sniff_Subrating event to the Upper Tester with the following parameters:

  - Max_TX_Latency = 80 slots
  - Max_RX_Latency = 240 slots
  - Min_Remote_Timeout = 320 slots
  - Min_Local_Timeout = 480 slots

LMP/LIH/BV-124-C [Sniff Subrating Mode to Active Mode Transition Initiated By Lower Tester]

- Test Purpose

  Verify that the IUT can transition from Sniff Subrating mode to Active mode when the Lower Tester's host issues an HCI_Exit_Sniff_Mode command.

- Reference

  [1] 4.5.3.3, 5.2

- Initial Condition

  - The IUT is the Peripheral.
  - The Lower Tester and the IUT have a connection in Sniff mode.

- Test Procedure

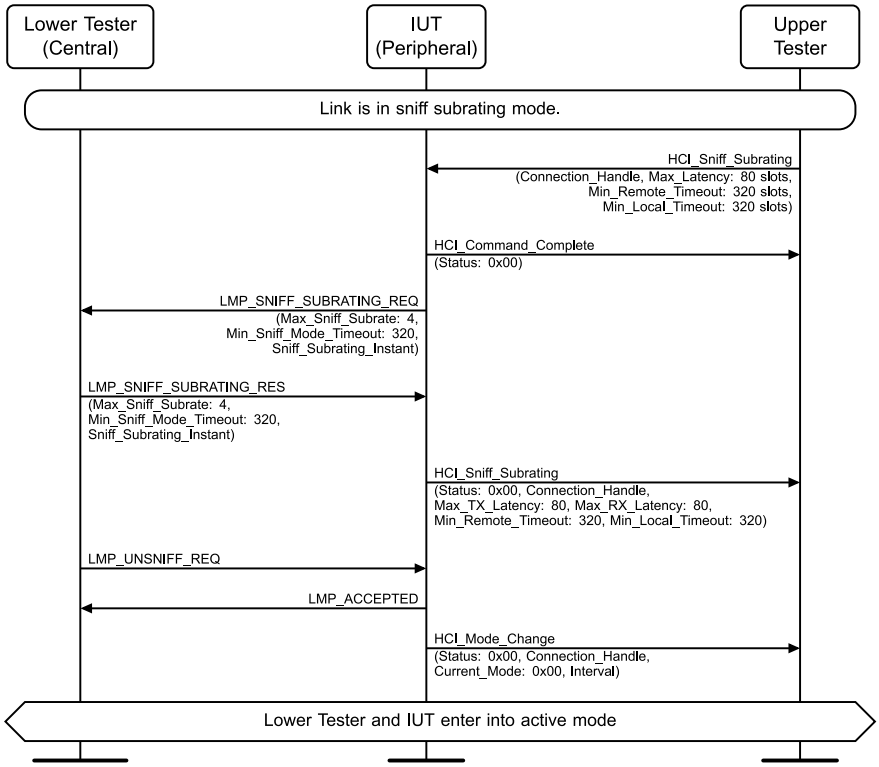


Figure 4.8-79: LMP/LIH/BV-124-C [Sniff Subrating Mode to Active Mode Transition Initiated By Lower Tester] MSC

1. The Upper Tester sends an HCI_Sniff_Subrating command to the IUT with the following parameters:
   - Max_Latency = 80 slots
   - Min_Remote_Timeout = 320 slots
   - Min_Local_Timeout = 320 slots
2. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
3. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
4. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES PDU with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
   - Sniff_Subrating_Instant = at least 240 slots ahead of the current piconet clock but not more than 360 slots
5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency set to 80 slots, Max_RX_Latency set to 80 slots, Min_Remote_Timeout set to 320 slots, and Min_Local_Timeout set to 320 slots.
6. The Lower Tester sends an LMP_UNSNIFF_REQ PDU to the IUT.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_UNSNIFF_REQ PDU Opcode.
8. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.

- Expected Outcome

  Pass verdict

  In Step 8, the IUT sends an HCI_Mode_Change event to the Upper Tester with Current_Mode set to 0x00 (Active Mode).

LMP/LIH/BV-125-C [Sniff Subrating Mode to Active Mode Transition Initiated By IUT]

- Test Purpose

  Verify that the IUT can transition from Sniff Subrating mode to Active mode when the Upper Tester issues an Exit_Sniff command.

- Reference

  [1] 4.5.3.3, 5.2

- Initial Condition

  - The IUT is the Peripheral.
  - The Lower Tester and the IUT have a connection in Sniff mode.

- Test Procedure

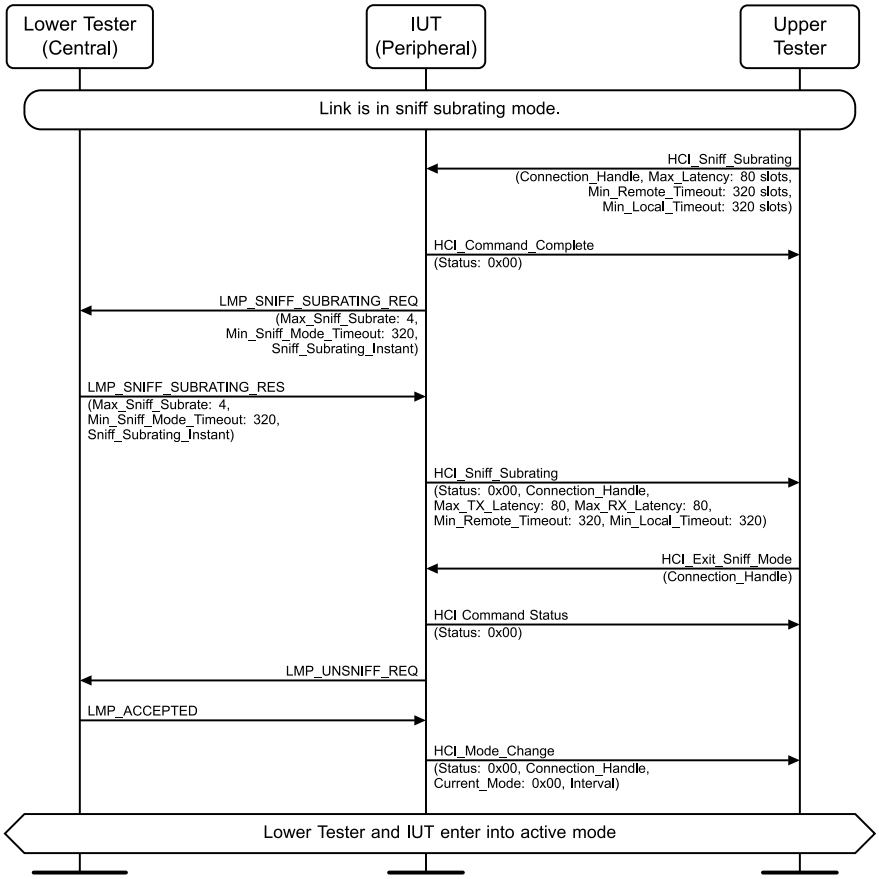


Figure 4.8-80: LMP/LIH/BV-125-C [Sniff Subrating Mode to Active Mode Transition Initiated By IUT] MSC

1. The Upper Tester sends an HCI_Sniff_Subrating command to the IUT with the following parameters:
   - Max_Latency = 80 slots
   - Min_Remote_Timeout = 320 slots
   - Min_Local_Timeout = 320 slots
2. The IUT responds to the Upper Tester with a successful HCI_Command_Complete event.
3. The IUT sends an LMP_SNIFF_SUBRATING_REQ PDU to the Lower Tester with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
4. The Lower Tester responds to the IUT with an LMP_SNIFF_SUBRATING_RES PDU with the following parameters:
   - Max_Sniff_Subrate = 4
   - Min_Sniff_Mode_Timeout = 320 slots
   - Sniff_Subrating_Instant = at least 240 slots ahead of the current piconet clock but not more than 360 slots
5. The IUT sends a successful HCI_Sniff_Subrating event to the Upper Tester with the Connection_Handle, Max_TX_Latency set to 80 slots, Max_RX_Latency set to 80 slots, Min_Remote_Timeout set to 320 slots, and Min_Local_Timeout set to 320 slots.
6. The Upper Tester sends an HCI_Exit_Sniff_Mode command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in response.
7. The IUT sends an LMP_UNSNIFF_REQ PDU to the Lower Tester.
8. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_UNSNIFF_REQ PDU Opcode.
9. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.

- Expected Outcome

  Pass verdict

  In Step 9, the IUT sends an HCI_Mode_Change event to the Upper Tester with Current_Mode set to 0x00 (Active Mode).

4.8.21 Multi-slot Packets – Peripheral

Verify that an IUT can request for the maximum number of slots to be used. The IUT is Peripheral.

LMP/LIH/BV-63-C [Accept Maximum Slot Request]

- Test Purpose

  Verify that the IUT can accept a request to use a maximum number of slots.

- Reference

  [1] 4.1.10

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - An ACL connection has been established with DM1 packets.

- Test Procedure

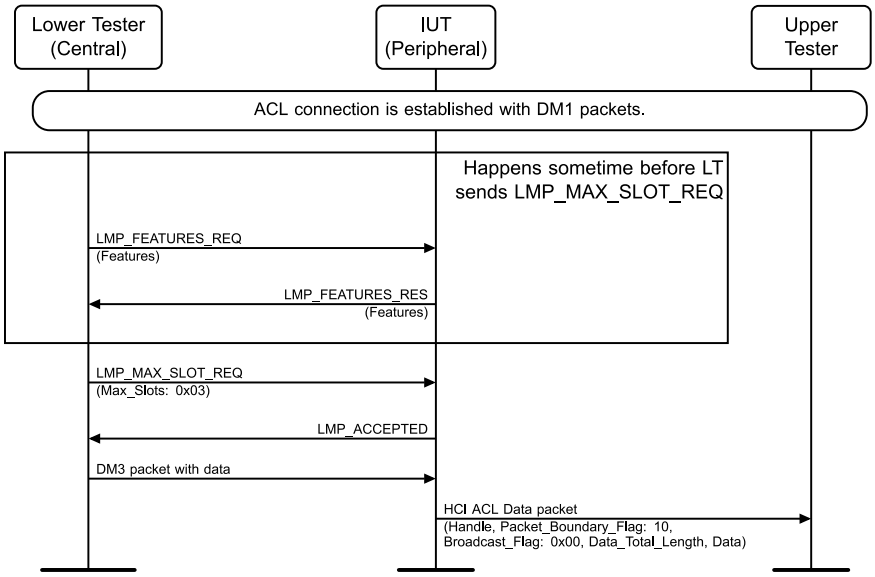


Figure 4.8-81: LMP/LIH/BV-63-C [Accept Maximum Slot Request] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT sends an LMP_FEATURES_RES PDU to the Lower Tester with the Features parameter.
3. The Lower Tester sends an LMP_MAX_SLOT_REQ PDU to the IUT with Max_Slots set to 0x03.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_MAX_SLOT_REQ PDU Opcode.
5. The Lower Tester sends DM3 packets with data to the IUT.
6. The IUT sends HCI ACL Data packets to the Upper Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU to the Lower Tester upon reception of the LMP_MAX_SLOT_REQ PDU from the Lower Tester.

  In Step 6, the IUT sends an HCI ACL Data Packet to the Upper Tester.

- Notes

  If DM3 packets are not supported by the IUT, then they are replaced with DH3 or DM5 or DH5.

LMP/LIH/BV-146-C [Maximum Slot after Role Switch as Peripheral]

- Test Purpose

  Verify that the IUT maximum number of slots is set to 1 after a role switch.

- Reference

  [1] 4.1.10

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral and initiates the service. The Lower Tester is the Central.
  - An ACL connection has been established with DM3, DH3, DM1, or DH1 packets.
- Test Procedure

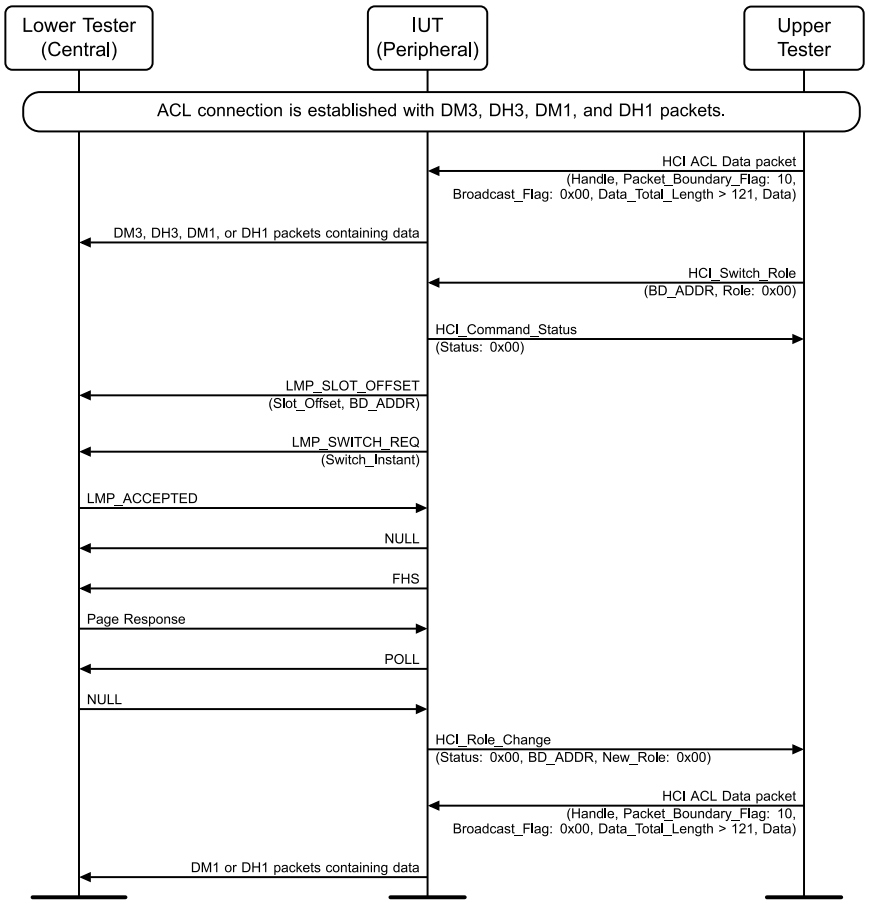


Figure 4.8-82: LMP/LIH/BV-146-C [Maximum Slot after Role Switch as Peripheral] MSC

1. The Upper Tester sends HCI ACL Data packets to the IUT.
2. The IUT sends DM3, DH3, DM1, or DH1 data packets to the Lower Tester.
3. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
4. The IUT sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the Lower Tester.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
6. The IUT sends a NULL packet to the IUT.
7. The IUT sends an FHS packet to the Lower Tester.
8. The Lower Tester responds to the IUT with a Page Response packet.
9. The IUT sends a POLL packet to the Lower Tester.
10. The Lower Tester responds to the IUT with a NULL packet.
11. The Upper Tester sends a successful HCI_Role_Change event with the BD_ADDR and New_Role set to 0x00 (Central) to the Upper Tester.

12. The Upper Tester sends HCI ACL Data packets to the IUT.
13. The IUT sends DM1 or DH1 data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT uses DM1 or DH1 packets after a successful role switch.

- Notes

  If DM3 or DH3 packets are not supported by the IUT, then they are replaced with DM5 or DH5.

4.8.22 Multi-slot Packets – Central

Verify that a unit can request for a maximum of slots to be used. The IUT is the Central.

LMP/LIH/BV-148-C [Maximum Slot after Role Switch as Central]

- Test Purpose

  Verify that the IUT's maximum number of slots is 1 after a role switch.

- Reference

  [1] 4.1.10

- Initial Condition
  - See the "Default settings" section.
  - The IUT is the Central and initiates the service. The Lower Tester is the Peripheral.
  - An ACL connection has been established with DM3, DH3, DM1, or DH1 packets.
  - The IUT must page the Lower Tester to become the Central of the piconet.

- Test Procedure

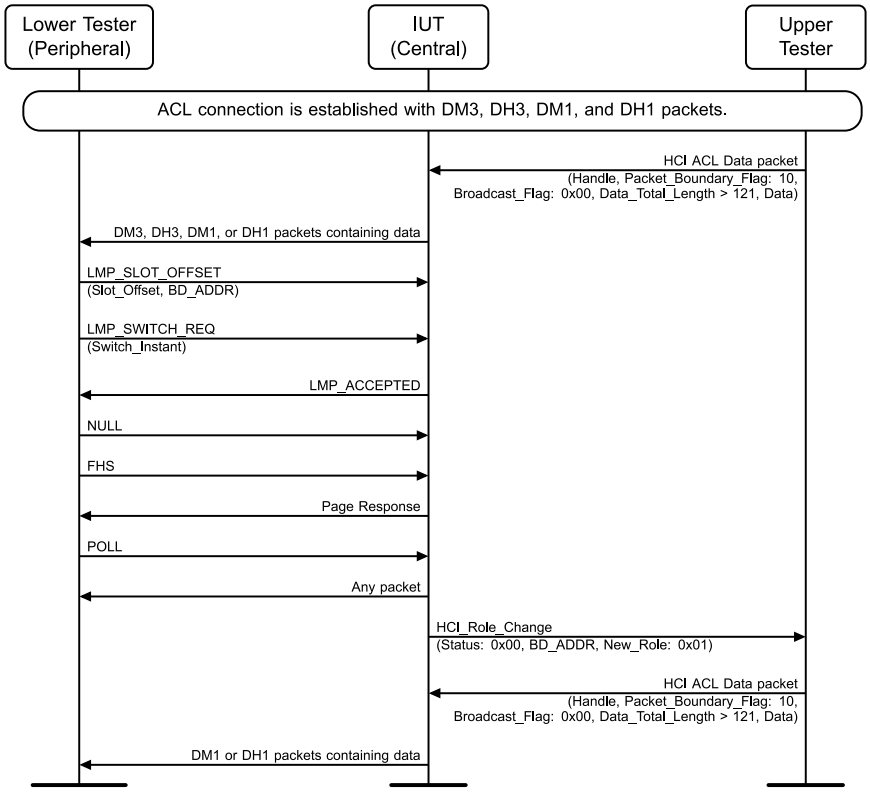


Figure 4.8-83: LMP/LIH/BV-148-C [Maximum Slot after Role Switch as Central] MSC

1. The Upper Tester sends HCI ACL Data packets to the IUT.
2. The IUT sends DM3, DH3, DM1, or DH1 data packets to the Lower Tester.
3. The Lower Tester sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the IUT.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
5. The Lower Tester sends a NULL packet to the IUT.
6. The Lower Tester sends an FHS packet to the IUT.
7. The IUT responds to the Lower Tester with a Page Response packet.
8. The Lower Tester sends a POLL packet to the IUT.
9. The IUT responds to the Lower Tester with any packet.
10. The Upper Tester sends a successful HCI_Role_Change event with the BD_ADDR and New_Role set to 0x01 (Peripheral) to the Upper Tester.
11. The Upper Tester sends HCI ACL Data packets to the IUT.
12. The IUT sends DM1 or DH1 data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT uses DM1 or DH1 packets after a successful role switch.

- Notes

  If DM3 or DH3 packets are not supported by the IUT, then they are replaced with DM5 or DH5.

4.8.23 Multi-slot Packets – Both Central and Peripheral

4.8.23.1 Request Maximum Slots

- Test Purpose

  Verify that the IUT can request and accept a maximum number of slots.

- Reference

  [1] 4.1.10

- Initial Condition

  - See the “Default settings” section.
  - The IUT is in the role indicated in Table 4.8-11 and initiates the service.
  - An ACL connection has been established with DM1 packets.
  - If the IUT is in the Central role, the IUT must page the Lower Tester to become the Central of the piconet.

- Test Case Configuration

Test Case	Role
LMP/LIH/BV-61-C [Request Maximum Slots as Peripheral]	Peripheral
LMP/LIH/BV-64-C [Request Maximum Slots as Central]	Central

Table 4.8-11: Request Maximum Slots test cases

- Test Procedure

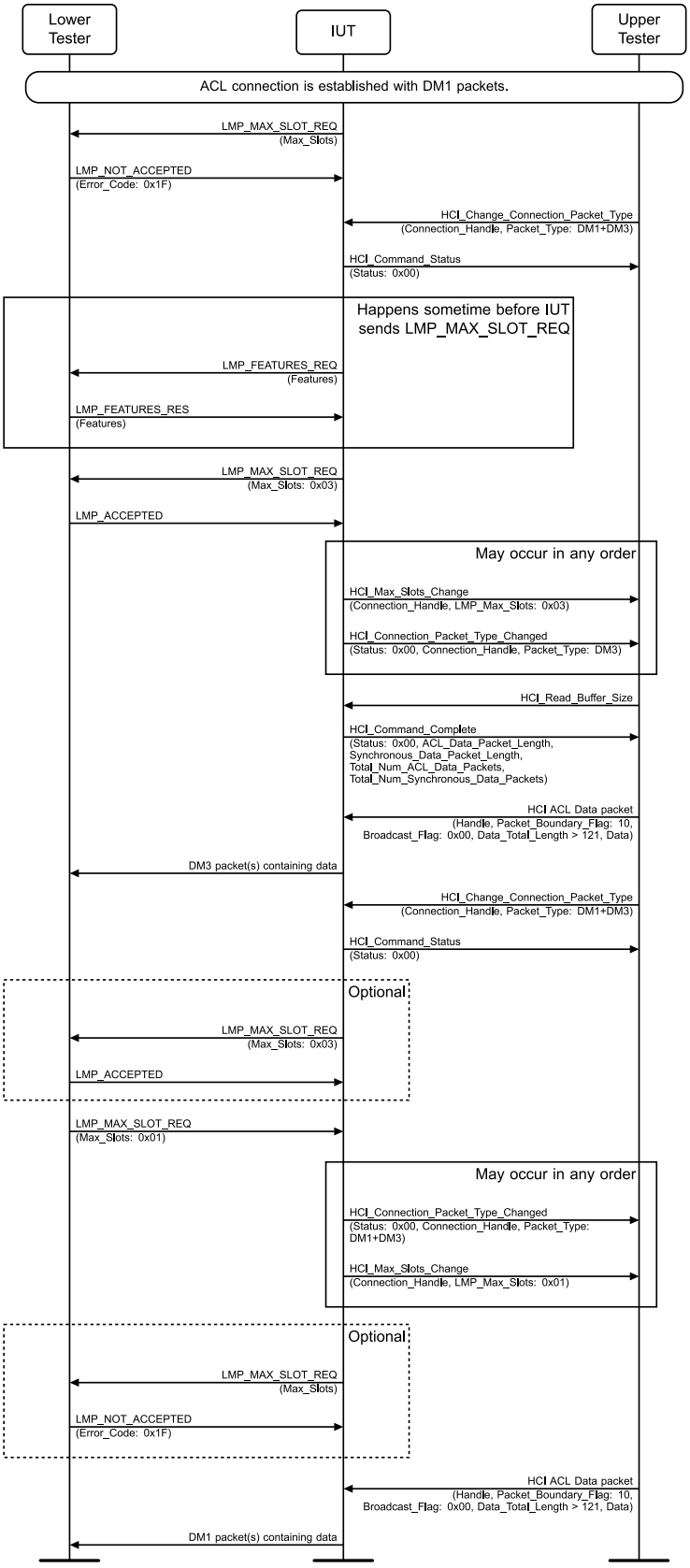


Figure 4.8-84: Request Maximum Slots MSC

1. The IUT sends an LMP_MAX_SLOT_REQ PDU to the Lower Tester with Max_Slots.
2. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_MAX_SLOT_REQ PDU Opcode and Error_Code set to 0x1F (Unspecified Error).
3. The Upper Tester sends an HCI_Change_Connection_Packet_Type command with the Connection_Handle and Packet_Type set to DM1+DM3 and receives a successful HCI_Command_Status event in response.
4. The IUT sends an LMP_FEATURES_REQ PDU to the Lower Tester with the Features parameter.
5. The Lower Tester sends an LMP_FEATURES_RES PDU to the IUT with the Features parameter.
6. The IUT sends an LMP_MAX_SLOT_REQ PDU to the Lower Tester with Max_Slots set to 0x03.
7. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_MAX_SLOT_REQ PDU Opcode.

Steps 8 and 9 can occur in either order.

8. The IUT sends an HCI_Max_Slots_Change event to the Upper Tester with the Connection_Handle and LMP_Max_Slots set to 0x03.
9. The IUT sends a successful HCI_Connection_Packet_Type_Changed event to the Upper Tester with the Connection_Handle and Packet_Type set to DM3.
10. The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event with ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, and Total_Num_Synchronous_Data_Packets.
11. The Upper Tester sends HCI ACL Data packets to the IUT.
12. The IUT sends DM3 data packets to the Lower Tester.
13. The Upper Tester sends an HCI_Change_Connection_Packet_Type command with the Connection_Handle and Packet_Type set to DM1+DM3 and receives a successful HCI_Command_Status event in response.
14. Optionally, the IUT sends an LMP_MAX_SLOT_REQ PDU to the Lower Tester with Max_Slots set to 0x03 and receives an LMP_ACCEPTED PDU with the LMP_MAX_SLOT_REQ PDU Opcode in response.
15. The Lower Tester sends an LMP_MAX_SLOT_REQ PDU to the IUT with Max_Slots set to 0x01.

Steps 16 and 17 can occur in either order.

16. The IUT sends a successful HCI_Connection_Packet_Type_Changed event to the Upper Tester with the Connection_Handle and Packet_Type set to DM1+DM3.
17. The IUT sends an HCI_Max_Slots_Change event to the Upper Tester with the Connection_Handle and LMP_Max_Slots set to 0x01.
18. Optionally, the IUT sends an LMP_MAX_SLOT_REQ PDU to the Lower Tester with Max_Slots and receives an LMP_NOT_ACCEPTED PDU with the LMP_MAX_SLOT_REQ PDU Opcode and Error_Code set to 0x1F (Unspecified Error) in response.
19. The Upper Tester sends HCI ACL Data packets to the IUT.
20. The IUT sends DM1 data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT uses DM3 packets after reception of the LMP_ACCEPTED PDU from the Lower Tester in Step 7.

  The IUT changes to DM1 packets after reception of the LMP_MAX_SLOT PDU from the Lower Tester in Step 15.

- Inconclusive verdict

  The IUT does not initiate sending the LMP_MAX_SLOT_REQ PDU to the Lower Tester to use multi-slot packets.

- Notes

  If DM3 packets are not supported by the IUT, then they are replaced with DH3 or DM5 or DH5.

4.8.23.2 Maximum Slot after a Connection

- Test Purpose

  Verify that the IUT maximum number of slots is 1 after a new connection.

- Reference

  [1] 4.1.10

- Initial Condition

  - See the “Default settings” section.
  - The IUT is in the role indicated in Table 4.8-12 and initiates the service.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Role
LMP/LIH/BV-145-C [Maximum Slot after a Connection as Peripheral]	Peripheral
LMP/LIH/BV-147-C [Maximum Slot after a Connection as Central]	Central

Table 4.8-12: Maximum Slot after a Connection test cases

- Test Procedure

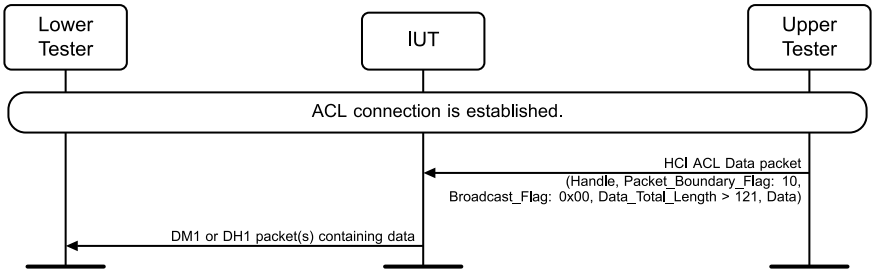


Figure 4.8-85: Maximum Slot after a Connection MSC

1. The Upper Tester sends HCI ACL Data packets to the IUT.
2. The IUT sends DM1 or DH1 data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT uses DM1 or DH1 packets after a new connection.

4.8.24 Paging_Scheme – Both Central and Peripheral

Verify that the IUT declines the Paging_Scheme changes in a correct manner. The role of the IUT is of no importance.

4.8.24.1 Reject Page Mode or Page Scan Negotiation

- Test Purpose

  Verify that the IUT responds to the Lower Tester that it does not support Paging_Scheme negotiation when the Lower Tester tries to negotiate the Page mode or Page Scan mode.

- Reference

  [1] 4.1.9.1, 4.1.9.2

- Initial Condition

  - See the “Default settings” section.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	LMP PDU
LMP/LIH/BV-71-C [Reject Page Mode Negotiation]	LMP_PAGE_MODE_REQ
LMP/LIH/BV-72-C [Reject Page Scan Negotiation]	LMP_PAGE_SCAN_MODE_REQ

Table 4.8-13: Reject Page Mode or Page Scan Negotiation test cases

- Test Procedure

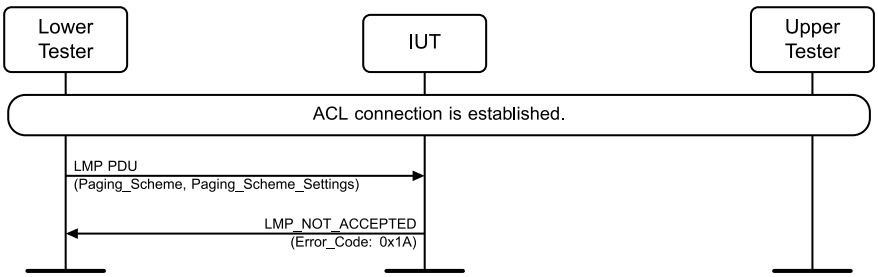


Figure 4.8-86: Reject Page Mode or Page Scan Negotiation MSC

1. The Lower Tester sends the LMP PDU indicated in Table 4.8-13 to the IUT with Paging_Scheme and Paging_Scheme_Settings.
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_PAGE_MODE_REQ PDU Opcode and Error_Code set to 0x1A (Unsupported Remote Feature).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1A (Unsupported Remote Feature) upon reception of the LMP PDU indicated in Table 4.8-13 from the Lower Tester.

4.8.25 Link Supervision

Verify that the IUT can set the Supervision_Timeout.

LMP/LIH/BV-74-C [Set Supervision Timer as Central]

- Test Purpose

  Verify that the IUT sets the supervision timer. The IUT is the Central. The Lower Tester is the Peripheral.

- Reference

  [1] 4.1.6

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Central. The Lower Tester is the Peripheral.
  - An ACL connection has been established.
  - The IUT has to page the Lower Tester to become the Central of the piconet.

- Test Procedure

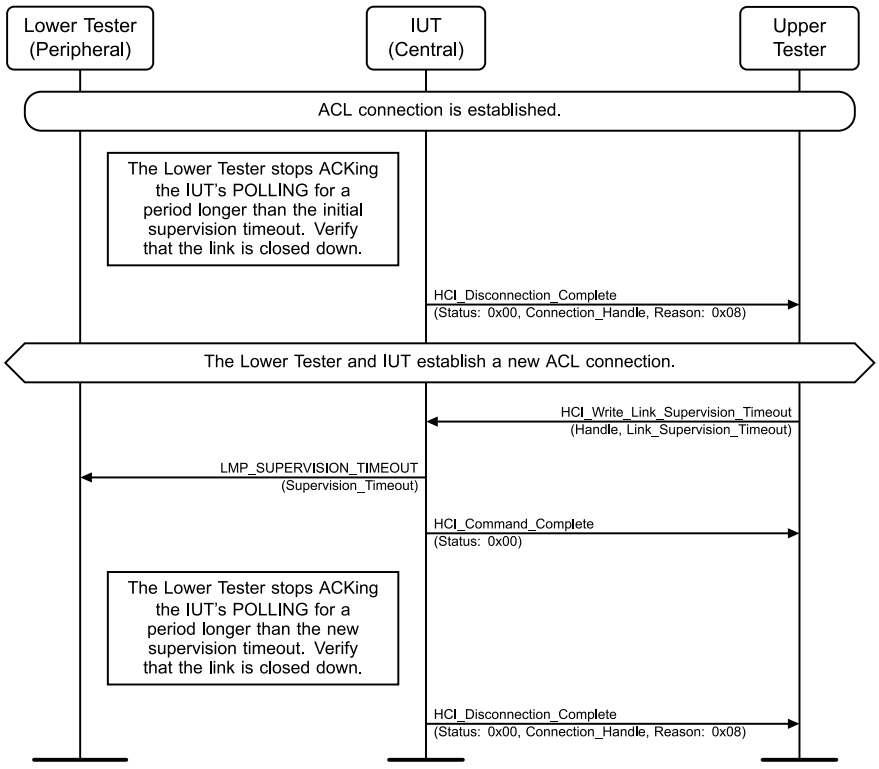


Figure 4.8-87: LMP/LIH/BV-74-C [Set Supervision Timer as Central] MSC

1. The Lower Tester stops ACKing the IUT’s POLL packets for a period longer than the initial supervision timeout.
2. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x08 (Connection Timeout).
3. The Lower Tester and the IUT establish a new ACL connection.

4. The Upper Tester sends an HCI_Write_Link_Supervision_Timeout command to the IUT with the Handle and Link_Supervision_Timeout.
5. The IUT sends an LMP_SUPERVISION_TIMEOUT PDU to the Lower Tester with the Supervision_Timeout.
6. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
7. The Lower Tester stops ACKing the IUT's POLL packets for a period longer than the new supervision timeout.
8. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x08 (Connection Timeout).

- Expected Outcome

  Pass verdict

  The IUT closes down the connection after expiration of the Supervision_Timeout, can change the Supervision_Timeout, and closes the connection after the changed timer expires.

LMP/LIH/BV-126-C [Set Supervision Timer as Peripheral]

- Test Purpose

  Verify that the IUT sends an HCI_Link_Supervision_Timeout_Changed event to the Host when the Central changes the link supervision timeout.

- Reference

  [1] 4.1.6

- Initial Condition

  - See the "Default settings" section.
  - The IUT is the Peripheral. The Lower Tester is the Central.
  - An ACL connection has been established.

- Test Procedure

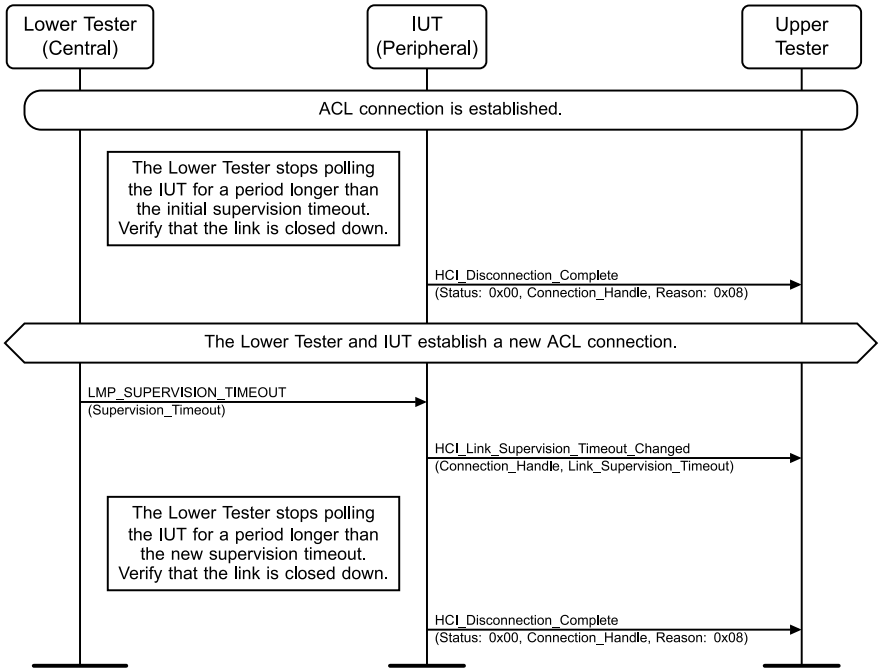


Figure 4.8-88: LMP/LIH/BV-126-C [Set Supervision Timer as Peripheral] MSC

1. The Lower Tester stops ACKing the IUT's POLL packets for a period longer than the initial supervision timeout.
2. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x08 (Connection Timeout).
3. The Lower Tester and the IUT establish a new ACL connection.
4. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU to the IUT with the Supervision_Timeout set to the old supervision timeout value +5 seconds.
5. The IUT sends an HCI_Link_Supervision_Timeout_Changed event to the Upper Tester with the Connection_Handle and Link_Supervision_Timeout.
6. The Lower Tester stops ACKing the IUT's POLL packets for a period longer than the new supervision timeout.
7. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x08 (Connection Timeout).

- Expected Outcome

  Pass verdict

  In Step 5, the IUT sends an HCI_Link_Supervision_Timeout_Changed event to the Upper Tester with the Link_Supervision_Timeout set to the new timeout specified by the Lower Tester in Step 4.

  The IUT closes the connection after expiration of the Supervision Timeout, changes the Supervision Timeout upon notice from the Lower Tester, and closes the connection after the changed timer expires.

4.8.26 Deadlock Avoidance

Verify that the IUT does not run into a deadlock situation.

LMP/LIH/BV-80-C [Avoid Deadlock as Central]

- Test Purpose

  Verify that the IUT does not create a deadlock situation during ACL connection setup.

- Reference

  [1] 2.7, 4.3.4

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Central. The Lower Tester is the Peripheral.
  - No ACL connection has been established.

- Test Procedure

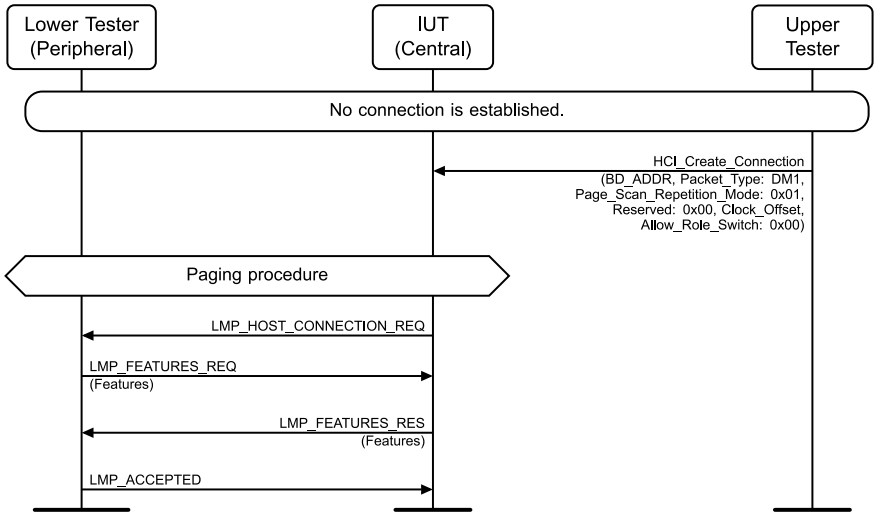


Figure 4.8-89: LMP/LIH/BV-80-C [Avoid Deadlock as Central] MSC

1. The Upper Tester sends an HCI_Create_Connection command to the IUT with the BD_ADDR, Packet_Type set to DM1, Page_Scan_Repetition_Mode set to 0x01 (R1), Reserved set to 0x00, Clock_Offset, and Allow_Role_Switch set to 0x00.
2. The IUT and the Lower Tester execute the paging procedure.
3. The IUT sends an LMP_HOST_CONNECTION_REQ PDU to the Lower Tester.
4. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
5. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
6. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_HOST_CONNECTION_REQ PDU Opcode.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_FEATURES_RES PDU to the Lower Tester before receiving the LMP_ACCEPTED PDU from the Lower Tester.

LMP/LIH/BV-81-C [Avoid Deadlock as Peripheral]

- Test Purpose

  Verify that the IUT does not create a deadlock situation on an active ACL connection.

- Reference

  [1] 2.7, 4.3.4

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central.
  - No ACL connection has been established.
  - The IUT’s supported Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation in the HCI_QoS_Setup command is defined by TSPX_qos_service_type, TSPX_qos_token_rate, TSPX_qos_peak_bandwidth, TSPX_qos_latency, and TSPX_qos_delay_variation, respectively.

- Test Procedure

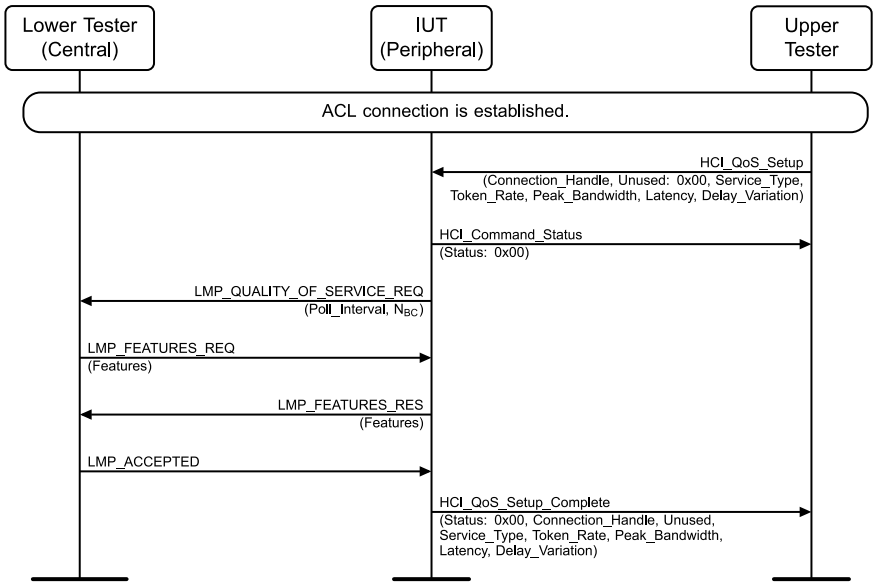


Figure 4.8-90: LMP/LIH/BV-81-C [Avoid Deadlock as Peripheral] MSC

1. The Upper Tester sends an HCI_QoS_Setup command to the IUT with the Connection_Handle, Unused set to 0x00, Service_Type set to TSPX_qos_service_type, Token_Rate set to TSPX_qos_token_rate, Peak_Bandwidth set to TSPX_qos_peak_bandwidth, Latency set to TSPX_qos_latency, and Delay_Variation set to TSPX_qos_delay_variation and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_QUALITY_OF_SERVICE_REQ PDU to the Lower Tester with the Poll_Interval and N_{BC}.

3. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
4. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
5. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_QUALITY_OF_SERVICE_REQ PDU Opcode.
6. The IUT sends a successful HCI_QoS_Setup_Complete event to the Upper Tester with the Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_FEATURES_RES PDU to the Lower Tester before receiving the LMP_ACCEPTED PDU from the Lower Tester.

  The IUT sends a successful HCI_QoS_Setup_Complete event to the Upper Tester.

4.8.27 Test for Devices that do not Support Enhanced Data Rate

Verify that the IUTs that do not support Enhanced Data Rate do not accept Enhanced Data Rate initiation.

LMP/LIH/BV-83-C [Test for Devices that do not support Enhanced Data_Rate]

- Test Purpose

  Verify that the IUT that does not support Enhanced Data Rate ACL does not set up an EDR ACL link upon request from the Lower Tester and that the correct EDR ACL setup denial is used.

- Reference

  [1] 4.1.11

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - An ACL connection has been established; see the “Connection Establishment Lower Tester” preamble.

- Test Procedure

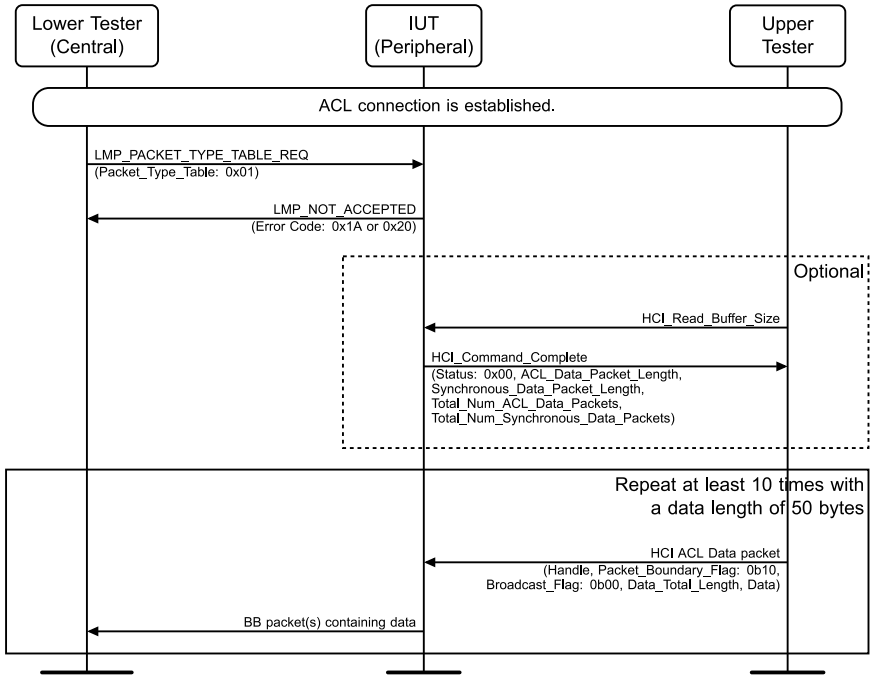


Figure 4.8-91: LMP/LIH/BV-83-C [Test for Devices that do not support Enhanced Data Rate] MSC

1. The Lower Tester sends an LMP_PACKET_TYPE_TABLE_REQ PDU to the IUT with Packet_Type_Table set to 0x01 (2/3 Mb/s).
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_PACKET_TYPE_TABLE_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x1A (Unsupported Remote Feature) or 0x20 (Unsupported LMP Parameter Value / Unsupported LL Parameter Value).
3. Optionally, the Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event with ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, and Total_Num_Synchronous_Data_Packets.

Repeat Steps 4–5 at least 10 times with a data length of 50 bytes.

4. The Upper Tester sends an HCI ACL Data packet to the IUT.
5. The IUT sends BB data packets to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with the Error_Code set to 0x1A (Unsupported Remote Feature) or 0x20 (Unsupported LMP Parameter Value / Unsupported LL Parameter Value) upon reception of the LMP_PACKET_TYPE_TABLE_REQ PDU from the Lower Tester.

4.8.28 Setting up and Removing Enhanced Data_Rate ACL connection

Verify that the IUT can initiate and remove an Enhanced Data Rate ACL link. The IUT is the Peripheral.

Test the behavior of the IUT in relation to the syntactically and contextually correct behavior of the test system.

LMP/LIH/BV-84-C [EDR ACL Link Setup]

- Test Purpose

  Verify that the IUT sets up an EDR ACL link upon request from the Lower Tester and that the correct EDR ACL setup is used.

- Reference

  [1] 4.1.11

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - An ACL connection has been established; see the “Connection Establishment Lower Tester” preamble.

- Test Procedure

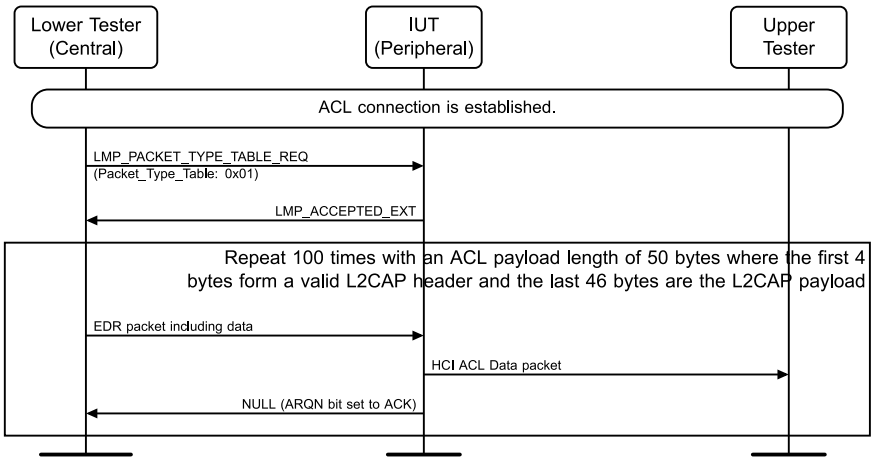


Figure 4.8-92: LMP/LIH/BV-84-C [EDR ACL Link Set Up] MSC

1. The Lower Tester sends an LMP_PACKET_TYPE_TABLE_REQ PDU to the IUT with Packet_Type_Table set to 0x01 (2/3 Mb/s).
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_PACKET_TYPE_TABLE_REQ PDU Extended_Opcode and Escape_Opcode.

Repeat Steps 3–5 100 times with an ACL payload length of 50 bytes where the first 4 bytes form a valid L2CAP header and the last 46 bytes are the L2CAP payload.

3. The Lower Tester sends an EDR packet including data to the IUT.
4. The IUT sends an HCI ACL Data packet to the Upper Tester.
5. The IUT sends a NULL packet to the Lower Tester with the ARQN bit set to ACK.

- Expected Outcome

  Pass verdict

  In Step 2, the IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester.

  At least 90% of the EDR ACL packets are acknowledged and transferred to the Upper Tester.

- Notes

  The IUT can substitute a DM1 packet for any or all of the NULL packets in Step 5.

LMP/LIH/BV-85-C [EDR ACL Link Remove]

- Test Purpose

  Verify that the IUT, upon reception of an EDR packet including data, can either send a NAK or not answer.

- Reference

  [1] 4.1.11

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - An EDR ACL connection has been established; see the “Connection Establishment Lower Tester” preamble.

- Test Procedure

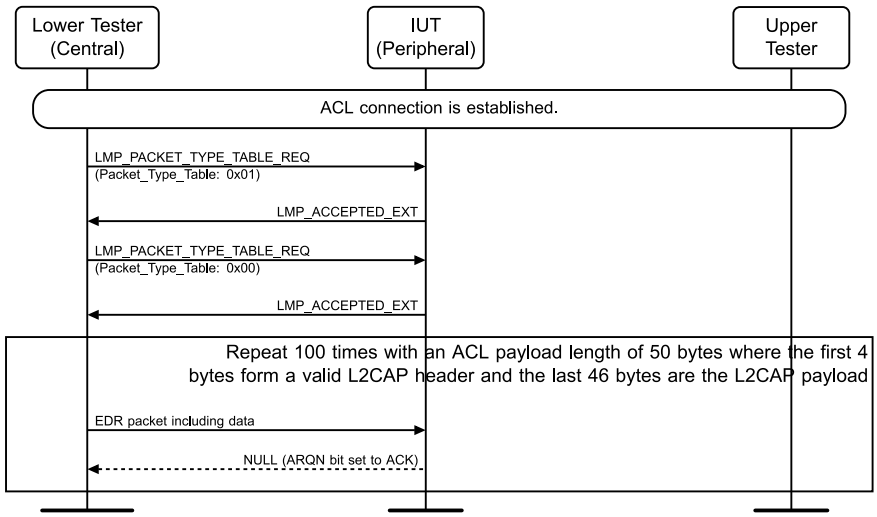


Figure 4.8-93: LMP/LIH/BV-85-C [EDR ACL Link Remove] MSC

1. The Lower Tester sends an LMP_PACKET_TYPE_TABLE_REQ PDU to the IUT with Packet_Type_Table set to 0x01 (2/3 Mb/s).
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_PACKET_TYPE_TABLE_REQ PDU Extended_Opcode and Escape_Opcode.
3. The Lower Tester sends an LMP_PACKET_TYPE_TABLE_REQ PDU to the IUT with Packet_Type_Table set to 0x00 (1 Mb/s only).
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_PACKET_TYPE_TABLE_REQ PDU Extended_Opcode and Escape_Opcode.

Repeat Steps 5–6 10 times with an ACL payload length of 50 bytes where the first 4 bytes form a valid L2CAP header and the last 46 bytes are the L2CAP payload.

5. The Lower Tester sends an EDR packet including data to the IUT.
6. Optionally, the IUT sends a NULL packet to the Lower Tester with the ARQN bit set to NACK.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester after reception from the Lower Tester of the LMP_PACKET_TYPE_TABLE_REQ PDU with Packet_Type_Table set to 0x01.

  The IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester after reception from the Lower Tester of the LMP_PACKET_TYPE_TABLE_REQ PDU with Packet_Type_Table set to 0x00.

  Each EDR data packet sent by the Lower Tester in Step 5 is negatively acknowledged (explicitly or implicitly), and none is delivered to the Upper Tester.

- Notes

  The IUT can substitute a DM1 packet for any or all of the NULL packets in Step 6. Alternatively, the IUT can send no packet at all in response to any or all of the EDR packets.

4.8.29 Setting an Enhanced Data Rate eSCO Connection

Verify that the IUT can initiate and remove an eSCO link.

LMP/LIH/BV-86-C [EDR 2-EV3 eSCO Link Setup]

- Test Purpose

  Verify that the IUT sets up an Enhanced Data Rate 2-EV3 eSCO link upon request from the Lower Tester and that the correct Enhanced Data Rate eSCO setup is used.

- Reference

  [1] 4.6.2

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central and initiates the service.
  - An ACL connection has been established.
  - Whether the IUT requires eSCO data to be provided over HCI to cause eSCO packets to be transmitted is defined by TSPX_hci_esco_data_packets_needed.

- Test Procedure

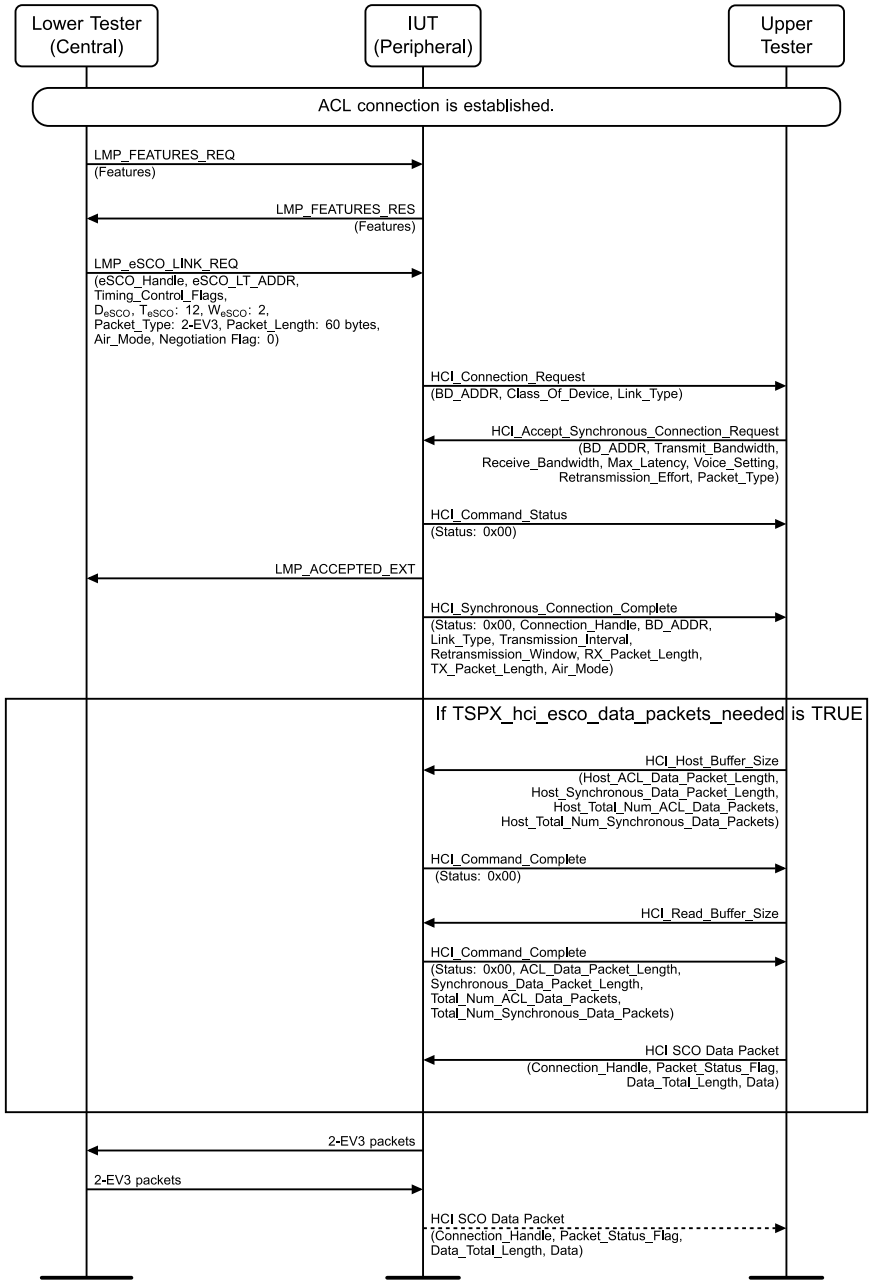


Figure 4.8-94: LMP/LIH/BV-86-C [EDR 2-EV3 eSCO Link Setup] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
2. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
3. The Lower Tester sends an LMP_eSCO_LINK_REQ PDU to the IUT with:
   - eSCO_Handle and eSCO_LT_ADDR set to any valid number
   - Timing_Control_Flags derived from the Central's clock
   - D_{eSCO} set to any number in the range [0, T_{eSCO} – 2]
   - T_{eSCO} set to 12 slots
   - W_{eSCO} set to 2 slots
   - Packet type C→P set to 2-EV3

- Packet type P→C set to 2-EV3
- Packet_Length C→P set to 60 bytes
- Packet_Length P→C set to 60 bytes
- Air_Mode set to any supported Air_Mode
- Negotiation Flag set to 0 (Initiate Negotiation)

4. The IUT sends an HCI_Connection_Request event to the Upper Tester with the BD_ADDR, Class_Of_Device, and Link_Type.
5. The Upper Tester responds to the IUT with an HCI_Accept_Synchronous_Connection_Request command with the BD_ADDR, Transmit_Bandwidth, Receive_Bandwidth, Max_Latency, Voice_Setting, Retransmission_Effort, and Packet_Type and receives a successful HCI_Command_Status event in response.
6. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
7. The IUT sends a successful HCI_Synchronous_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type, Transmission_Interval, Retransmission_Window, RX_Packet_Length, TX_Packet_Length, and Air_Mode.
8. If TSPX_hci_esco_data_packets_needed is TRUE, then perform this step:
   - 8.1 The Upper Tester sends an HCI_Host_Buffer_Size command to the IUT with the Host_ACL_Data_Packet_Length, Host_Synchronous_Data_Packet_Length, Host_Total_Num_ACL_Data_Packets, and Host_Total_Num_Synchronous_Data_Packets and receives a successful HCI_Command_Complete event in response.
   - 8.2 The Upper Tester sends an HCI_Read_Buffer_Size command to the IUT and receives a successful HCI_Command_Complete event with ACL_Data_Packet_Length, Synchronous_Data_Packet_Length, Total_Num_ACL_Data_Packets, and Total_Num_Synchronous_Data_Packets.
   - 8.3 The Upper Tester sends an HCI Synchronous Data packet to the IUT with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.
9. The IUT sends 2-EV3 packets to the Lower Tester.
10. The Lower Tester responds to the IUT with 2-EV3 packets.
11. Optionally, the IUT sends an HCI Synchronous Data packet to the Upper Tester with the Connection_Handle, Packet_Status_Flag, Data_Total_Length, and Data.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester upon reception of the LMP_eSCO_LINK_REQ PDU from the Lower Tester.

  An eSCO link is established and the 2-EV3 packets are transmitted at the eSCO instants and retransmitted inside the retransmission window.

LMP/LIH/BV-87-C [EDR eSCO Link Remove]

- Test Purpose

  Verify that the IUT accepts a request from the Lower Tester to remove the Enhanced Data Rate eSCO link.

- Reference

  [1] 4.6.2

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central.
  - An eSCO link using 2-EV3 packets has been established either by executing LMP/LIH/BV-86-C [EDR 2-EV3 eSCO Link Setup] or by other means.
- Test Procedure

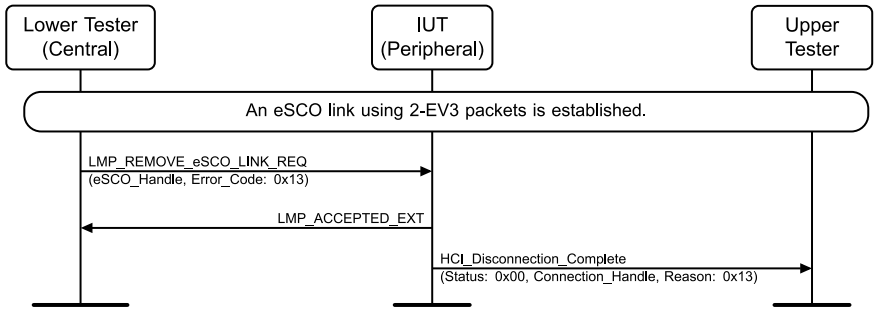


Figure 4.8-95: LMP/LIH/BV-87-C [EDR eSCO Link Remove] MSC

1. The Lower Tester sends an LMP_REMOVE_eSCO_LINK_REQ PDU to the IUT with the eSCO_Handle and Error_Code set to 0x13 (Remote User Terminated Connection).
2. The IUT sends an LMP_ACCEPTED_EXT PDU to the Lower Tester with the LMP_REMOVE_eSCO_LINK_REQ PDU Extended_Opcode and Escape_Opcode.
3. The IUT sends a successful HCI_Disconnection_Complete event to the Upper Tester with the Connection_Handle and Reason set to 0x13 (Remote User Terminated Connection).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED_EXT PDU to the Lower Tester upon reception of the LMP_REMOVE_eSCO_LINK_REQ PDU from the Lower Tester.

  The eSCO link is closed, and eSCO packets are not transmitted.

LMP/LIH/BV-150-C [APB Ignores PDUs Other Than Clock Adjustment]

- Test Purpose

  Verify that the IUT with an APB logical link ignores PDUs except for LMP_CLK_ADJ.

- Reference

  [10] 5.1

- Initial Condition
  - The IUT is the Peripheral and in the CONNECTION state (Active mode, APB link).
  - The Lower Tester is the Central and in the CONNECTION state (Active mode, APB link).

- Test Procedure

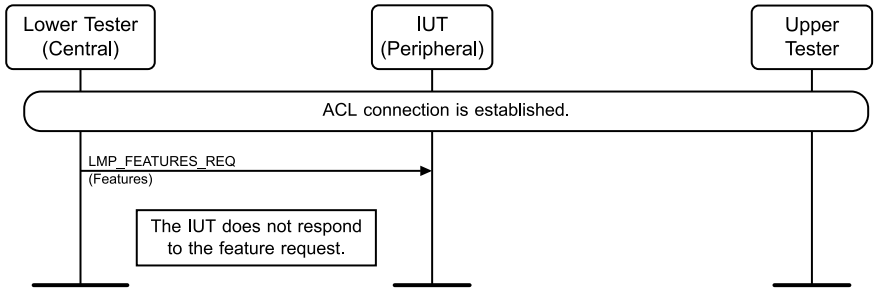


Figure 4.8-96: LMP/LIH/BV-150-C [APB Ignores PDUs Other Than Clock Adjustment] MSC

1. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT on the APB-C link.
2. The IUT does not send an LMP_FEATURES_RES PDU to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT does not respond to a feature request sent on an APB-C link.

4.9 Test Modes

Verify the correct implementation of the Test Modes services.

4.9.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.9.2 Enabled Mode – Peripheral

Verify that the IUT rejects a request to be put into Test mode if not in enabled mode. The IUT is the Peripheral.

LMP/TEM/BV-01-C [Reject Test Mode Request]

- Test Purpose

  Verify that the IUT rejects the request from the Lower Tester to be put into Test mode.

- Reference

  [1] 4.7.1, 4.7.2

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Peripheral and is not locally enabled. The Lower Tester is the Central.
  - An ACL connection has been established.
- Test Procedure

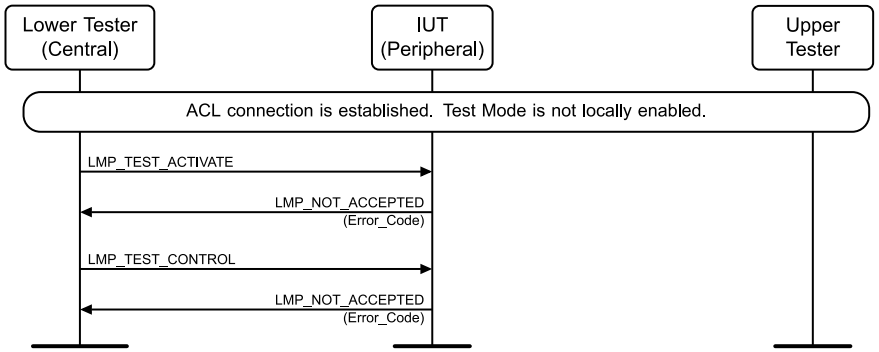


Figure 4.9-1: LMP/TEM/BV-01-C [Reject Test Mode Request] MSC

1. The Lower Tester sends an LMP_TEST_ACTIVATE PDU to the IUT.
2. The IUT responds with an LMP_NOT_ACCEPTED PDU with the LMP_TEST_ACTIVATE PDU Opcode and a valid Error_Code.
3. The Lower Tester sends an LMP_TEST_CONTROL PDU to the IUT with valid values.
4. The IUT responds with an LMP_NOT_ACCEPTED PDU with the LMP_TEST_CONTROL PDU Opcode and a valid Error_Code.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester upon reception of the LMP_TEST_ACTIVATE PDU from the Lower Tester.

  The IUT does not enter Test mode.

4.10 Adaptive Frequency Hopping

4.10.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.10.2 Adaptive Frequency Hopping test cases

LMP/AFH/BV-01-C [AFH Enable – Peripheral]

- Test Purpose

  Verify that the IUT switches from AFH disabled (normal operation) to AFH enabled after the Switch_Instant.

- Reference

  [1] 4.1.4

- Initial Conditions
  - See the “Default settings” section.
  - The Lower Tester and the IUT are in a normal connected state.
  - The Lower Tester is the Central and the IUT is the Peripheral.
  - The IUT is in the AFH disabled state.

- Test Procedure

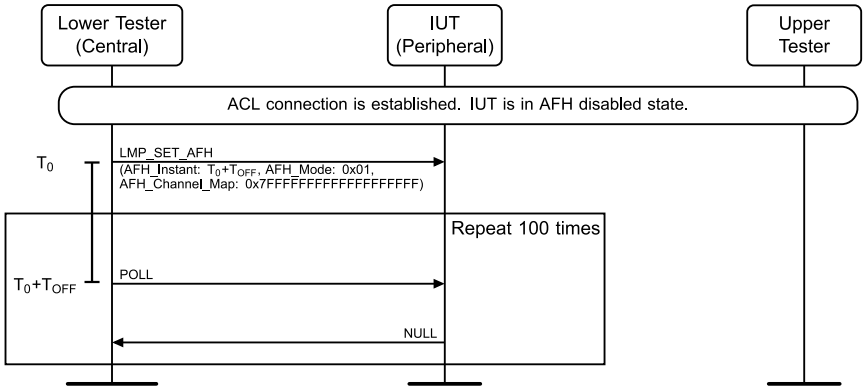


Figure 4.10-1: LMP/AFH/BV-01-C [AFH Enable – Peripheral] MSC

1. The Lower Tester sends an LMP_SET_AFH PDU to the IUT with the AFH_Instant set to $T_0 + T_{OFF}$ where T_0 is when this PDU was sent and $T_{OFF} >= 6*T_{POLL}$, AFH_Mode set to 0x01 (Enabled), and AFH_Channel_Map set to 0x7FFFFFFFFFFFFFFFFFFF (each 1-bit field in the AFH channel map used for the IUT is set to 1 to indicate that all channels are good).
2. At the Switch_Instant $T_0 + T_{OFF}$, the Lower Tester begins sending 100 POLL packets to the IUT on consecutive Central-to-Peripheral slots.
3. The IUT responds to the POLL packets with a NULL packet to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT responds with NULL packets to at least 95% of the Lower Tester's 100 POLL packets.

- Notes

  The test requirement of 95% returned packets is to take into account the imperfect radio path but not to allow for any errors due to an incorrect implementation of the hopping kernel.

  A standardized cable interface is assumed for the baseband connection.

LMP/AFH/BV-02-C [AFH Disable - Peripheral]

- Test Purpose

  Verify that the IUT switches from AFH enabled to AFH disabled after the Switch_Instant.

- Reference

  [1] 4.1.4

- Initial Condition

  - See the "Default settings" section.
  - The Lower Tester and the IUT are in a normal connected state.
  - The Lower Tester is the Central and the IUT is the Peripheral.
  - The IUT is in the AFH enabled state with AFH_Channel_Map = 0x7FFFFFFFFFFFFFFFFFFF (each 1-bit field in the AFH channel map used for the IUT is set to 1 to indicate that all channels are good).

- Test Procedure

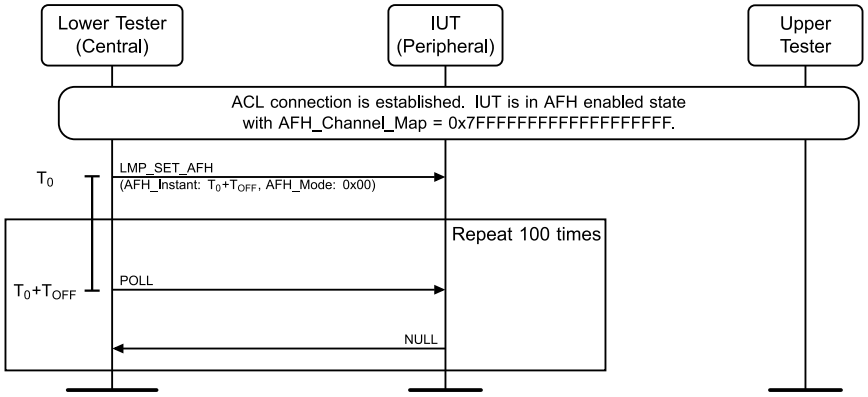


Figure 4.10-2: LMP/AFH/BV-02-C [AFH Disable - Peripheral] MSC

1. The Lower Tester sends an LMP_SET_AFH PDU to the IUT with the AFH_Instant set to $T_0 + T_{OFF}$ where T_0 is when this PDU was sent and $T_{OFF} >= 6*T_{POLL}$ and AFH_Mode set to 0x00 (Disabled).
2. At the Switch_Instant $T_0 + T_{OFF}$, the Lower Tester begins sending 100 POLL packets to the IUT on consecutive Central-to-Peripheral slots.
3. The IUT responds to the POLL packets with a NULL packet to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT responds with NULL packets to at least 95% of the Lower Tester's POLLs.

- Notes

  The test requirement of 95% returned packets is to take into account the imperfect radio path but not to allow for any errors due to an incorrect implementation of the hopping kernel.

  A standardized cable interface is assumed for the baseband connection.

LMP/AFH/BV-03-C [AFH Switch – Peripheral]

- Test Purpose

  Verify that the IUT switches from AFH enabled to AFH enabled with different channel masks after the Switch_Instant.

- Reference

  [1] 4.1.4

- Initial Conditions

  - See "Default settings" section.
  - The Lower Tester and the IUT are in a normal connective state.
  - The Lower Tester is the Central and the IUT is the Peripheral.
  - The IUT is in the AFH enabled state with AFH_Channel_Map = 0x5DDDDDDDFFFF77777777.

- Test Procedure

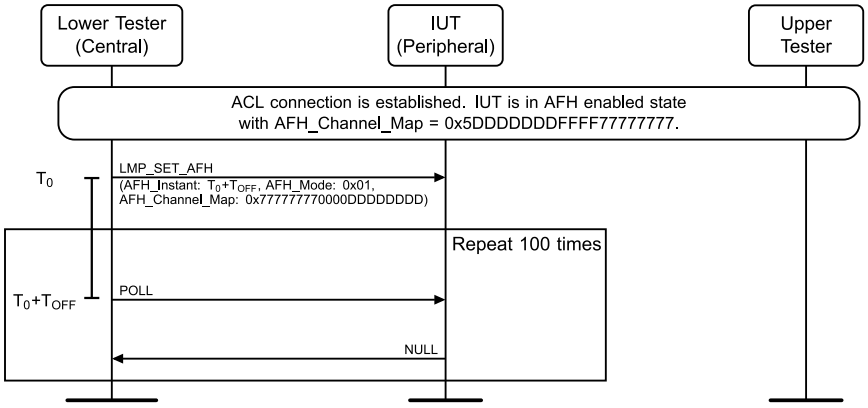


Figure 4.10-3: LMP/AFH/BV-03-C [AFH Switch – Peripheral] MSC

1. The Lower Tester sends an LMP_SET_AFH PDU to the IUT with the AFH_Instant set to $T_0 + T_{OFF}$ where T_0 is when this PDU was sent and $T_{OFF} >= 6*T_{POLL}$, AFH_Mode set to 0x01 (Enabled), and AFH_Channel_Map set to 0x777777770000DDDDDDDD.
2. At the Switch_Instant $T_0 + T_{OFF}$, the Lower Tester begins sending 100 POLL packets to the IUT on consecutive Central-to-Peripheral slots.
3. The IUT responds to the POLL packets with a NULL packet to the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 3, the IUT responds with NULL packets to at least 95% of the Lower Tester's POLLs.

- Notes

  The test requirement of 95% returned packets is to take into account the imperfect radio path but not to allow for any errors due to an incorrect implementation of the hopping kernel.

  A standardized cable interface is assumed for the baseband connection.

LMP/AFH/BV-04-C [Classification Reporting – Normal Operation]

- Test Purpose

  Verify that the IUT starts reporting channel classification messages when requested.

- Reference

  [1] 4.1.4

- Initial Condition

  - See the "Default settings" section.
  - The Lower Tester is the Central and the IUT is the Peripheral.
  - The Lower Tester pages the IUT to become the Central.
  - The Lower Tester and the IUT are in normal connection state.
  - The Upper Tester disables the local channel assessment capabilities of the IUT by sending the HCI_Write_AFH_Channel_Assessment_Mode command.

- Adaptive frequency hopping is enabled by the Lower Tester using all channels: AHS(79).
- The Lower Tester disables Peripheral channel classification by sending the LMP_CHANNEL_CLASSIFICATION_REQ PDU to the IUT with AFH_Reporting set to 0x00 (Disabled).

- Test Procedure

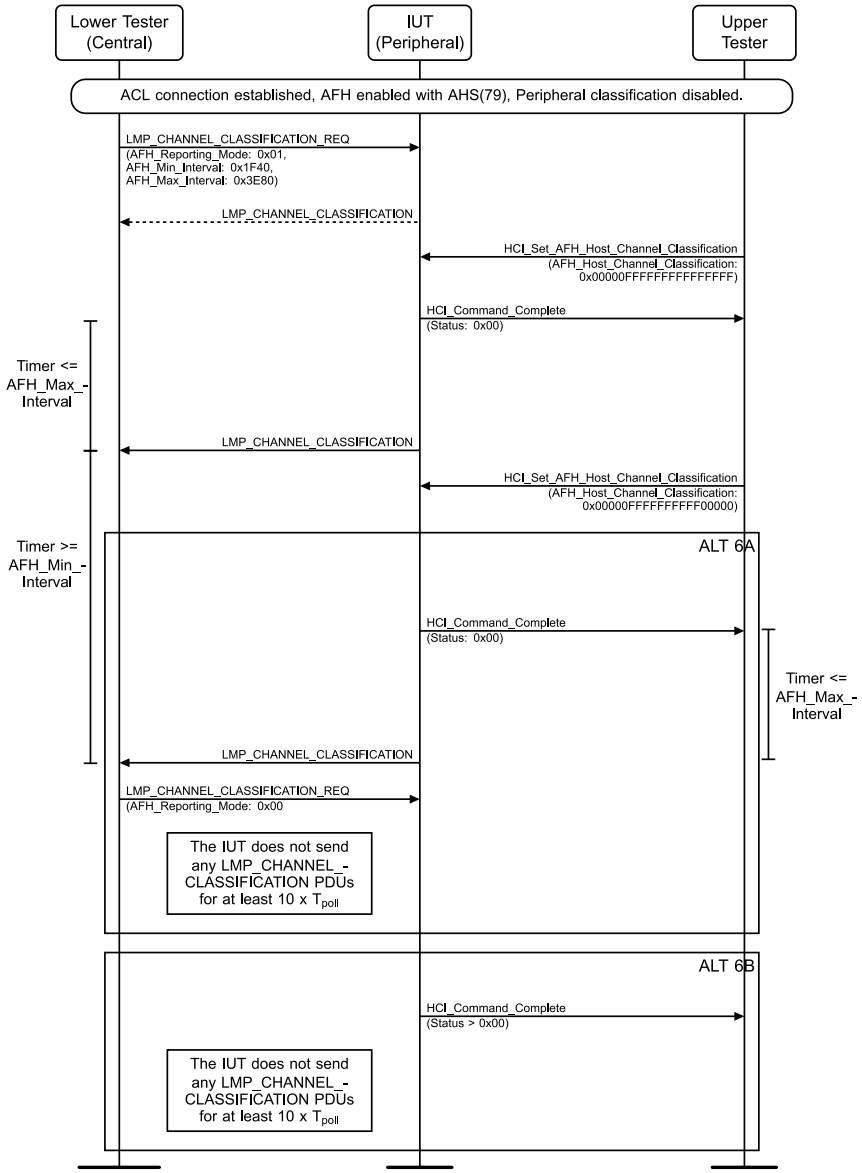


Figure 4.10-4: LMP/AFH/BV-04-C [Classification Reporting – Normal Operation] MSC

1. The Lower Tester sends an LMP_CHANNEL_CLASSIFICATION_REQ PDU to the IUT with AFH_Reporting_Mode set to 0x01 (Enabled), AFH_Min_Interval set to 0x1F40 (5 seconds), and AFH_Max_Interval set to 0x3E80 (10 seconds).
2. Optionally, the IUT responds to the Lower Tester with an LMP_CHANNEL_CLASSIFICATION PDU with the AFH_Channel_Classification.
3. The Upper Tester sends an HCI_Set_AFH_Host_Channel_Classification command to the IUT with AFH_Host_Channel_Classification set to 0x00000FFFFFFFFFFFFFFF and receives a successful HCI_Command_Complete event in response.

4. After Step 3 but before the AFH_Max_Interval, the IUT sends an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester with the AFH_Channel_Classification.
5. The Upper Tester sends an HCI_Set_AFH_Host_Channel_Classification command to the IUT with AFH_Host_Channel_Classification set to 0x00000FFFFFFFFFFFFFFF.
6. Perform either alternative 6A or 6B depending on the IUT's response.
   Alternative 6A (The IUT sends a successful HCI_Command_Complete event to the Upper Tester):
   - 6A.1. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
   - 6A.2. Longer than the AFH_Min_Interval after Step 4 and less than AFH_Max_Interval after Step 6A.1, the IUT sends an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester with the AFH_Channel_Classification.
   - 6A.3. The Lower Tester sends an LMP_CHANNEL_CLASSIFICATION_REQ PDU to the IUT with AFH_Reporting_Mode set to 0x00 (Disabled).
   - 6A.4. The IUT does not send any LMP_CHANNEL_CLASSIFICATION PDUs to the Lower Tester for at least $10*T_{poll}$.

   Alternative 6B (The IUT sends an unsuccessful HCI_Command_Complete event to the Upper Tester):
   - 6B.1. The IUT sends an HCI_Command_Complete event to the Upper Tester with Status > 0x00.
   - 6B.2. The IUT does not send any LMP_CHANNEL_CLASSIFICATION PDUs to the Lower Tester for at least $10*T_{poll}$.

- Expected Outcome

  Pass verdict

  In alternative 6A, the IUT sends the second LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester at least AFH_Min_Interval after the first LMP_CHANNEL_CLASSIFICATION PDU and less than AFH_Max_Interval after sending a successful HCI_Command_Complete event to the Upper Tester.

  In alternative 6B, the IUT sends an HCI_Command_Complete event to the Upper Tester with a non-zero Status. No LMP_CHANNEL_CLASSIFICATION PDU is sent by the IUT for at least $10 \times T_{poll}$.

- Notes

  The Poll_Interval is the default value of 40 slots.

LMP/AFH/BV-05-C [Classification Reporting – After Successful Role Switch]

- Test Purpose

  Verify that the IUT implicitly disables reporting of channel classification after a successful role switch.

- Reference

  [1] 4.1.4

- Initial Condition
  - See the "Default settings" section.
  - The IUT starts as a Central. The Lower Tester starts as a Peripheral.
  - The IUT pages the Lower Tester to become the Central.
  - The Lower Tester and the IUT are in normal connection state.

- The Upper Tester disables the channel classification capabilities of the IUT by sending the HCI_Write_AFH_Channel_Assessment_Mode command.
- Adaptive frequency hopping is enabled by the IUT using any channel map.

- Test Procedure

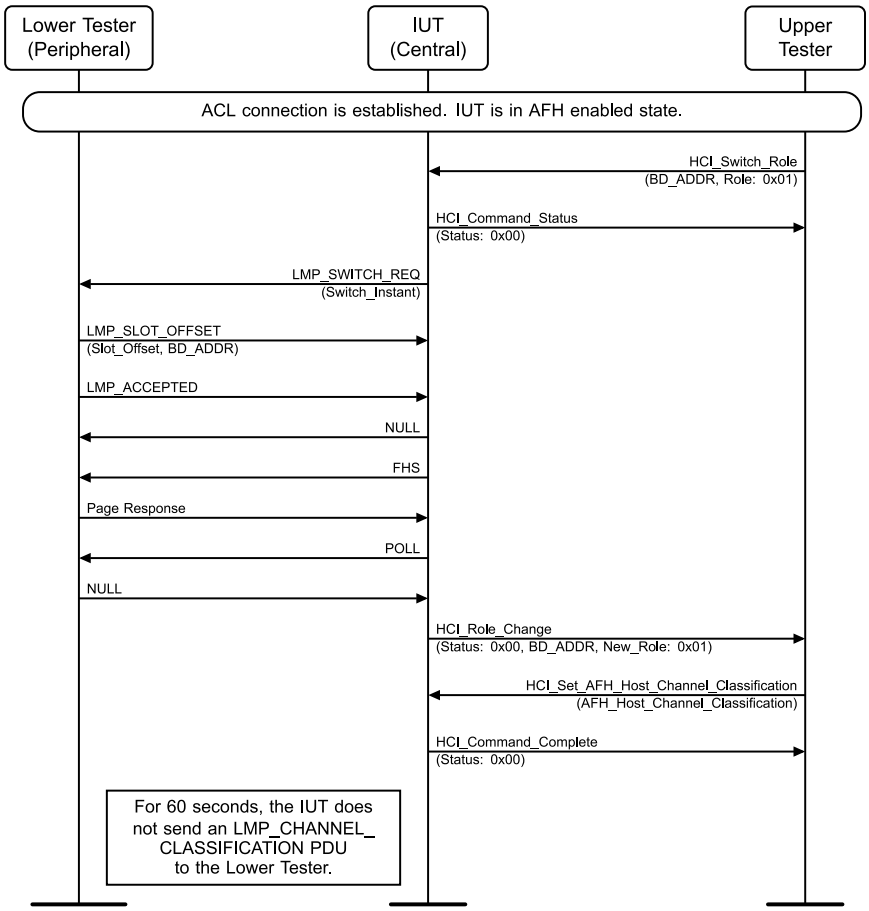


Figure 4.10-5: LMP/AFH/BV-05-C [Classification Reporting – After Successful Role Switch] MSC

1. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x01 (Peripheral) and receives a successful HCI_Command_Status event in response.
2. The IUT sends an LMP_SWITCH_REQ PDU to the Lower Tester with the Switch_Instant.
3. The Lower Tester responds to the IUT with an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
4. The IUT sends a NULL packet and an FHS packet to the Lower Tester.
5. The Lower Tester responds to the IUT with a Page Response packet.
6. The IUT sends a POLL packet to the Lower Tester.
7. The Lower Tester responds to the IUT with a NULL packet.
8. The IUT sends a successful HCI_Role_Change event with the BD_ADDR and New_Role set to 0x01 (Peripheral) to the Upper Tester.
9. The Upper Tester sends an HCI_Set_AFH_Host_Channel_Classification command to the IUT with AFH_Host_Channel_Classification and receives a successful HCI_Command_Complete event in response.
10. For 60 seconds after Step 9, the IUT does not send an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT does not send an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester within 60 seconds after the role switch.

LMP/AFH/BV-06-C [Classification Reporting – After Unsuccessful Role Switch]

- Test Purpose

  Verify that the IUT implicitly restores the channel classification reporting mode after an unsuccessful role switch.

- Reference

  [1] 4.1.5

- Initial Condition
  - See the “Default settings” section.
  - The IUT starts as the Peripheral. The Lower Tester starts as the Central.
  - The Lower Tester pages the IUT to become the Central.
  - The Lower Tester and the IUT are in normal connection state.
  - Adaptive frequency hopping is enabled by the Lower Tester using AHS(79).
  - The Upper Tester disables the channel classification capabilities of the IUT by sending the HCI_Write_AFH_Channel_Assessment_Mode command.

- Test Procedure

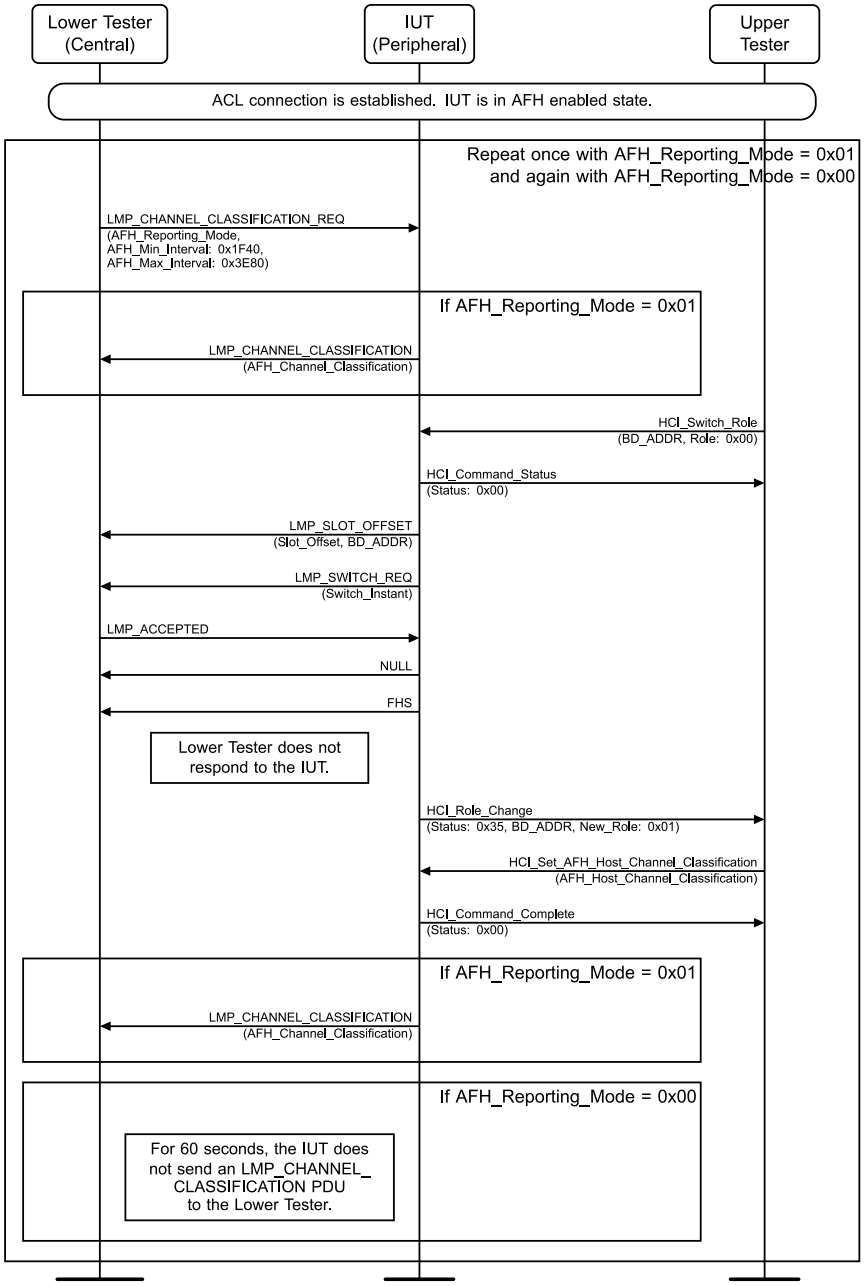


Figure 4.10-6: LMP/AFH/BV-06-C [Classification Reporting – After Unsuccessful Role Switch] MSC

1. The Lower Tester sends an LMP_CHANNEL_CLASSIFICATION_REQ PDU to the IUT with AFH_Reporting_Mode set to 0x01 (Enabled), AFH_Min_Interval set to 5 seconds, and AFH_Max_Interval set to 10 seconds.
2. If AFH_Reporting_Mode is 0x01, the IUT sends an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester with the AFH_Channel_Classification.
3. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
4. The IUT sends an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant to the Lower Tester.
5. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
6. The IUT sends a NULL packet and an FHS packet to the Lower Tester.
7. The Lower Tester does not respond to the IUT.
8. The IUT sends an HCI_Role_Change event with the Status set to 0x35 (Role Switch Failed), BD_ADDR, and New_Role set to 0x01 (Peripheral) to the Upper Tester.
9. The Upper Tester sends an HCI_Set_AFH_Host_Channel_Classification command to the IUT with AFH_Host_Channel_Classification and receives a successful HCI_Command_Complete event in response.
10. If AFH_Reporting_Mode is 0x01, the IUT sends an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester with the AFH_Channel_Classification. Otherwise, the IUT does not send an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester for 60 seconds.
11. Repeat Steps 1–10 except AFH_Reporting_Mode in Step 1 is set to 0x00 (Disabled).

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester after the HCI_Set_AFH_Host_Channel_Classification command from the Upper Tester when AFH_Reporting_Mode is enabled and does not transmit the LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester when AFH_Reporting_Mode is disabled.

LMP/AFH/BV-08-C [Classification Reporting – After Unhold]

- Test Purpose

  Verify that the IUT implicitly restores the channel classification reporting mode after a successful unhold.

- Reference

  [1] 4.1.5

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Peripheral. The Lower Tester is the Central.
  - The Lower Tester pages the IUT to become the Central.
  - The Lower Tester and the IUT are in normal connection state.
  - Adaptive frequency hopping is enabled by the Lower Tester using AHS(79).
  - The Upper Tester disables the channel classification capabilities of the IUT by sending the HCI_Write_AFH_Channel_Assessment_Mode command.
  - The Lower Tester enables classification reporting in the IUT.

- Test Procedure

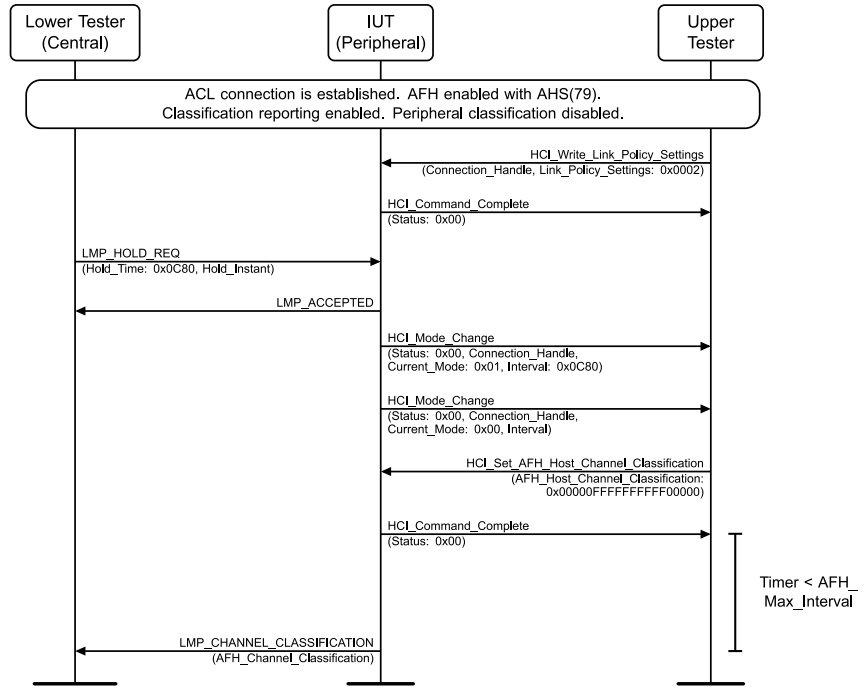


Figure 4.10-7: LMP/AFH/BV-08-C [Classification Reporting – After Unhold] MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0002 (Hold Mode) and receives a successful HCI_Command_Complete event in response.
2. The Lower Tester sends an LMP_HOLD_REQ PDU to the IUT with Hold_Time set to 0x0C80 and Hold_Instant.
3. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_HOLD_REQ PDU Opcode.
4. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x01 (Hold Mode), and Interval set to 0x0C80.
5. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.
6. The Upper Tester sends an HCI_Set_AFH_Host_Channel_Classification command to the IUT with AFH_Host_Channel_Classification set to 0x00000FFFFFFFFF00000 and receives a successful HCI_Command_Complete event in response.
7. Less than AFH_Max_Interval from Step 6, the IUT sends an LMP_CHANNEL_CLASSIFICATION PDU to the Lower Tester with the AFH_Channel_Classification.

- Expected Outcome

  Pass verdict

  Upon reception of the HCI_Set_AFH_Host_Channel_Classification command from the Upper Tester, the IUT sends the LMP_CHANNEL_CLASSIFICATION PDU before AFH_Max_Interval.

LMP/AFH/BV-09-C [Peripheral device does not send an LMP_SET_AFH PDU – after successful role switch]

- Test Purpose

  Verify that the IUT does not send an LMP_SET_AFH PDU after a successful role switch in which the IUT becomes the Peripheral after the role switch.

- Reference

  [1] 4.1.4.1

- Initial Condition

  - The IUT starts as the Central. The Lower Tester starts as the Peripheral.
  - The IUT pages the Lower Tester to become the Central.
  - The Lower Tester and the IUT are in normal connection state.

- Test Procedure

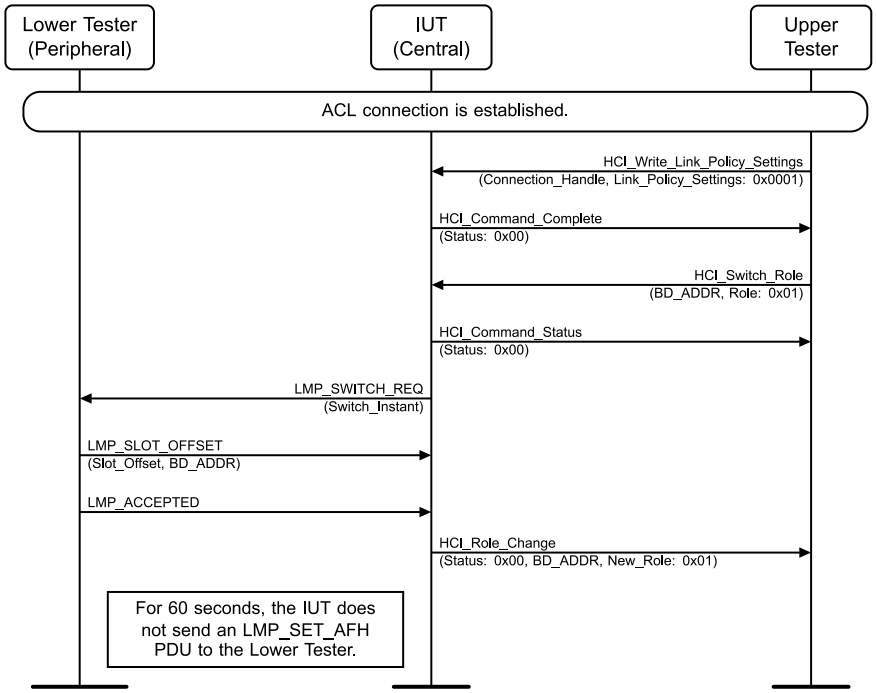


Figure 4.10-8: LMP/AFH/BV-09-C [Peripheral device does not send a LMP_SET_AFH PDU -after successful role switch] MSC

1. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0001 (Enable Role Switch) and receives a successful HCI_Command_Complete event in response.
2. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x01 (Peripheral) and receives a successful HCI_Command_Status event in response.
3. The IUT sends an LMP_SWITCH_REQ PDU to the Lower Tester with the Switch_Instant.
4. The Lower Tester responds to the IUT with an LMP_SLOT_OFFSET PDU with the Slot_Offset and BD_ADDR and an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
5. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x01 (Peripheral).
6. For 60 seconds after Step 5, the IUT does not send an LMP_SET_AFH PDU to the Lower Tester.

- Expected Outcome

  Pass verdict

  The IUT does not send an LMP_SET_AFH PDU to the Lower Tester within 60 seconds after the role switch.

4.11 Simple Pairing procedures

Verify the Simple Pairing procedures.

4.11.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.11.2 Backward Compatibility procedures

LMP/SP/BV-01-C [Secure Simple Pairing Capable Controller - Pairing - IUT Initiator]

- Test Purpose

  Verify that the IUT initiates legacy pairing when the remote Controller does not have the Secure Simple Pairing LMP feature bit set.

- Reference

  [1] 4.1.4.1, 4.3.4

- Initial Condition

  - See the “Connection Establishment IUT Central” preamble and the “Secure Simple Pairing” default settings.
  - The IUT is the Initiator. The Lower Tester is the Responder.

- The Lower Tester does not support Secure Simple Pairing.
- The IUT has the Secure Simple Pairing feature (Controller Support and Host Support) Link Manager bits set.

- Test Procedure

  Execute the test procedure of LMP/AUT/BV-52-C [Pairing, IUT Initiator – HCI Command Required to Pair].

- Expected Outcome

  Pass verdict

  The IUT sends the correct LMP_IN_RAND PDU.

  The IUT sends the correct LMP_COMB_KEY PDU.

  The correct Link Key is created and checked by an authentication (SRES is checked).

LMP/SP/BV-02-C [Secure Simple Pairing Capable Controller - Pairing - IUT Responder]

- Test Purpose

  Verify that the IUT initiates legacy pairing when the remote device does not have the Secure Simple Pairing LMP feature (Controller Support and Host Support) bits set.

- Reference

  [1] 4.2.2, 4.3.4

- Initial Condition
  - See the “Connection Establishment Lower Tester” preamble and the “Secure Simple Pairing” default settings.
  - The Lower Tester is the Initiator. The IUT is the Responder.
  - The Lower Tester does not support Secure Simple Pairing.
  - The IUT has the Secure Simple Pairing feature (Controller Support and Host Support) Link Manager bits set.

- Test Procedure

  Execute the test procedure of LMP/AUT/BV-03-C [Create Link Key].

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_ACCEPTED PDU to the Lower Tester with the LMP_IN_RAND PDU Opcode upon reception of the LMP_IN_RAND PDU from the Lower Tester.

  The IUT sends the LMP_COMB_KEY PDU to the Lower Tester upon reception of the LMP_COMB_KEY PDU from the Lower Tester.

LMP/SP/BV-03-C [Secure Simple Pairing Capable Controller - Legacy Host- IUT Initiator]

- Test Purpose

  Verify that the IUT initiates pairing when the local Host does not set the Secure Simple Pairing Mode to enabled.

- Reference

  [1] 4.2.7

- Initial Condition

  - Use the “Authentication” default settings.
  - An ACL connection has been established.
  - The IUT is the Initiator. The Lower Tester is the Responder.

- Test Procedure

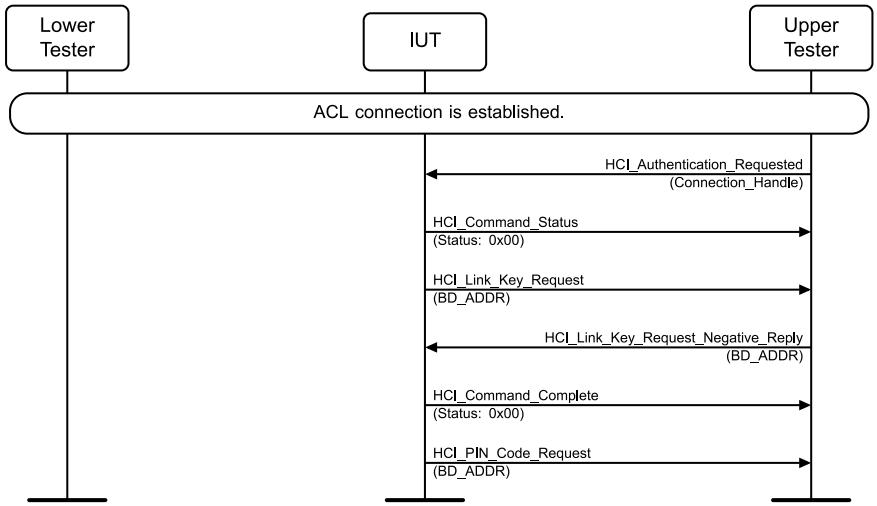


Figure 4.11-1: LMP/SP/BV-03-C [Secure Simple Pairing Capable Controller - Legacy Host- IUT Initiator] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR of the Lower Tester.

- Expected Outcome

  Pass verdict

  In Step 4, the IUT sends an HCI_PIN_Code_Request event to the Upper Tester.

  The IUT does not send an HCI_IO_Capability_Request event to the Upper Tester.

  The IUT does not set the Secure Simple Pairing Mode (Host support) bit.

LMP/SP/BV-04-C [Secure Simple Pairing Capable Controller - Legacy Host - IUT Responder]

- Test Purpose

  Verify that the IUT responds to pairing when the local Host has not set the support Secure Simple Pairing Mode to enabled.

- Reference

  [1] 4.2.7

- Initial Condition

  - Use the “Authentication” default settings.
  - The IUT is the Responder. The Lower Tester is the Initiator.
  - The Lower Tester uses Secure Simple Pairing (feature bit set to 1).
  - An ACL connection has been established.

- Test Procedure

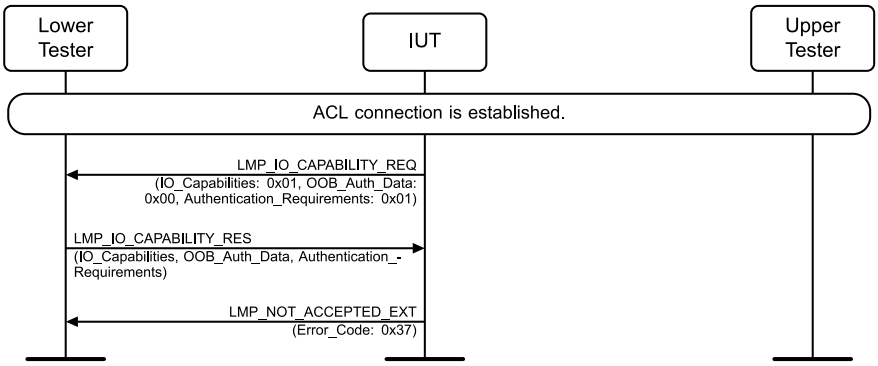


Figure 4.11-2: LMP/SP/BV-04-C [Secure Simple Pairing Capable Controller - Legacy Host - IUT Responder] MSC

1. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities, OOB_Auth_Data, and Authentication_Requirements.
3. The IUT sends an LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with the LMP_IO_CAPABILITY_RES PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x37 (Secure Simple Pairing not Supported by Host).

- Expected Outcome

  Pass verdict

  The IUT responds to the LMP_IO_CAPABILITY_RES PDU sent by the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the Error_Code set to 0x37 (Secure Simple Pairing not Supported by Host).

  The IUT does not set the Secure Simple Pairing Mode (Host support) bit.

LMP/SP/BV-05-C [Secure Simple Pairing Capable Controller - Legacy Remote Host - IUT Initiator]

- Test Purpose

  Verify that the IUT initiates pairing when the local Host sets the Secure Simple Pairing Mode to enabled and the remote Controller's LMP feature bits indicate support for Secure Simple Pairing in the Controller but not in the Host.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the "Connection Establishment IUT Central" preamble and the "Secure Simple Pairing" default settings.
  - The IUT is the Initiator. The Lower Tester is the Responder.

- Test Procedure

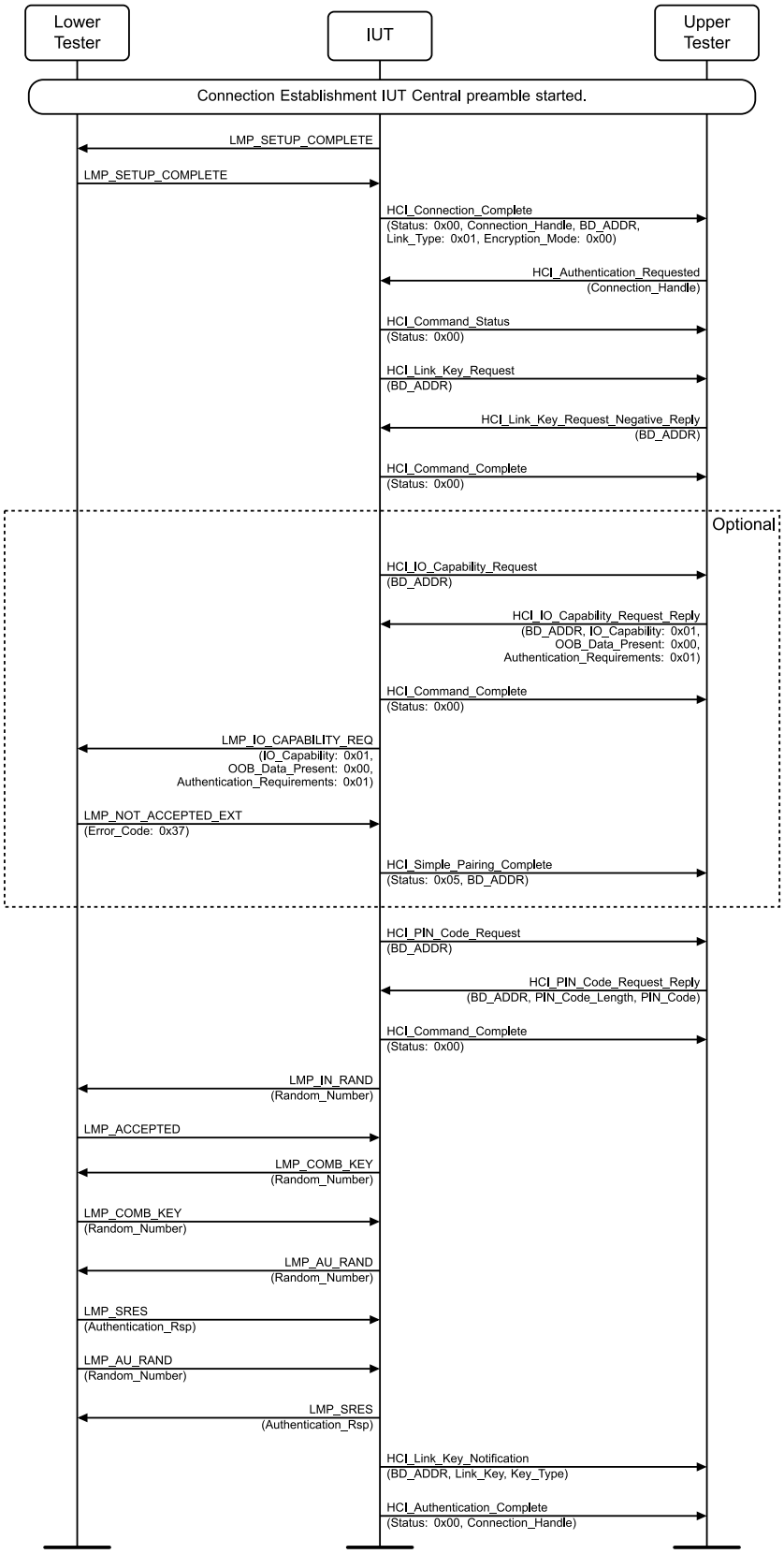


Figure 4.11-3: LMP/SP/BV-05-C [Secure Simple Pairing Capable Controller - Legacy Remote Host - IUT Initiator] MSC

1. The IUT sends an LMP_SETUP_COMPLETE PDU to the Lower Tester.
2. The Lower Tester responds to the IUT with an LMP_SETUP_COMPLETE PDU.
3. The IUT sends a successful HCI_Connection_Complete event to the Upper Tester with the Connection_Handle, BD_ADDR, Link_Type set to 0x01 (ACL), and Encryption_Mode set to 0x00 (Disabled).
4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
6. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.

Steps 7–11 are optional to execute.

7. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
8. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
9. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
10. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_IO_CAPABILITY_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x37 (Secure Simple Pairing not Supported by Host).
11. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.
12. The IUT sends an HCI_PIN_Code_Request event to the Upper Tester with the BD_ADDR.
13. The Upper Tester responds to the IUT with an HCI_PIN_Code_Request_Reply command with the BD_ADDR, PIN_Code_Length, and PIN_Code and receives a successful HCI_Command_Complete event in response.
14. The IUT sends an LMP_IN_RAND PDU to the Lower Tester with a Random_Number.
15. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_IN_RAND PDU Opcode.
16. The IUT sends an LMP_COMB_KEY PDU to the Lower Tester with a Random_Number.
17. The Lower Tester responds to the IUT with an LMP_COMB_KEY PDU with a Random_Number.
18. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
19. The Lower Tester responds to the IUT with an LMP_SRES PDU with the Authentication_Rsp and an LMP_AU_RAND PDU with a Random_Number.
20. The IUT sends an LMP_SRES PDU to the Lower Tester with the Authentication_Rsp.
21. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with the BD_ADDR, Link_Key, and Key_Type. A warning is sent if the HCI_Link_Key_Notification is sent before Step 20.
22. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 12, the IUT sends the HCI_PIN_Code_Request event to the Upper Tester.

  In Step 14, the IUT sends the LMP_IN_RAND PDU to the Lower Tester.

  In Step 16, the IUT sends the correct LMP_COMB_KEY PDU to the Lower Tester.

  The correct Link Key is created and checked by an authentication (SRES is checked).

4.11.2.1 Secure Connections Capable Controller – Legacy Host – IUT Initiator

- Test Purpose

  Verify that the IUT reports the correct Key_Type to a Legacy Host at the end of a successful Secure Simple Pairing using the P-192 elliptic curve using the Numeric Comparison protocol that generates an unauthenticated or authenticated link key.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the Initiator and it does not matter if the IUT is the Central or the Peripheral.
  - The Upper Tester does not set the Secure Connections Host Support to enabled.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Authentication_Requirements	Key_Type
LMP/SP/BV-37-C [Secure Connections Capable Controller – Legacy Host – IUT Initiator – Unauthenticated Link Key]	0x00 (MITM Protection Not Required – No Bonding)	0x04 (Unauthenticated Combination Key generated from P-192)
LMP/SP/BV-38-C [Secure Connections Capable Controller – Legacy Host – IUT Initiator – Authenticated Link Key]	0x01 (MITM Protection Required – No Bonding)	0x05 (Authenticated Combination Key generated from P-192)

Table 4.11-1: Secure Connections Capable Controller – Legacy Host – IUT Initiator test cases

- Test Procedure

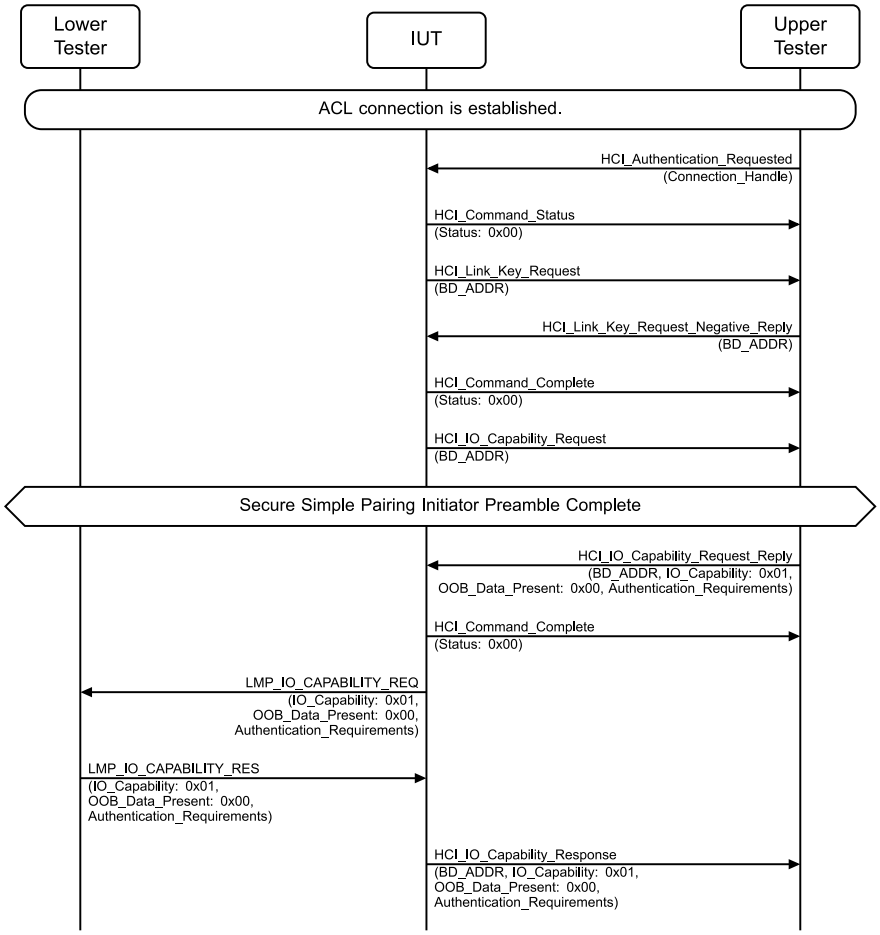


Figure 4.11-4: Secure Connections Capable Controller – Legacy Host – IUT Initiator MSC – Page 1 of 3

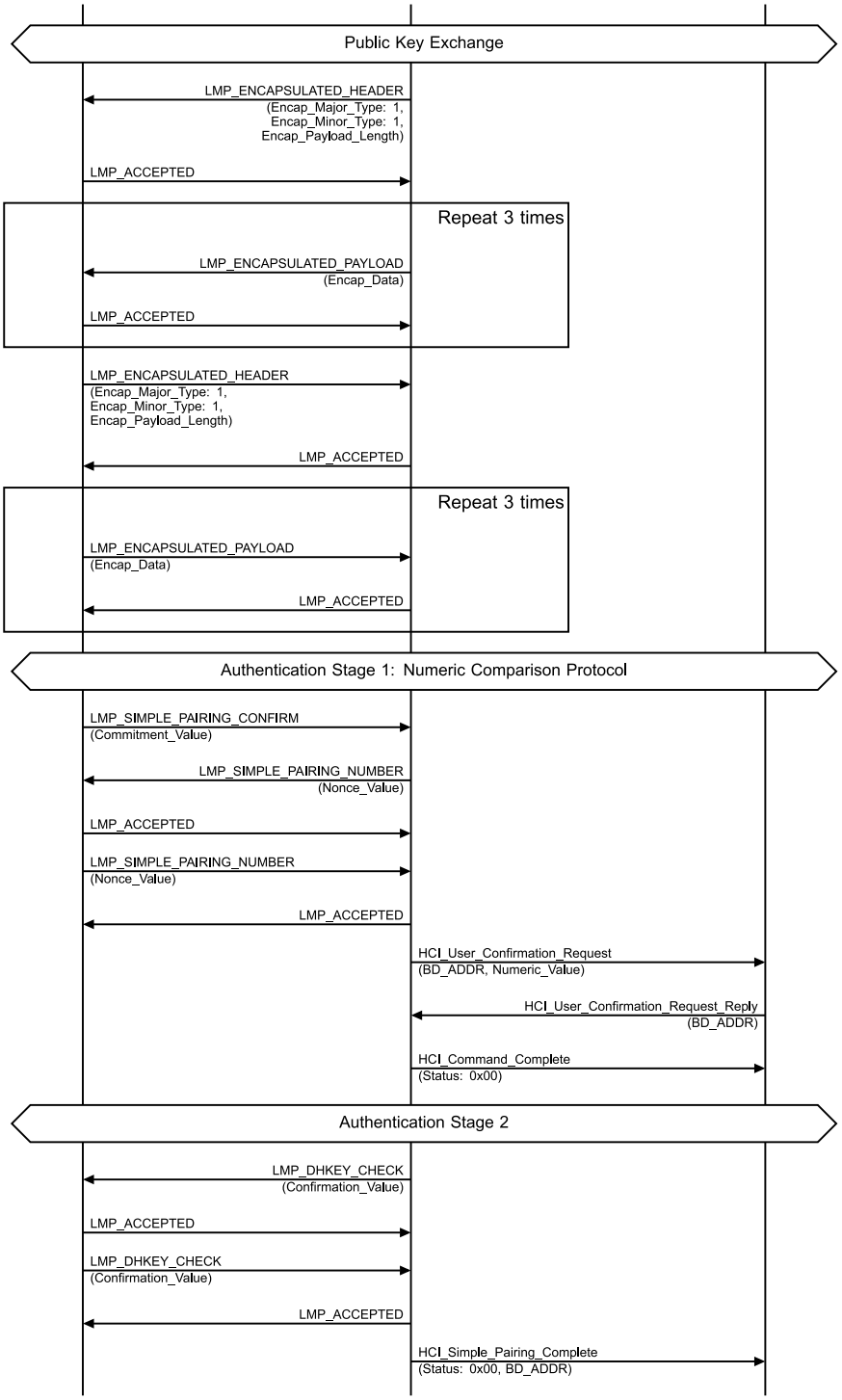


Figure 4.11-5: Secure Connections Capable Controller – Legacy Host – IUT Initiator MSC – Page 2 of 3

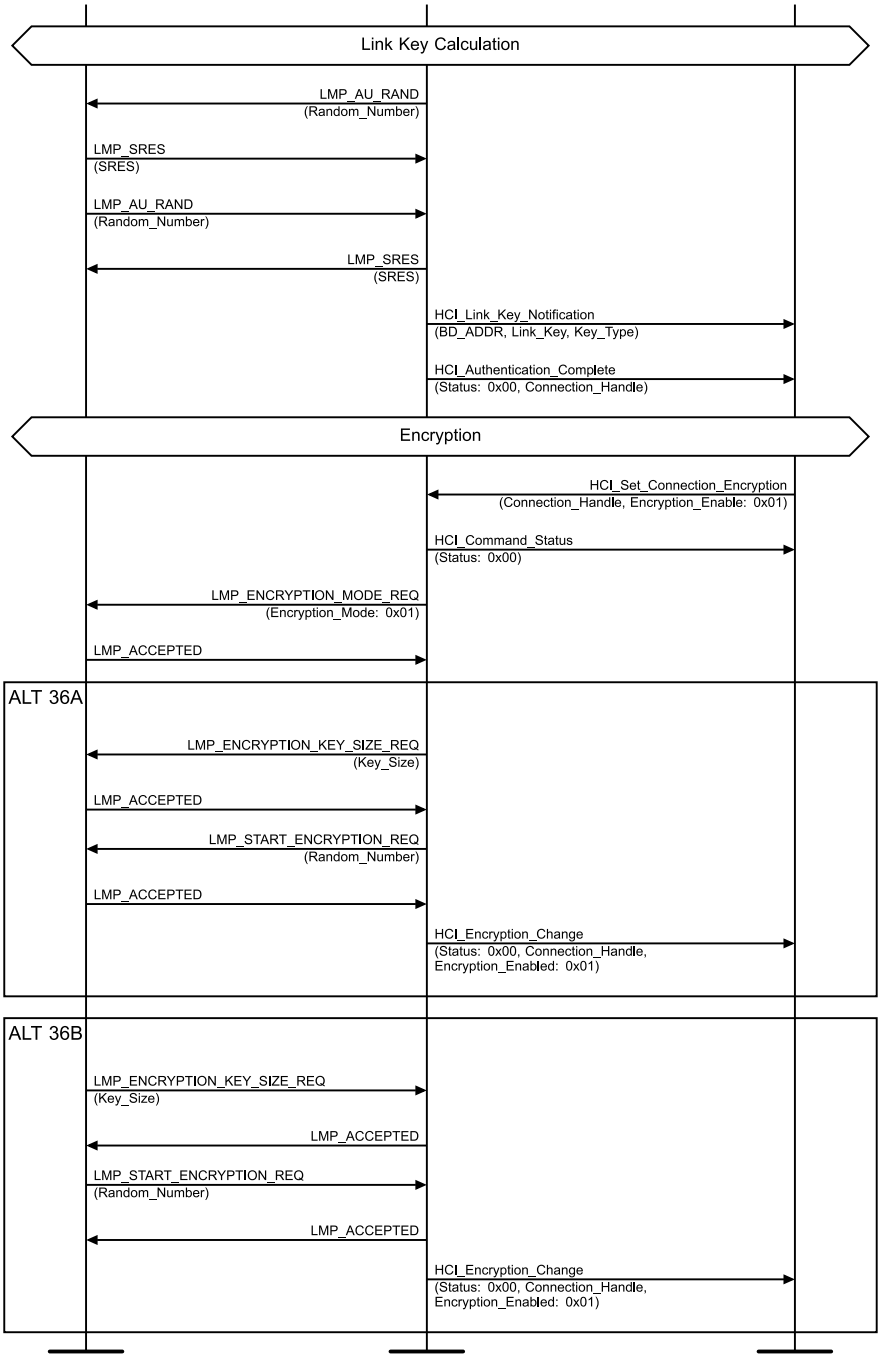


Figure 4.11-6: Secure Connections Capable Controller – Legacy Host – IUT Initiator MSC – Page 3 of 3

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB

authentication data not present), and Authentication_Requirements set to the value indicated in Table 4.11-1 and receives a successful HCI_Command_Complete event in response.

6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to the value indicated in Table 4.11-1.
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to the value indicated in Table 4.11-1.
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to the value indicated in Table 4.11-1.
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 1, and Encap_Payload_Length.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 1, and Encap_Payload_Length.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
18. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
23. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
24. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
25. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
26. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.

27. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
28. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
29. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
30. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
31. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
32. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to the value in Table 4.11-1 and a successful HCI_Authentication_Complete event with the Connection_Handle.
33. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response.
34. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
35. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
36. Perform either alternative 36A or 36B depending on the IUT's role.

    Alternative 36A (The IUT is the Central):

    36A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.

    36A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    36A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

    36A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    36A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

    Alternative 36B (The IUT is the Peripheral):

    36B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

    36B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    36B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

    36B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    36B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

- Expected Outcome

  Pass verdict

  In Step 22, the IUT sends the HCI_User_Confirmation_Request event to the Upper Tester with the same value calculated by the Lower Tester.

  In Step 24, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

In Step 28, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

In Step 32, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is set to the value indicated in Table 4.11-1.

LMP/SP/BV-39-C [Secure Connections Capable Controller – Host has no P256 OOB data available – IUT Initiator – OOB]

- Test Purpose

  Verify that the IUT switches to the Numeric Comparison association model when the Upper Tester indicates that it only has P-192 OOB data from the remote device available. Verify that the IUT reports the correct Key_Type to the Upper Tester at the end of a successful Secure Simple Pairing using the P-256 elliptic curve.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Initiator, and it does not matter if the IUT is the Central or the Peripheral.
  - An ACL connection has been established.

- Test Procedure

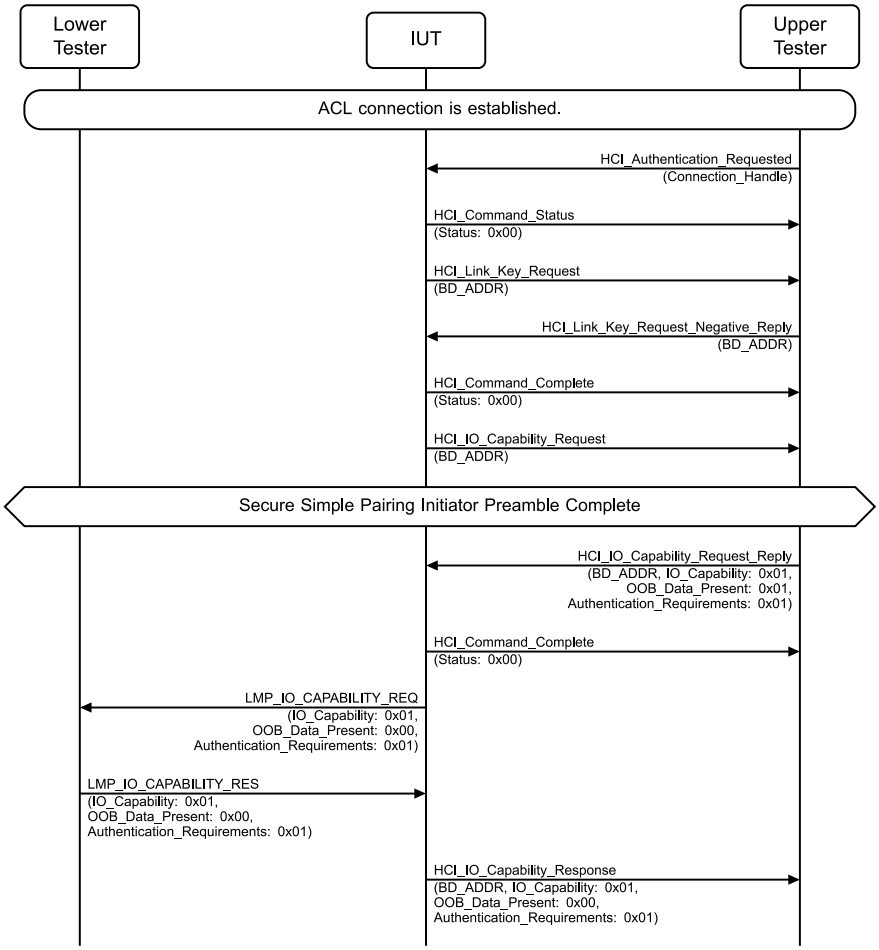


Figure 4.11-7: LMP/SP/BV-39-C [Secure Connections Capable Controller – Host has no P256 OOB data available – IUT Initiator – OOB] MSC – Page 1 of 3

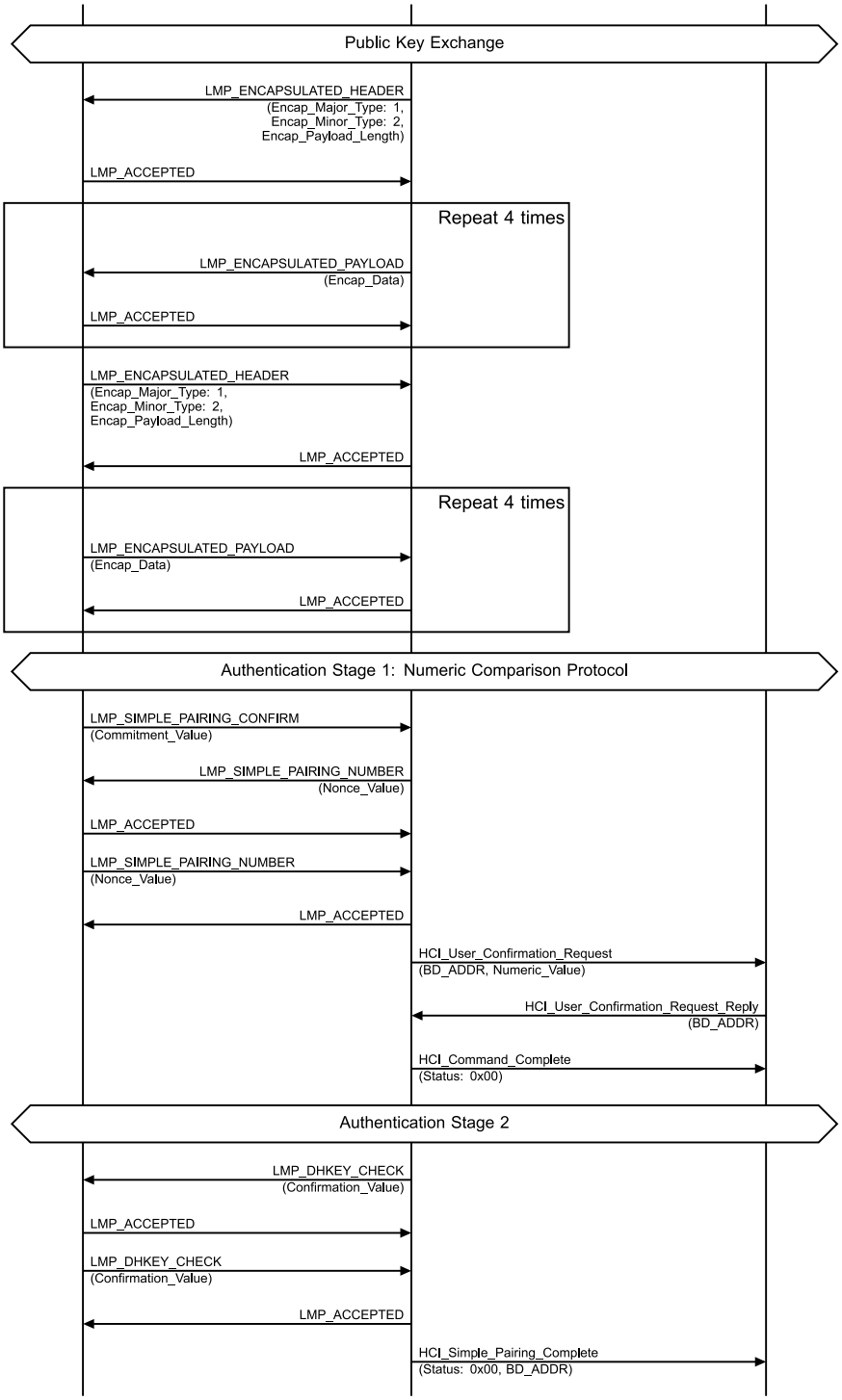


Figure 4.11-8: LMP/SP/BV-39-C [Secure Connections Capable Controller – Host has no P256 OOB data available – IUT Initiator – OOB] MSC – Page 2 of 3

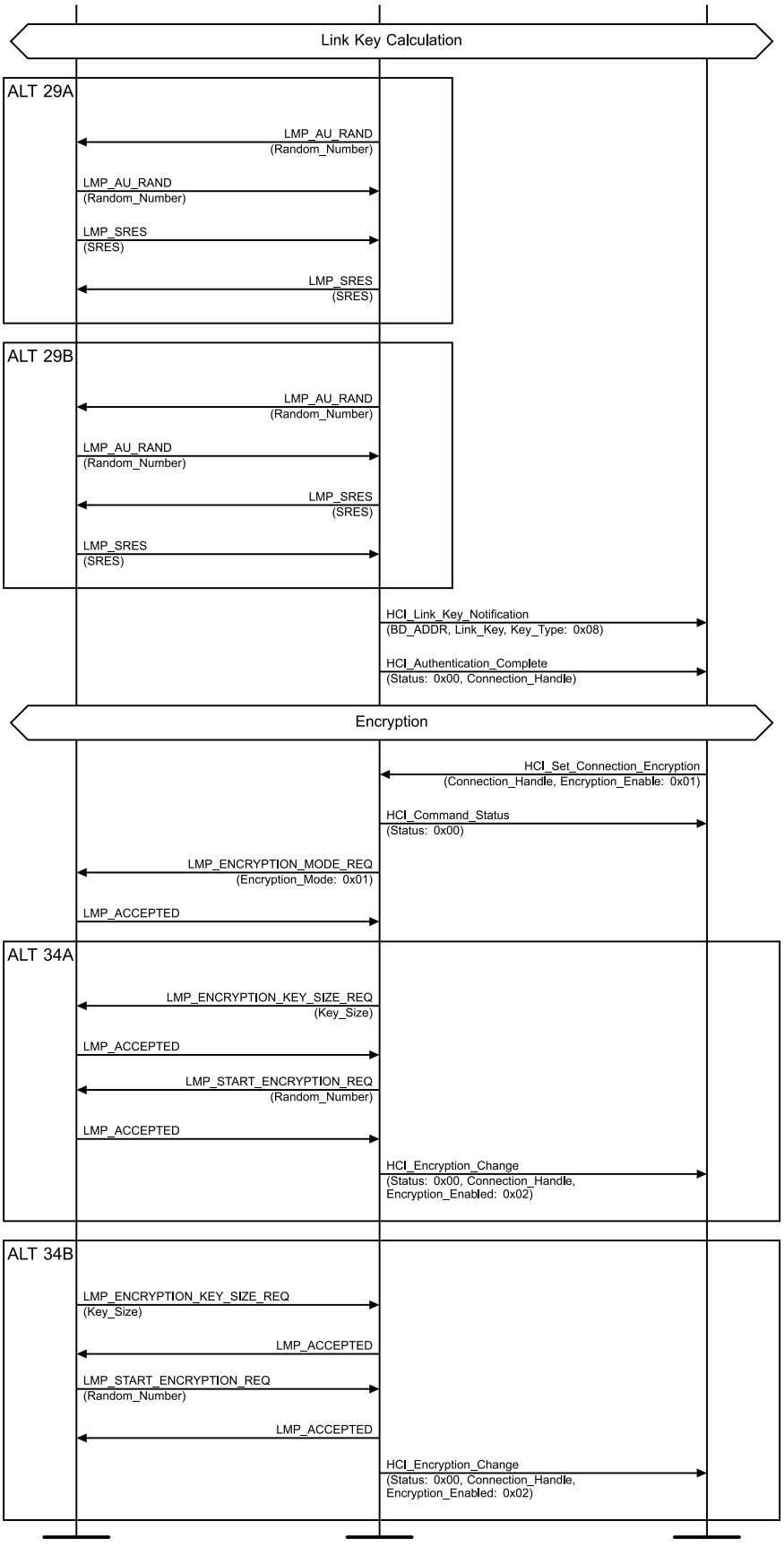


Figure 4.11-9: LMP/SP/BV-39-C [Secure Connections Capable Controller – Host has no P256 OOB data available – IUT Initiator – OOB] MSC – Page 3 of 3

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x01 (P-192 OOB authentication data from remote device present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 four times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 four times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
18. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.

21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
23. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
24. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
25. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
26. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
27. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
28. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
29. Perform either alternative 29A or 29B depending on the IUT's role.

    Alternative 29A (The IUT is the Central):

    29A.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.

    29A.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number and an LMP_SRES PDU with the SRES.

    29A.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.

    Alternative 29B (The IUT is the Peripheral):

    29B.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.

    29B.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.

    29B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.

    29B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES.
30. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x08 (Authenticated Combination Key generated from P-256) and a successful HCI_Authentication_Complete event with the Connection_Handle.
31. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response.
32. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
33. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
34. Perform either alternative 34A or 34B depending on the IUT's role.

    Alternative 34A (The IUT is the Central):

    34A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.

    34A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    34A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

    34A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    34A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

Alternative 34B (The IUT is the Peripheral):

34B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

34B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

34B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

34B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

34B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

- Expected Outcome

  Pass verdict

  The IUT does not send an HCI_Remote_OOB_Data_Request Event to the Upper Tester.

  In Step 22, the IUT sends the HCI_User_Confirmation_Request event to the Upper Tester with the same value calculated by the Lower Tester.

  In Step 24, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 28, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 30, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x08 (Authenticated Combination Key generated from P-256).

LMP/SP/BV-40-C [Secure Connections Capable Controller and Host – Legacy Remote Controller – IUT Initiator]

- Test Purpose

  Verify that the IUT initiates Secure Simple Pairing using the P-192 Elliptic Curve when the remote controller does not have support for the Secure Connections (Controller Support) LMP feature bit.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section.
  - The IUT is the Initiator, and it does not matter if the IUT is the Central or the Peripheral.
  - The Lower Tester does not set the Secure Connections (Controller Support) to enabled.
  - The IUT has the Secure Connections (Controller Support) and Secure Connections (Host Support) LMP feature bits set

- Test Procedure

  1. Execute LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success].

- Expected Outcome

  Pass verdict

  The Pass verdict of LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success] applies.

4.11.3 Numeric Comparison procedures

Note: In all the test cases in Section 4.11.3 Numeric Comparison procedures, it does not matter whether the IUT is the Central or the Peripheral.

LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success]

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol. Verify that the IUT generates a different Simple Pairing Number each time Authentication Stage 1 executes.

- Reference

  [1] 4.2.7

  [13] 7.2.1

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

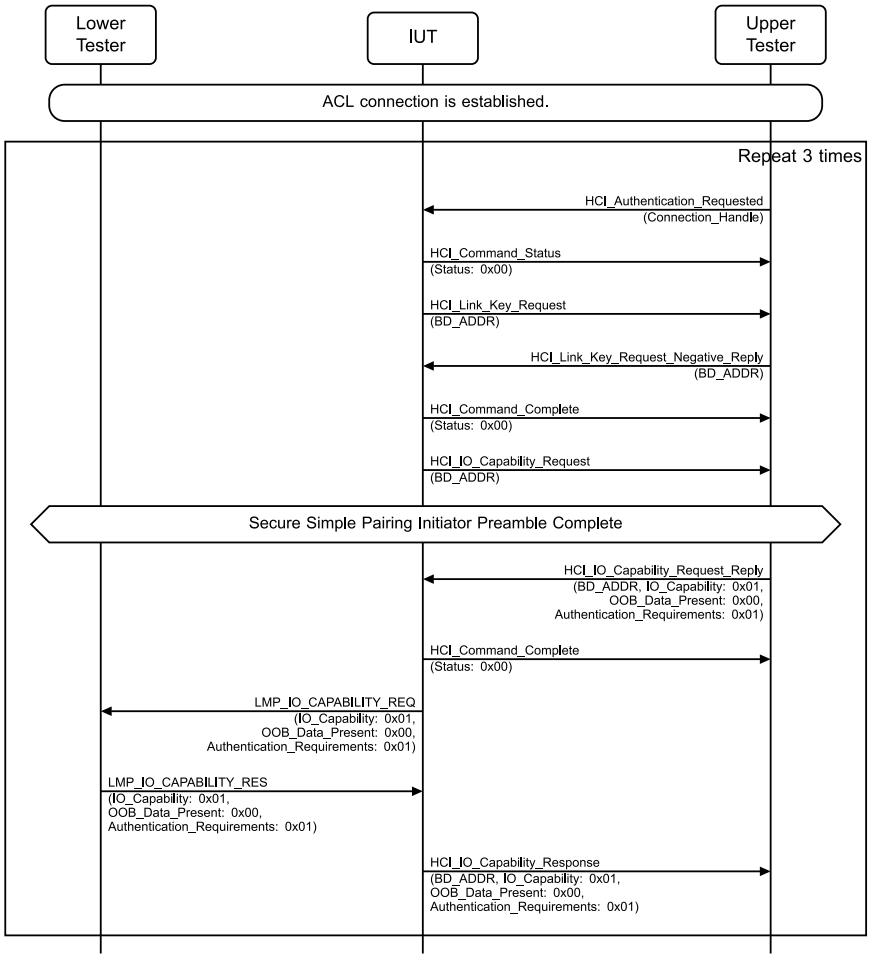


Figure 4.11-10: LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success] MSC – Page 1 of 3

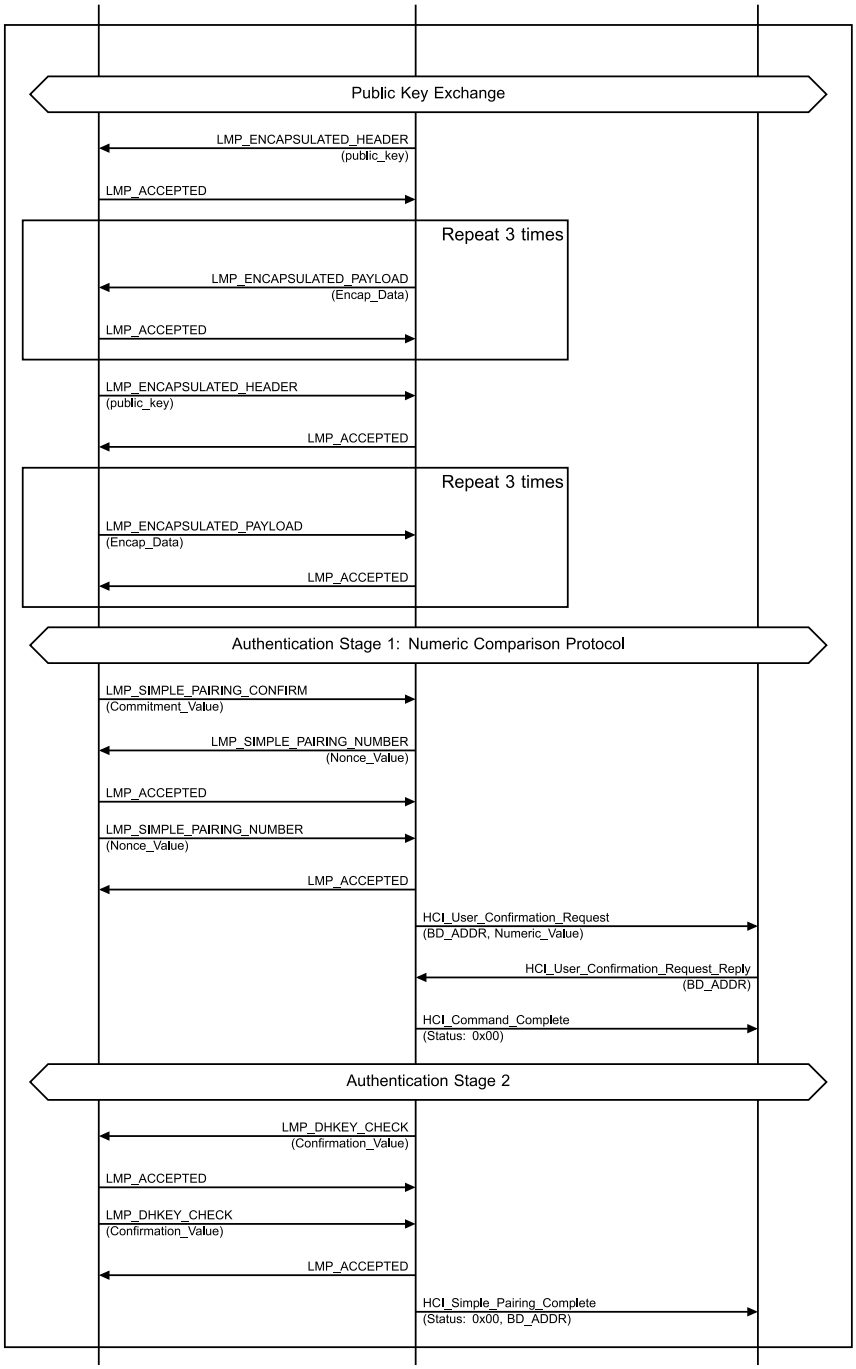


Figure 4.11-11: LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success] MSC – Page 2 of 3

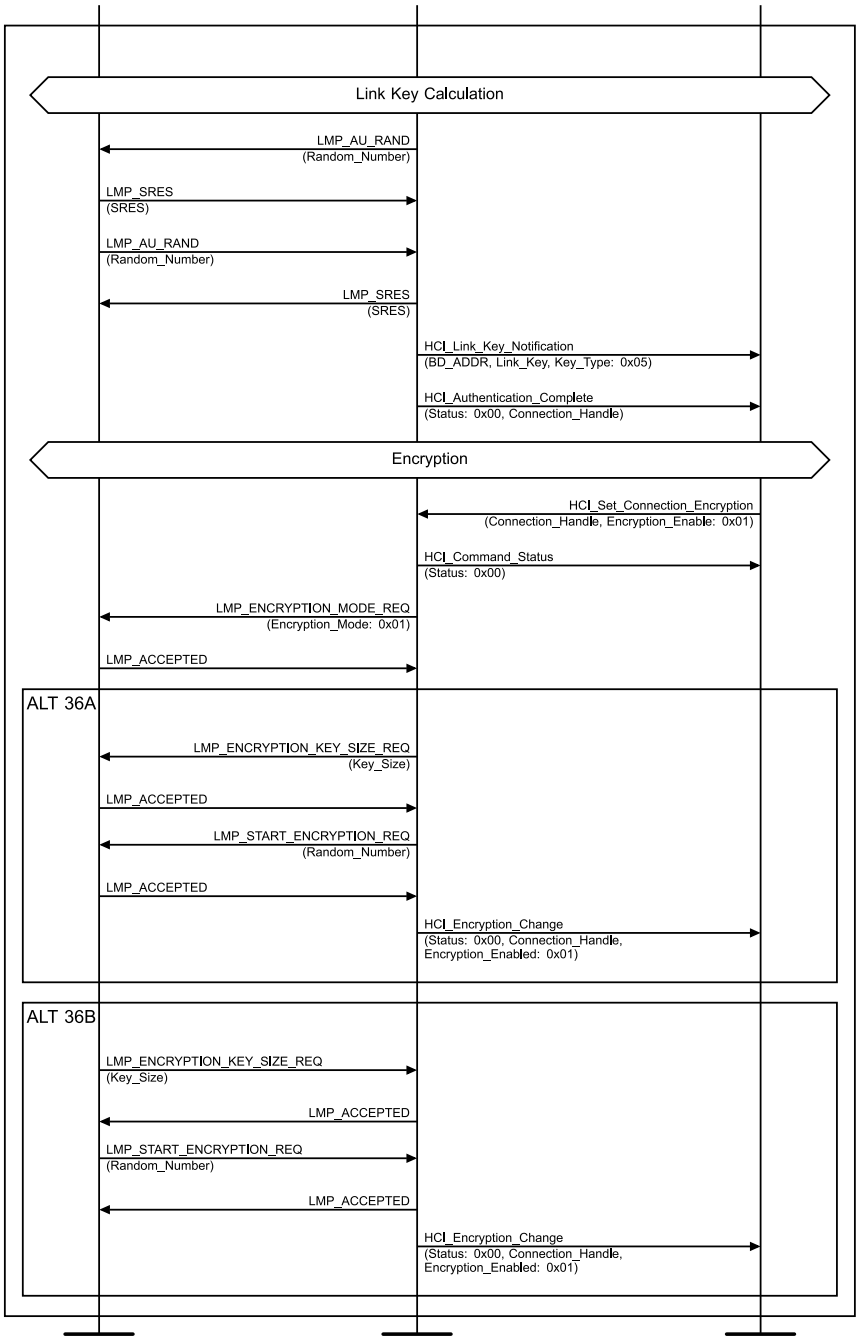


Figure 4.11-12: LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success] MSC – Page 3 of 3

Execute Steps 1–36 three times, with the Lower Tester storing the Nonce_Value sent by the IUT in Step 18 each time.

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.

5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
18. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
23. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
24. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.

25. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
26. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
27. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
28. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
29. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
30. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
31. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
32. The IUT sends an HCI_Link_Key_Notification event to the IUT with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192) and a successful HCI_Authentication_Complete event with the Connection_Handle.
33. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response.
34. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
35. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
36. Perform either alternative 36A or 36B depending on the IUT's role.

    Alternative 36A (The IUT is the Central):

    36A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.

    36A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    36A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

    36A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    36A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

    Alternative 36B (The IUT is the Peripheral):

    36B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

    36B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    36B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

    36B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    36B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

- Expected Outcome

  Pass verdict

  In Step 22, the IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the same value calculated by the Lower Tester.

  The IUT always generates a unique public key.

In Step 24, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

In Step 28, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

In Step 32, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

All three Nonce_Values generated by the IUT in Step 18 are different values.

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-07-C [Numeric Comparison - IUT Responder – Success]

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol and that the IUT generates a different Simple Pairing Number each time Authentication Stage 1 executes.

- Reference

  [1] 4.2.7

  [13] 7.2.1

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Procedure

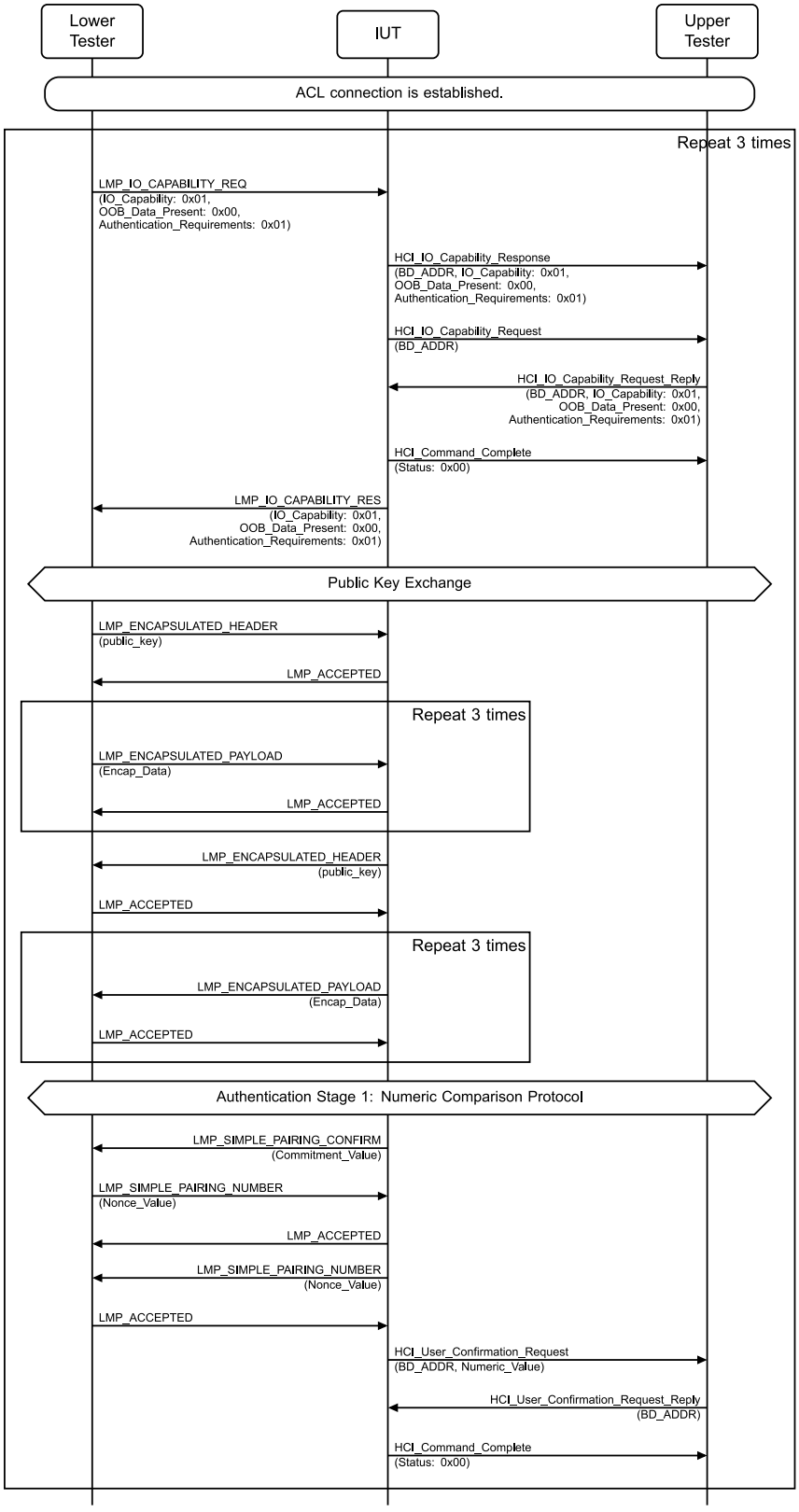


Figure 4.11-13: LMP/SP/BV-07-C [Numeric Comparison - IUT Responder – Success] MSC – Page 1 of 2

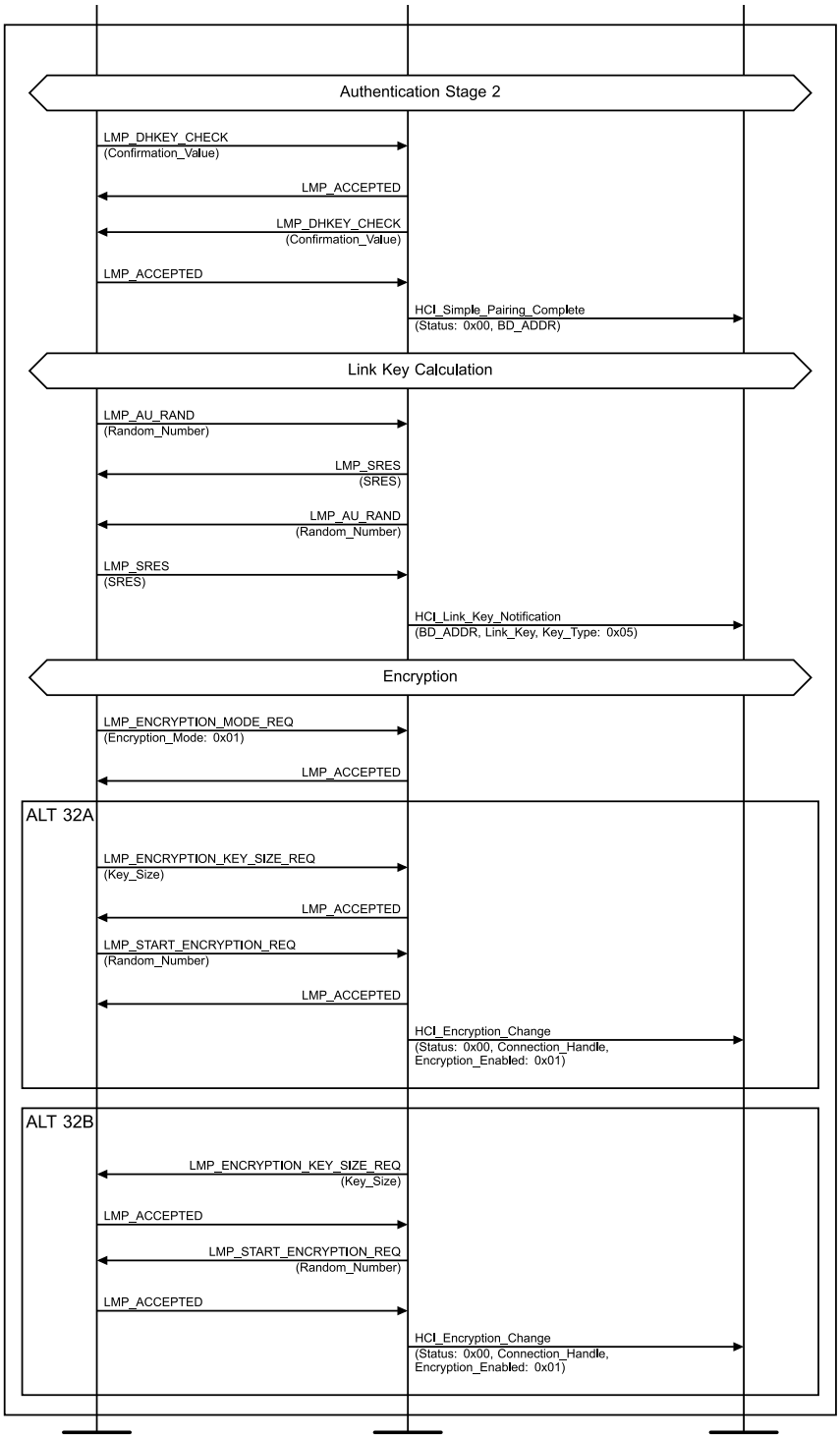


Figure 4.11-14: LMP/SP/BV-07-C [Numeric Comparison - IUT Responder – Success] MSC – Page 2 of 2

Execute Steps 1–32 three times, with the Lower Tester storing the Nonce_Value sent by the IUT in Step 17 each time.

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication

data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
15. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
17. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
18. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
19. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
20. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
21. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
22. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
23. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
24. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.

25. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
26. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
27. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
28. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.
29. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192).
30. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption).
31. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
32. Perform either alternative 32A or 32B depending on the IUT's role.

    Alternative 32A (The IUT is the Peripheral):

    32A.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

    32A.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    32A.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

    32A.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    32A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

    Alternative 32B (The IUT is the Central):

    32B.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.

    32B.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    32B.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

    32B.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    32B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

- Expected Outcome

  Pass verdict

  In Step 20, the IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the same value calculated by the Lower Tester.

  The IUT always generates a unique public key.

  In Step 23, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 25, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 29, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

  All three Nonce_Values generated by the IUT in Step 17 are different values.

- Notes

  If the Commitment_Value calculated by the Lower Tester does not match the Commitment_Value sent by the IUT, the Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT with the Authentication_Failure Error_Code.

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

4.11.3.1 Numeric Comparison – IUT Initiator – Failure on Initiating Side

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol and that the IUT responds correctly when the Upper Tester responds that the numeric comparison value did not verify correctly.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-2.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-08-C [Numeric Comparison - IUT Initiator - Failure on Initiating Side]	P-192
LMP/SP/BV-43-C [Numeric Comparison – IUT Initiator, Failure on Initiating side, P-256]	P-256

Table 4.11-2: Numeric Comparison – IUT Initiator – Failure on Initiating Side test cases

- Test Procedure

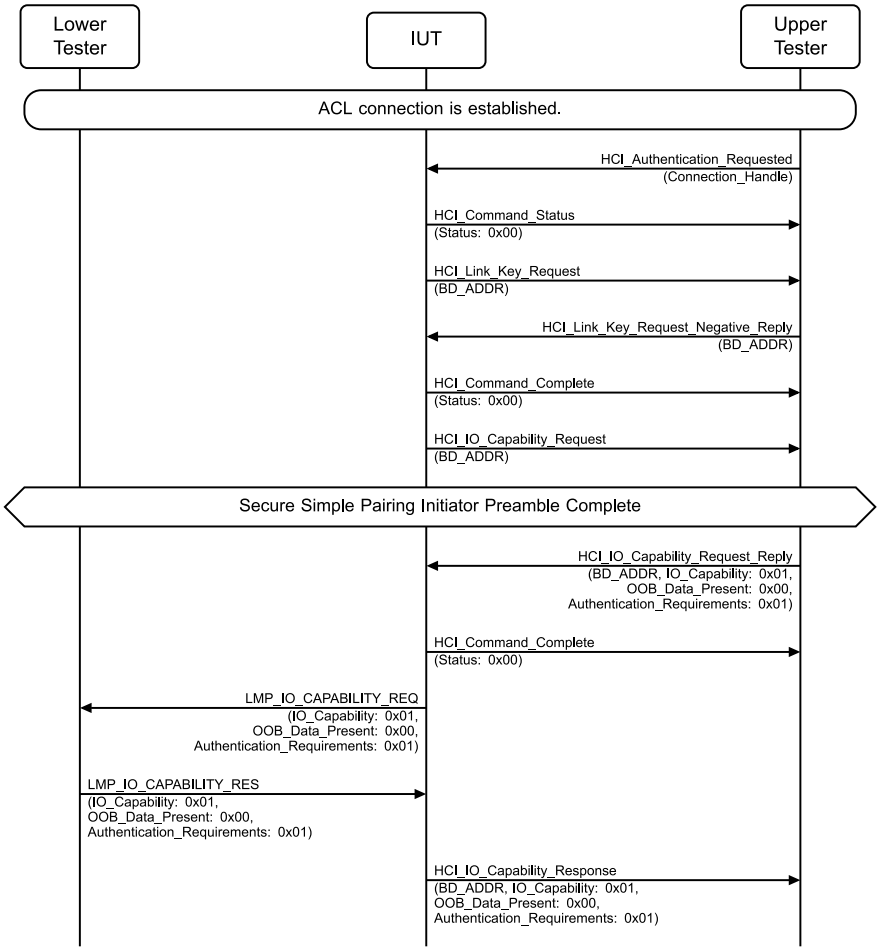


Figure 4.11-15: Numeric Comparison – IUT Initiator – Failure on Initiating Side MSC – Page 1 of 2

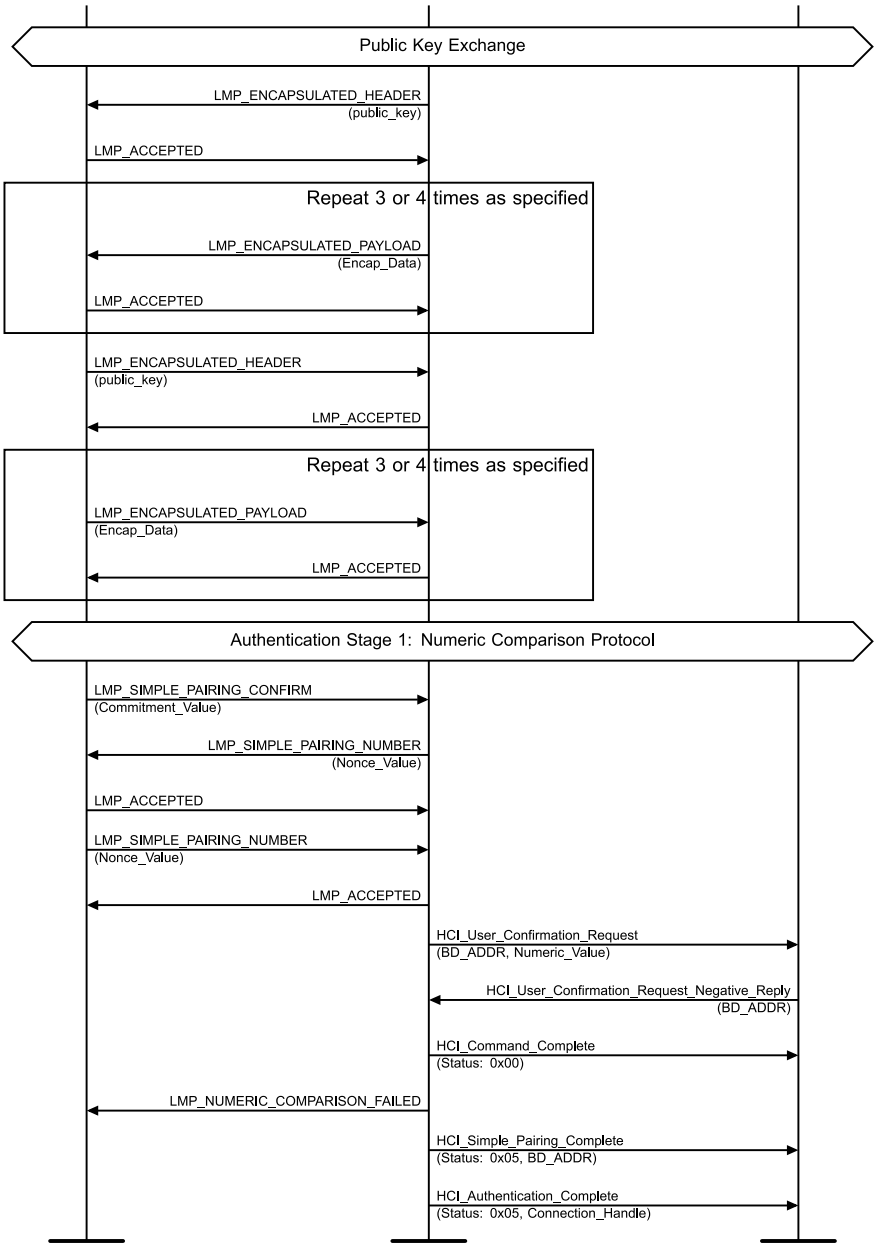


Figure 4.11-16: Numeric Comparison – IUT Initiator – Failure on Initiating Side MSC – Page 2 of 2

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.

6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-2.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-2.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
18. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
23. The Upper Tester sends an HCI_User_Confirmation_Request_Negative_Reply command to the IUT with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
24. The IUT sends an LMP_NUMERIC_COMPARISON_FAILED PDU to the Lower Tester.
25. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 24, the IUT sends the LMP_NUMERIC_COMPARISON_FAILED PDU to the Lower Tester.

  In Step 25, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

4.11.3.2 Numeric Comparison – IUT Responder – Failure on Initiating Side

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol and that the IUT responds correctly when the Upper Tester responds that the numeric comparison value did not verify correctly.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-3.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-09-C [Numeric Comparison - IUT Responder - Failure on Initiating Side]	P-192
LMP/SP/BV-44-C [Numeric Comparison – IUT Responder – Failure on Initiating Side, P-256]	P-256

Table 4.11-3: Numeric Comparison – IUT Responder – Failure on Initiating Side test cases

- Test Procedure

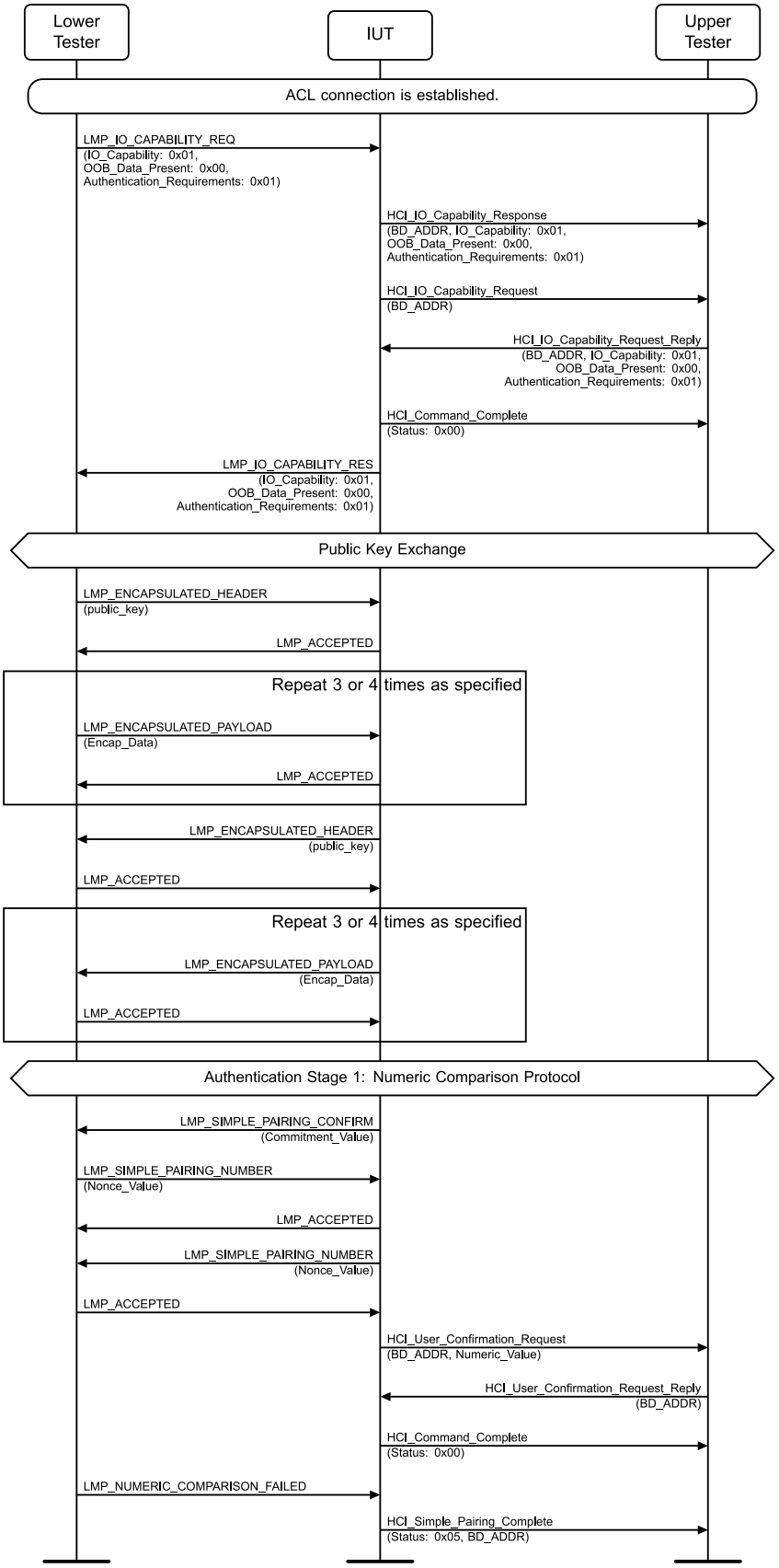


Figure 4.11-17: Numeric Comparison – IUT Responder – Failure on Initiating Side MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times if using P-192 or four times if using P-256 as specified in Table 4.11-3.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times if using P-192 or four times if using P-256 as specified in Table 4.11-3.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
15. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
17. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
18. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
19. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
20. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.

21. The Lower Tester sends an LMP_NUMERIC_COMPARISON_FAILED PDU to the IUT.
22. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after receiving the LMP_NUMERIC_COMPARISON_FAILED PDU from the Lower Tester.

4.11.3.3 Numeric Comparison – IUT Initiator – Failure on Responding Side

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol and that the IUT responds correctly when the responding side fails the numeric comparison check step.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-4.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-10-C [Numeric Comparison - IUT Initiator - Failure on Responding Side]	P-192
LMP/SP/BV-45-C [Numeric Comparison – IUT Initiator – Failure on Responding Side, P-256]	P-256

Table 4.11-4: Numeric Comparison – IUT Initiator – Failure on Responding Side test cases

- Test Procedure

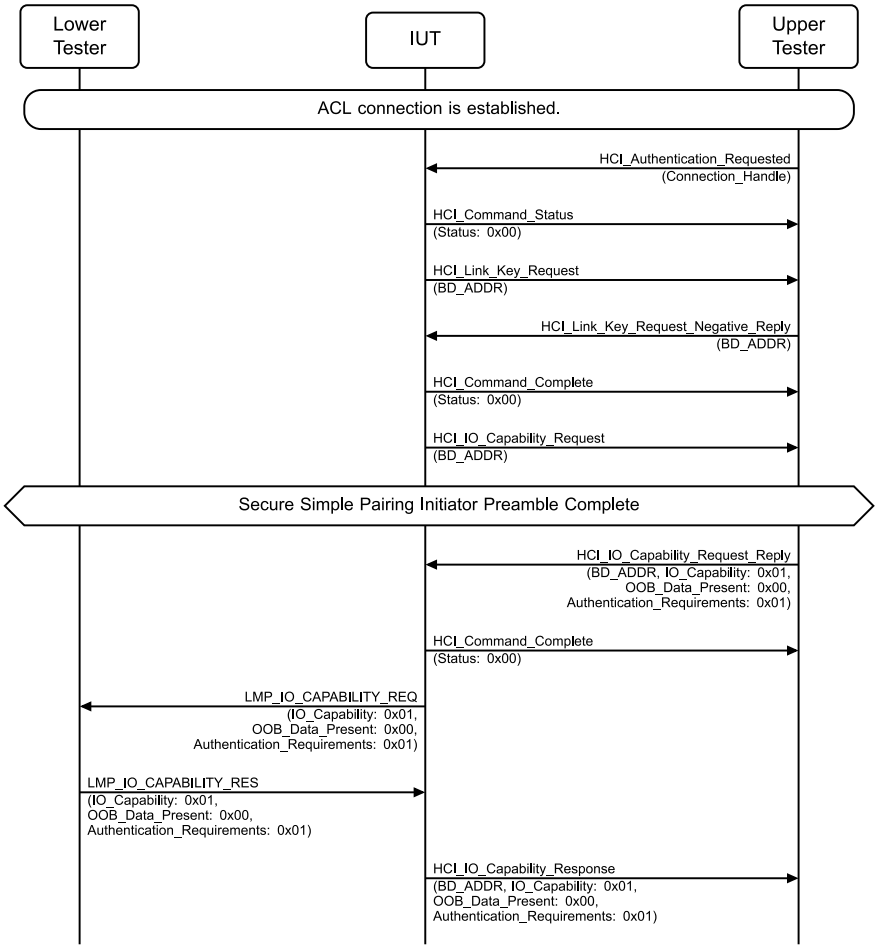


Figure 4.11-18: Numeric Comparison – IUT Initiator – Failure on Responding Side MSC – Page 1 of 2

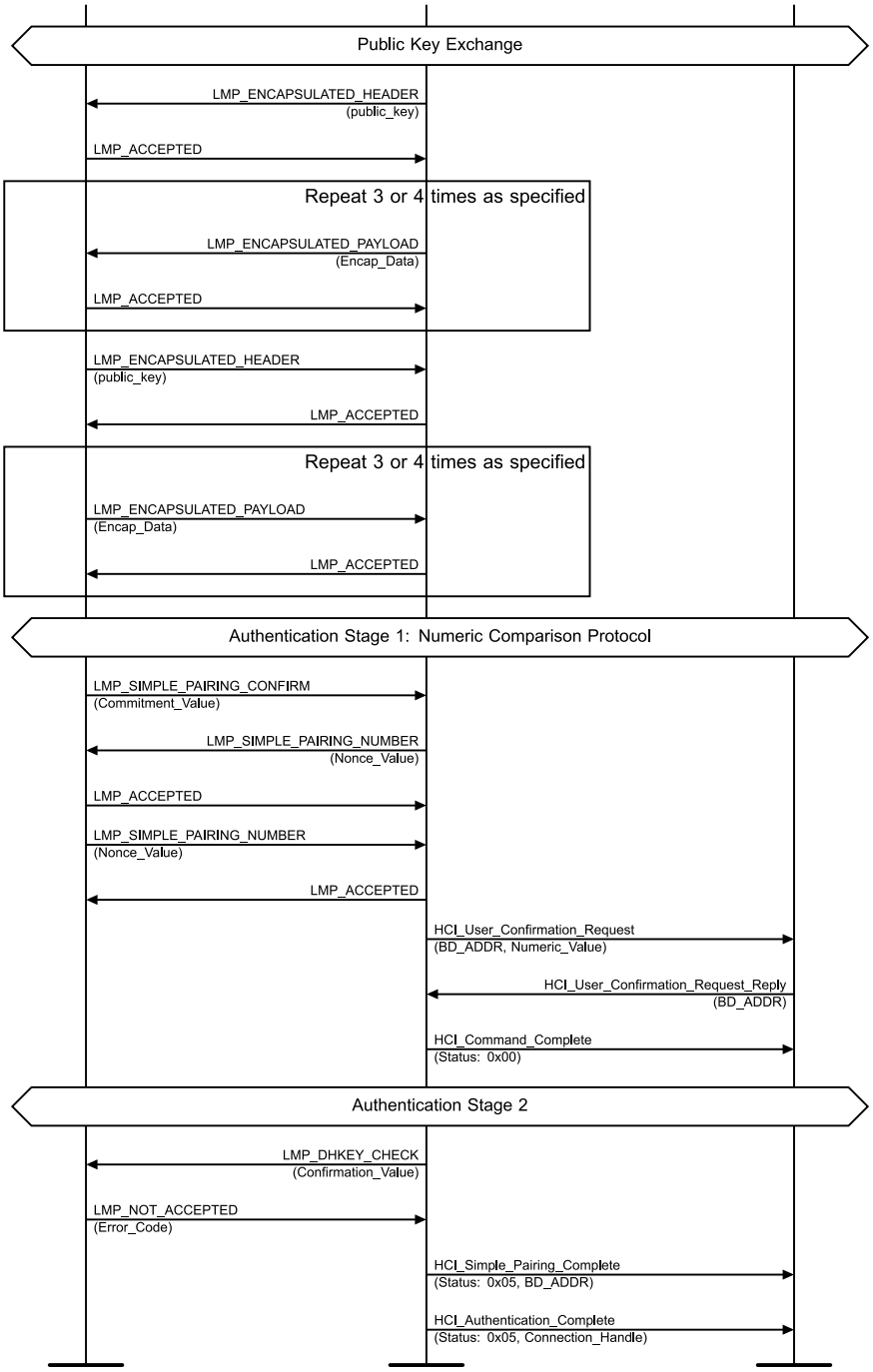


Figure 4.11-19: Numeric Comparison – IUT Initiator – Failure on Responding Side MSC – Page 2 of 2

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB

authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.

6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-4.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-4.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
18. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
23. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
24. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.

25. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and an Error_Code.
26. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after the Lower Tester responds to the LMP_DHKEY_CHECK PDU from the IUT with an LMP_NOT_ACCEPTED PDU.

4.11.3.4 Numeric Comparison – IUT Responder – Failure on Responding Side

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol and that the IUT responds correctly when the responding side fails the numeric comparison check step.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-5.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-11-C [Numeric Comparison - IUT Responder - Failure on Responding Side]	P-192
LMP/SP/BV-46-C [Numeric Comparison – IUT Responder – Failure on Responding Side, P-256]	P-256

Table 4.11-5: Numeric Comparison – IUT Responder – Failure on Responding Side test cases

- Test Procedure

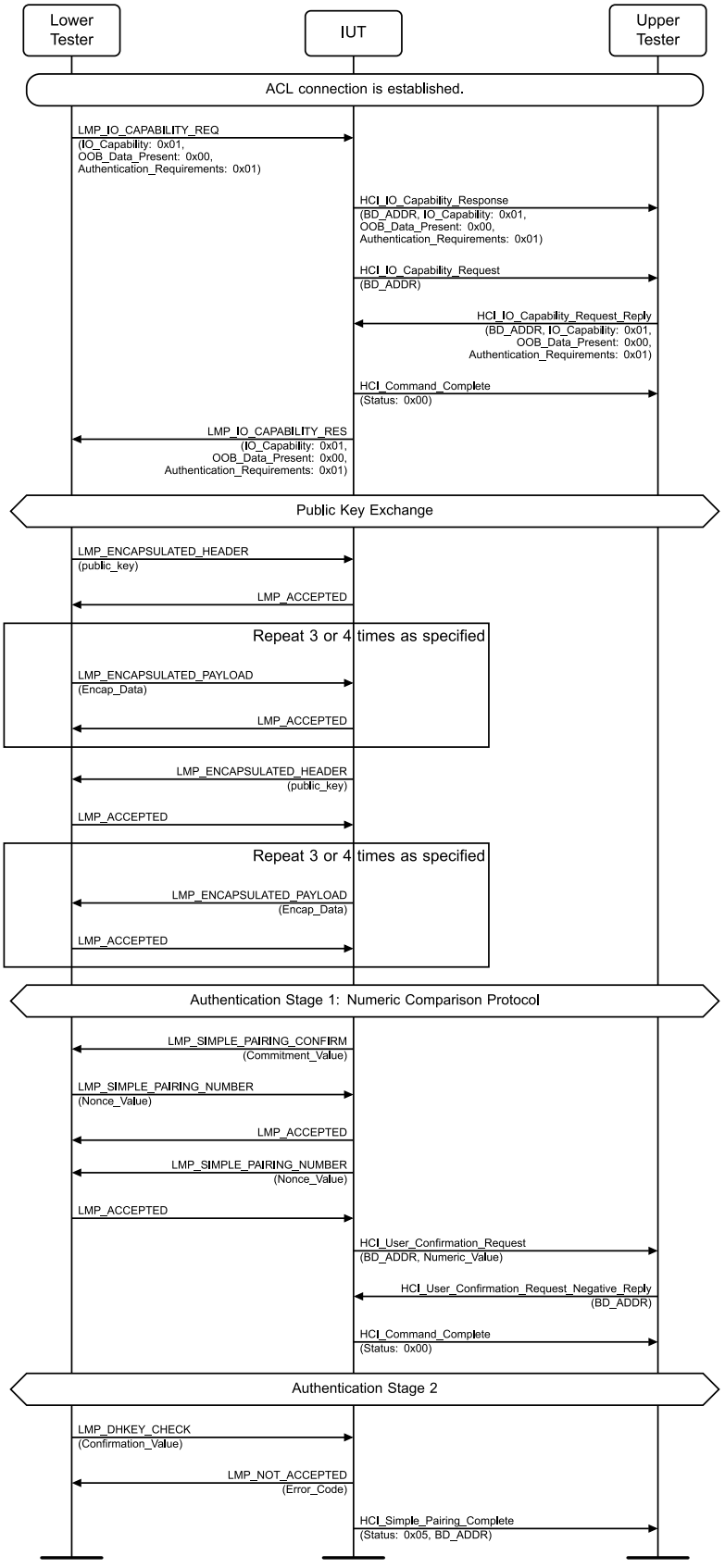


Figure 4.11-20: Numeric Comparison – IUT Responder – Failure on Responding Side MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times if using P-192 or four times if using P-256 as specified in Table 4.11-5.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times if using P-192 or four times if using P-256 as specified in Table 4.11-5.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
15. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
17. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
18. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
19. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
20. The Upper Tester sends an HCI_User_Confirmation_Request_Negative_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
21. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.

22. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and an Error_Code.
23. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after the IUT responds to the LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU.

LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256]

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol using the P-256 elliptic curve.

- Reference

  [1] 4.2.7

- Initial Condition
  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

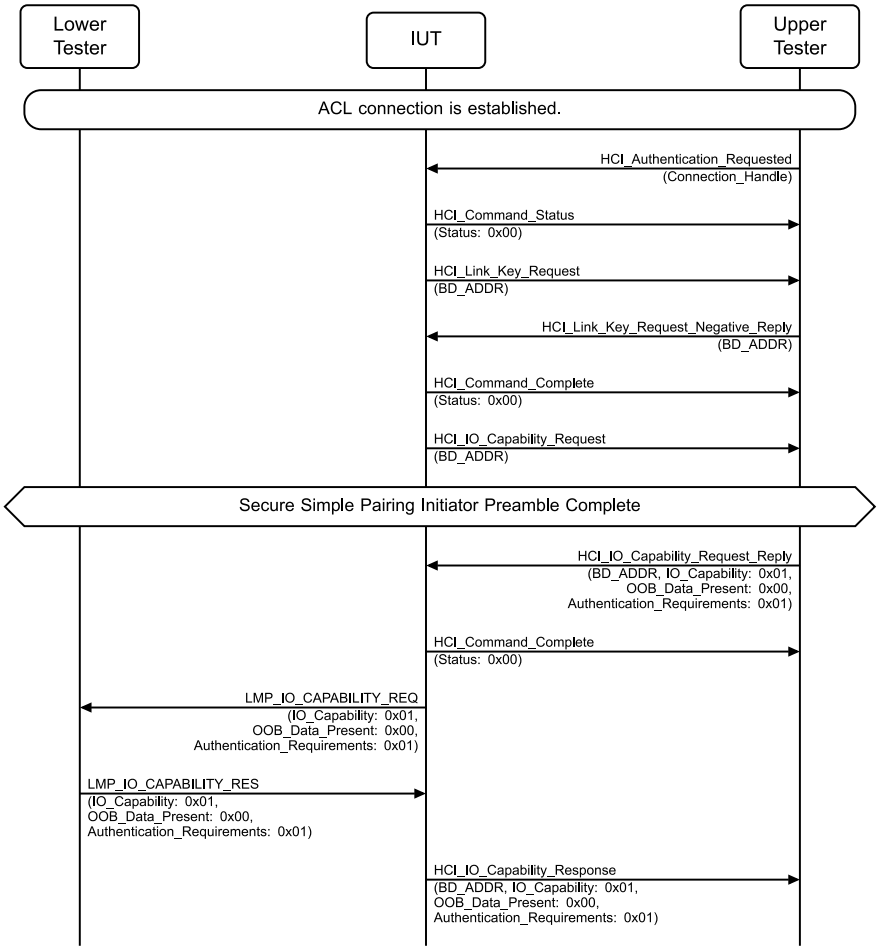


Figure 4.11-21: LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256] MSC – Page 1 of 3

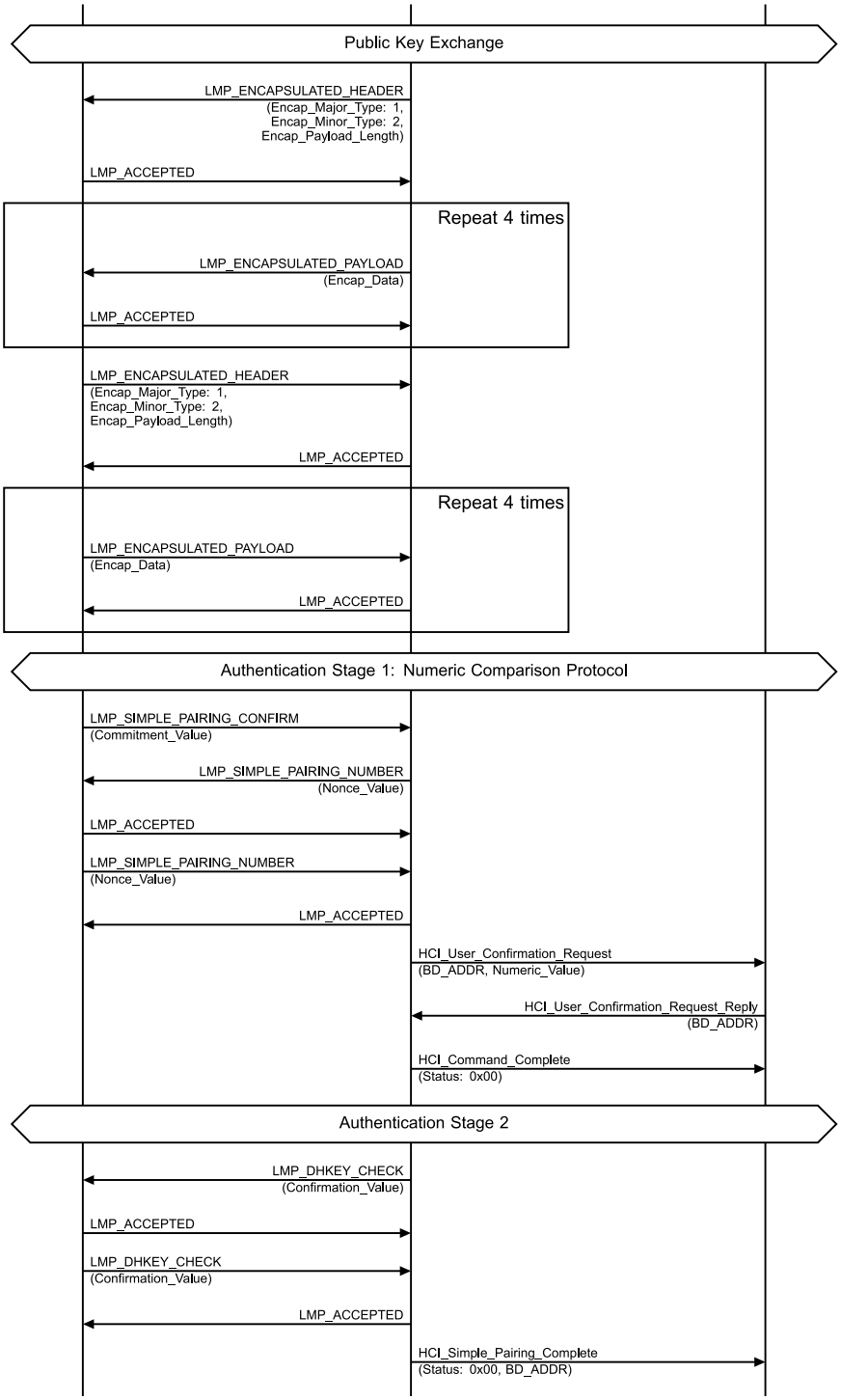


Figure 4.11-22: LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256] MSC – Page 2 of 3

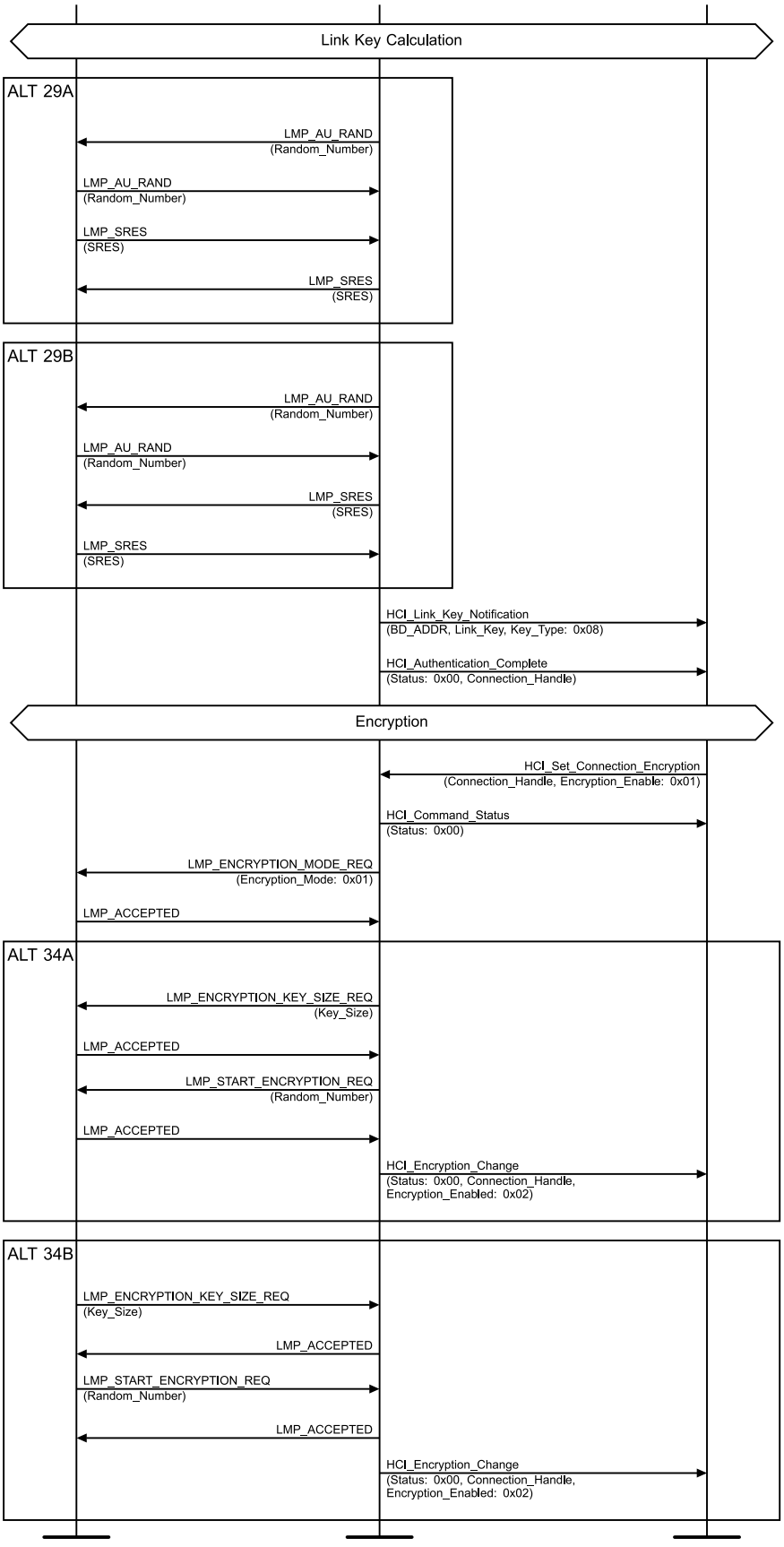


Figure 4.11-23: LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256] MSC – Page 3 of 3

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 four times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 four times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
18. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.

21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
23. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
24. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
25. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
26. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
27. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
28. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
29. Perform either alternative 29A or 29B depending on the IUT's role.
    Alternative 29A (The IUT is the Central):
    - 29A.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
    - 29A.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
    - 29A.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.
    - 29A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES.

    Alternative 29B (The IUT is the Peripheral):
    - 29B.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
    - 29B.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
    - 29B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
    - 29B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES.
30. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x08 (Authenticated Combination Key generated from P-256) and a successful HCI_Authentication_Complete event with the Connection_Handle.
31. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response.
32. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
33. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
34. Perform either alternative 34A or 34B depending on the IUT's role.
    Alternative 34A (The IUT is the Central):
    - 34A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
    - 34A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    - 34A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
    - 34A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    - 34A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

Alternative 34B (The IUT is the Peripheral):

34B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

34B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

34B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

34B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

34B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

- Expected Outcome

  Pass verdict

  In Step 22, the IUT sends the HCI_User_Confirmation_Request event to the Upper Tester with the same value calculated by the Lower Tester.

  In Step 24, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 28, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 30, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x08 (Authenticated Combination Key generated from P-256).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to an LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-42-C [Numeric Comparison – IUT Responder – Success, P-256]

- Test Purpose

  Verify that the IUT supports the Numeric Comparison protocol using the P-256 elliptic curve.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Procedure

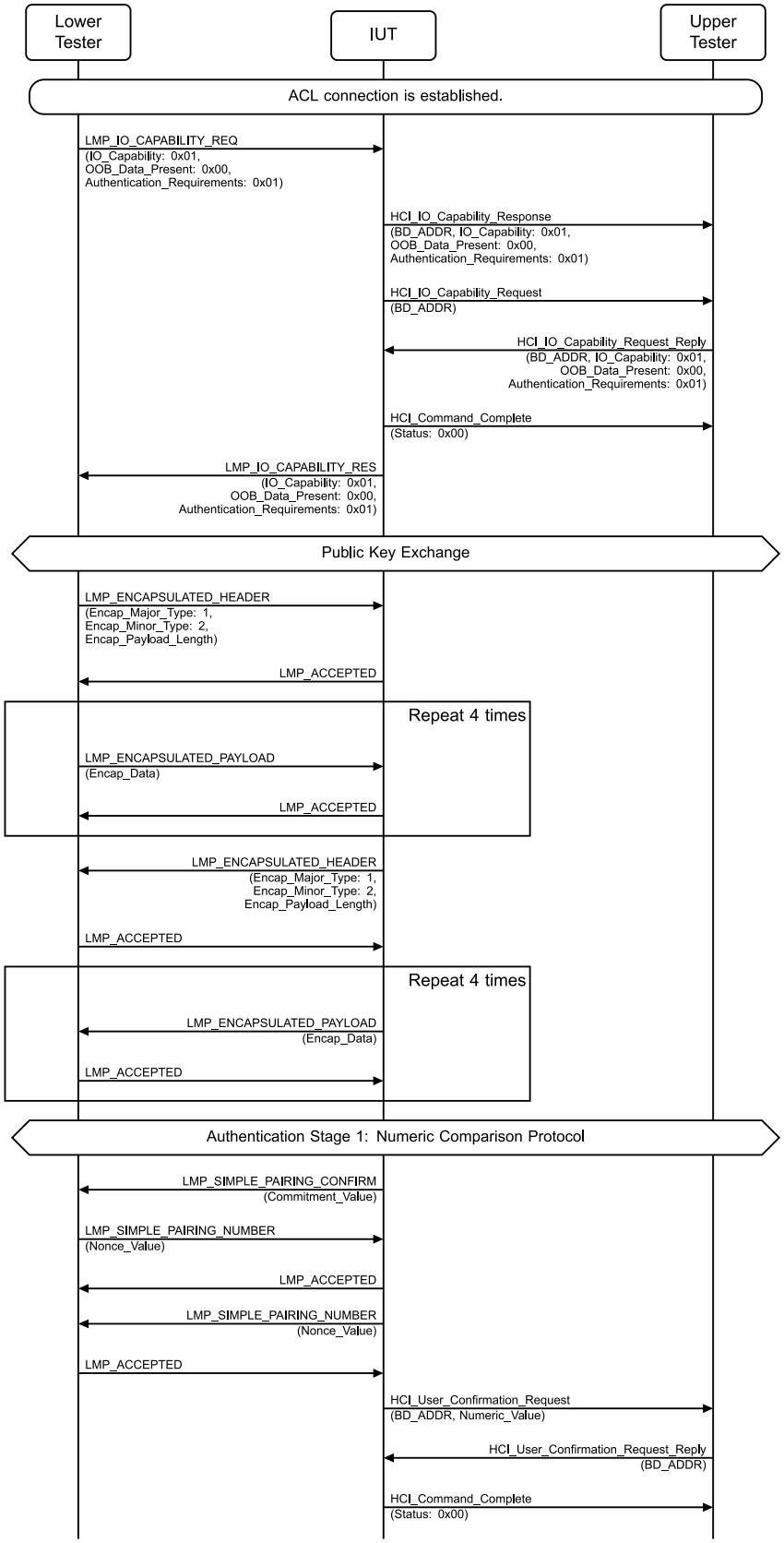


Figure 4.11-24: LMP/SP/BV-42-C [Numeric Comparison – IUT Responder – Success, P-256] MSC – Page 1 of 2

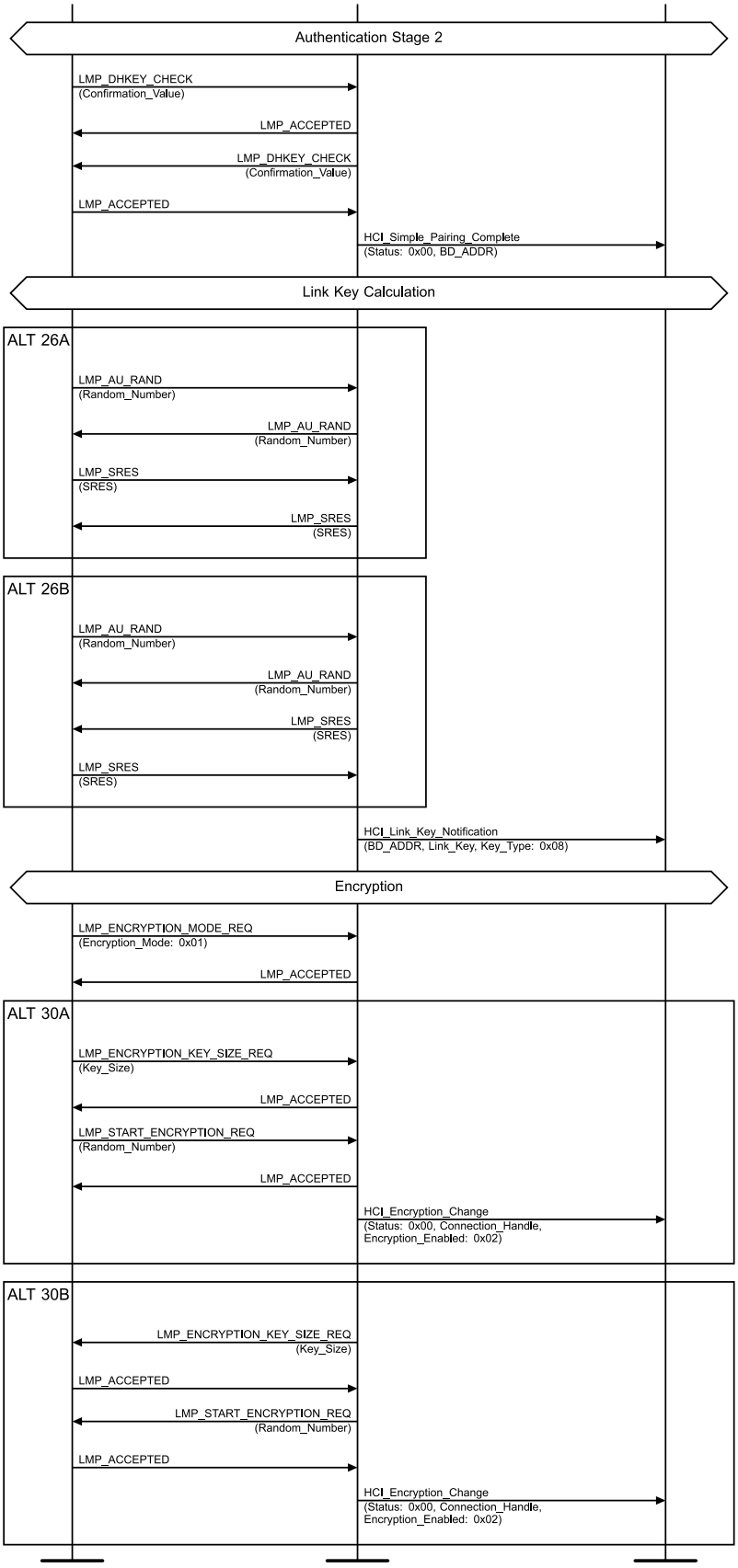


Figure 4.11-25: LMP/SP/BV-42-C [Numeric Comparison – IUT Responder – Success, P-256] MSC – Page 2 of 2

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 four times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 four times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
15. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
17. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
18. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
19. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
20. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
21. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.

22. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
23. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
24. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
25. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
26. Perform either alternative 26A or 26B depending on the IUT's role.

    Alternative 26A (The IUT is the Central):

    26A.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
    26A.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.
    26A.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.
    26A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES.

    Alternative 26B (The IUT is the Peripheral):

    26B.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
    26B.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.
    26B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
    26B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES.
27. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x08 (Authenticated Combination Key generated from P-256).
28. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption).
29. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
30. Perform either alternative 30A or 30B depending on the IUT's role.

    Alternative 30A (The IUT is the Peripheral):

    30A.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.
    30A.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    30A.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
    30A.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    30A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

    Alternative 30B (The IUT is the Central):

    30B.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
    30B.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    30B.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
    30B.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    30B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

- Expected Outcome

  Pass verdict

  In Step 19, the IUT sends the HCI_User_Confirmation_Request event to the Upper Tester with the same value calculated by the Lower Tester.

  In Step 23, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 25, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 27, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x08 (Authenticated Combination Key generated from P-256).

- Notes

  If the Commitment_Value calculated by the Lower Tester does not match the Commitment_Value sent by the IUT, then the Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT with the Authentication_Failure Error_Code.

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-47-C [Pairing on encrypted ACL – Numeric Comparison – IUT Initiator – Success, P-256]

- Test Purpose

  Verify that the IUT performs encryption pause and resume at the end of pairing using the Numeric Comparison protocol using the P-256 elliptic curve if the ACL connection was already encrypted.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Initiator.
  - An ACL connection has been established.
  - The IUT has successfully executed LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256].

- Test Procedure

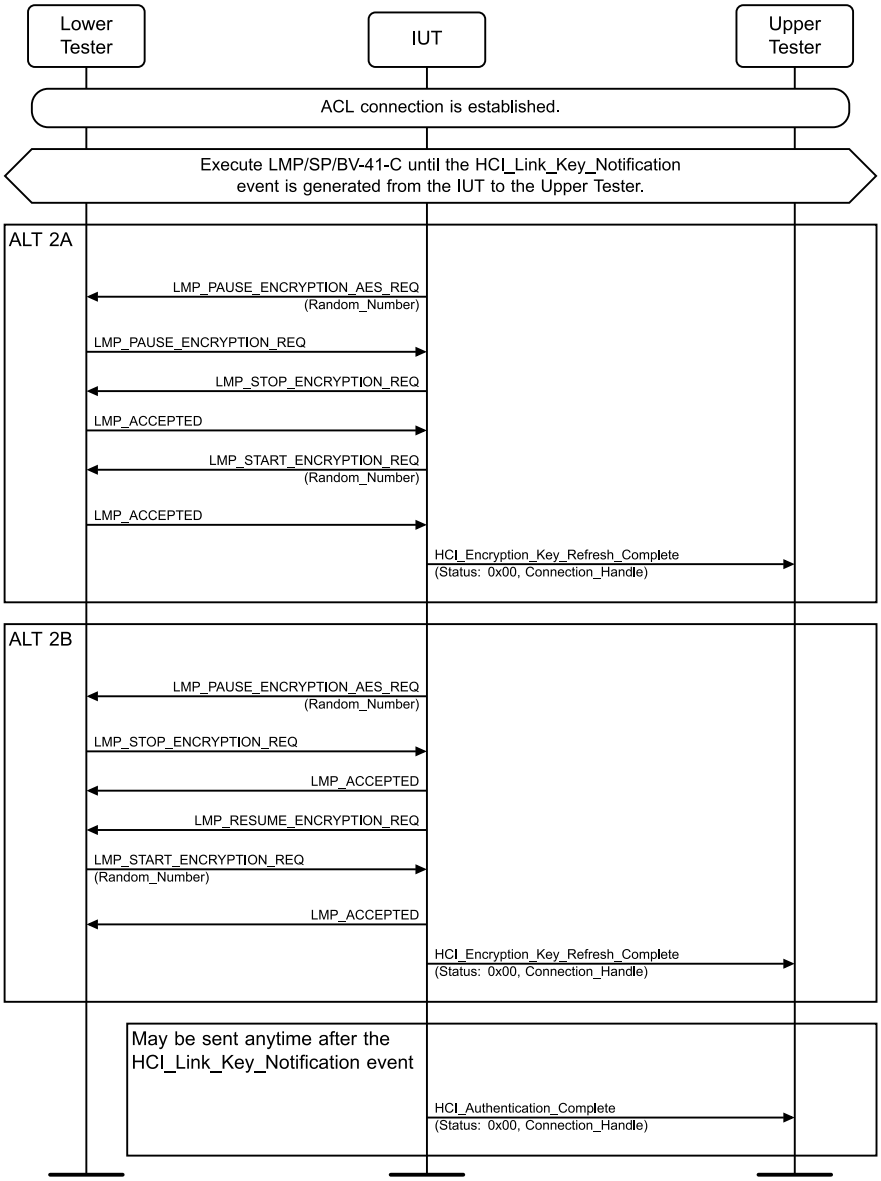


Figure 4.11-26: LMP/SP/BV-47-C [Pairing on encrypted ACL – Numeric Comparison – IUT Initiator – Success, P-256] MSC

1. Execute LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256] again up through Step 30 such that the HCI_Link_Key_Notification event is generated from the IUT to the Upper Tester.
2. Perform either alternative 2A or 2B depending on the IUT's role.
   Alternative 2A (The IUT is the Central):
   - 2A.1. The IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number.
   - 2A.2. The Lower Tester responds to the IUT with an LMP_PAUSE_ENCRYPTION_REQ PDU.
   - 2A.3. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
   - 2A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.

2A.5. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
2A.6. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
2A.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.

Alternative 2B (The IUT is the Peripheral):

2B.1. The IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester with a Random_Number.
2B.2. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
2B.3. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
2B.4. The IUT sends an LMP_RESUME_ENCRYPTION_REQ PDU to the Lower Tester.
2B.5. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
2B.6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
2B.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.

Step 3 can be sent at any time after Step 1.

3. The IUT sends a successful HCI_Authentication_Complete event to the Upper Tester with the Connection_Handle.

- Expected Outcome

  Pass verdict

  Both of the pairing procedures are successful.

  The IUT initiates an encryption pause resume at the end of the second pairing procedure and sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester.

4.11.4 Passkey Entry procedures

Note: In all the test cases in Section 4.11.4 Passkey Entry procedures, it does not matter whether the IUT is the Central or the Peripheral.

LMP/SP/BV-12-C [Passkey Entry - IUT Initiator – Success]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

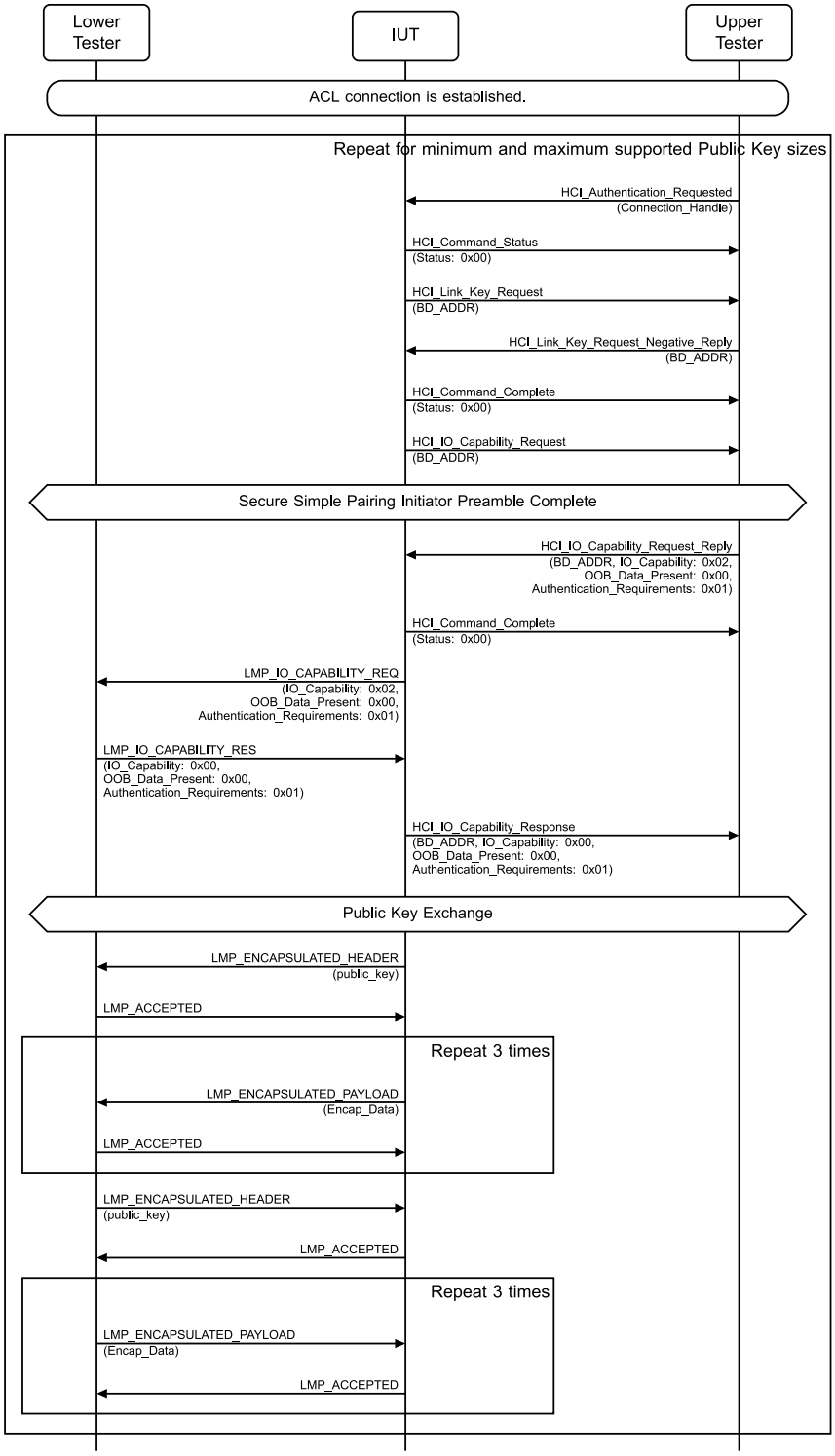


Figure 4.11-27: LMP/SP/BV-12-C [Passkey Entry - IUT Initiator – Success] MSC – Page 1 of 2

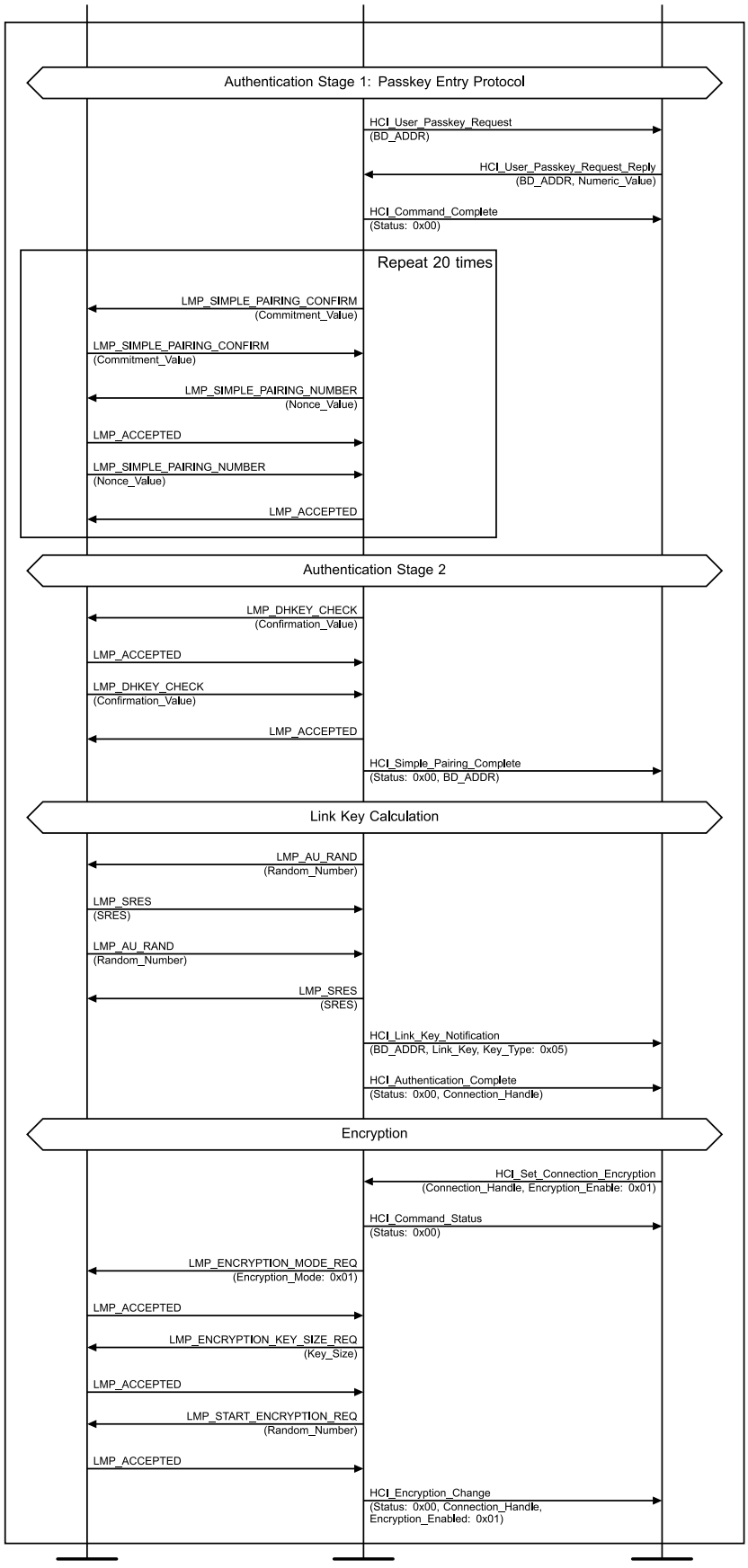


Figure 4.11-28: LMP/SP/BV-12-C [Passkey Entry - IUT Initiator – Success] MSC – Page 2 of 2

Repeat Steps 1–41 for the minimum and maximum Public Key size.

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds to the IUT with an HCI_User_Passkey_Request_Reply command with the BD_ADDR and Numeric_Value and receives a successful HCI_Command_Complete event in response.

Repeat Steps 19–24 20 times.

19. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.

20. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
21. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
22. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
23. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
24. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
25. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
26. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
27. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
28. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
29. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
30. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
31. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
32. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
33. The IUT sends an HCI_Link_Key_Notification event to the IUT with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192) and a successful HCI_Authentication_Complete event with the Connection_Handle.
34. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response.
35. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
36. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
37. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
38. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
39. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
40. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
41. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

- Expected Outcome

  Pass verdict

  The IUT always generates a unique public key.

  In Step 25, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 29, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester.

In Step 33, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

In Step 18, if the IUT allows a passkey to be entered that is shorter than 6 digits (20 bits), then the missing most significant bits are set to 0. For example, if the passkey is “4093”, then the confirm values are generated using the appropriate bits from the r value 0x00FFD.

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-13-C [Passkey Entry - IUT Responder – Success]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol.

- Reference

  [1] 4.2.7

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Procedure

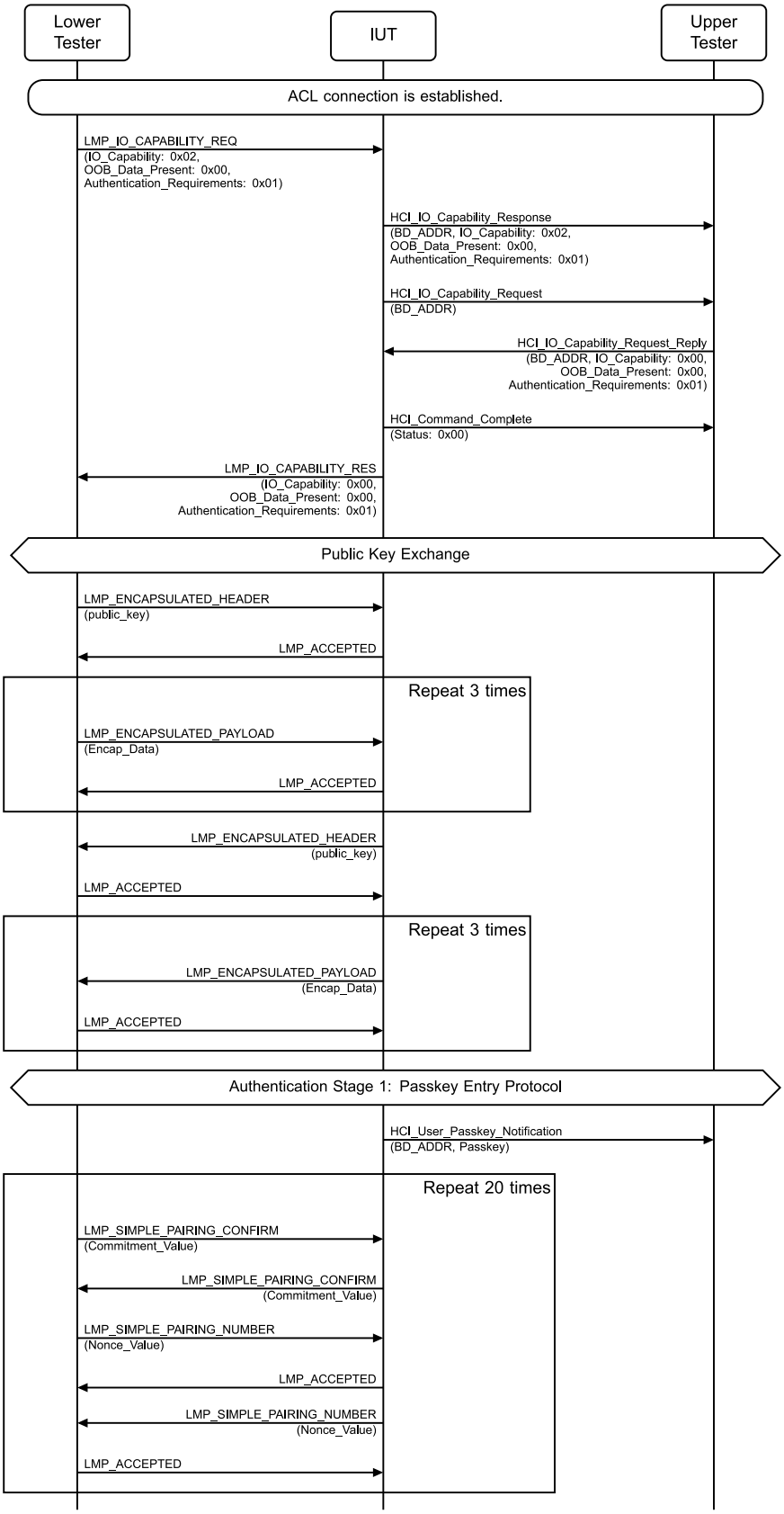


Figure 4.11-29: LMP/SP/BV-13-C [Passkey Entry - IUT Responder – Success] MSC – Page 1 of 2

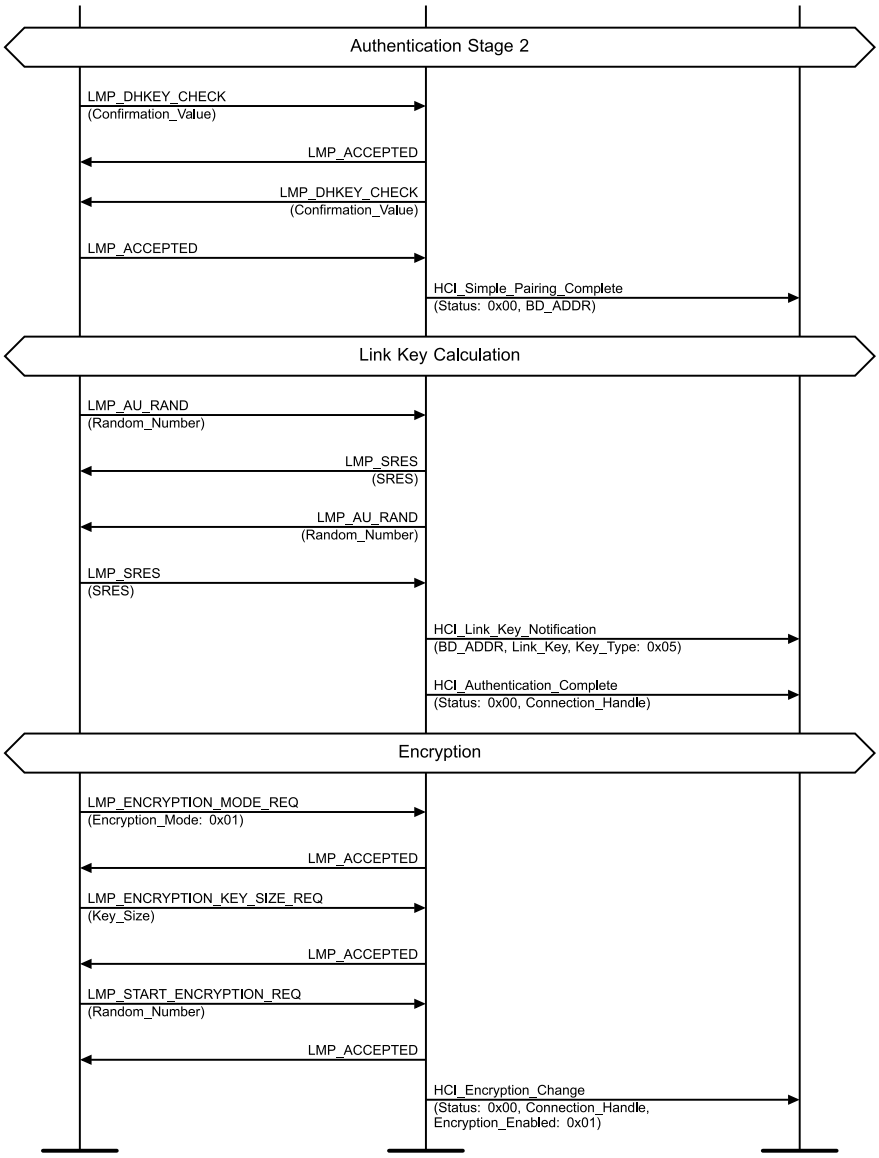


Figure 4.11-30: LMP/SP/BV-13-C [Passkey Entry - IUT Responder – Success] MSC – Page 2 of 2

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.

Repeat Steps 15–20 20 times.

15. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
16. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
18. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
19. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
20. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
21. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
22. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
23. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
24. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
25. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
26. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
27. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
28. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.
29. The IUT sends an HCI_Link_Key_Notification event to the IUT with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192) and a successful HCI_Authentication_Complete event with the Connection_Handle.

30. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption).
31. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
32. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.
33. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
34. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
35. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
36. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

- Expected Outcome

  Pass verdict

  The IUT always generates a unique public key.

  In Step 21, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 25, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 29, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

- Notes

  If the Commitment_Value calculated by the Lower Tester does not match the Commitment_Value sent by the IUT, the Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT with the Authentication_Failure Error_Code.

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

4.11.4.1 Passkey Entry – IUT Initiator – Failure on Initiating Side

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol where the user on the initiating side does not enter the number.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-6.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-14-C [Passkey Entry - IUT Initiator - Failure on Initiating Side]	P-192
LMP/SP/BV-50-C [Passkey Entry – IUT Initiator – Failure on Initiating Side, P-256]	P-256

Table 4.11-6: Passkey Entry – IUT Initiator – Failure on Initiating Side test cases

- Test Procedure

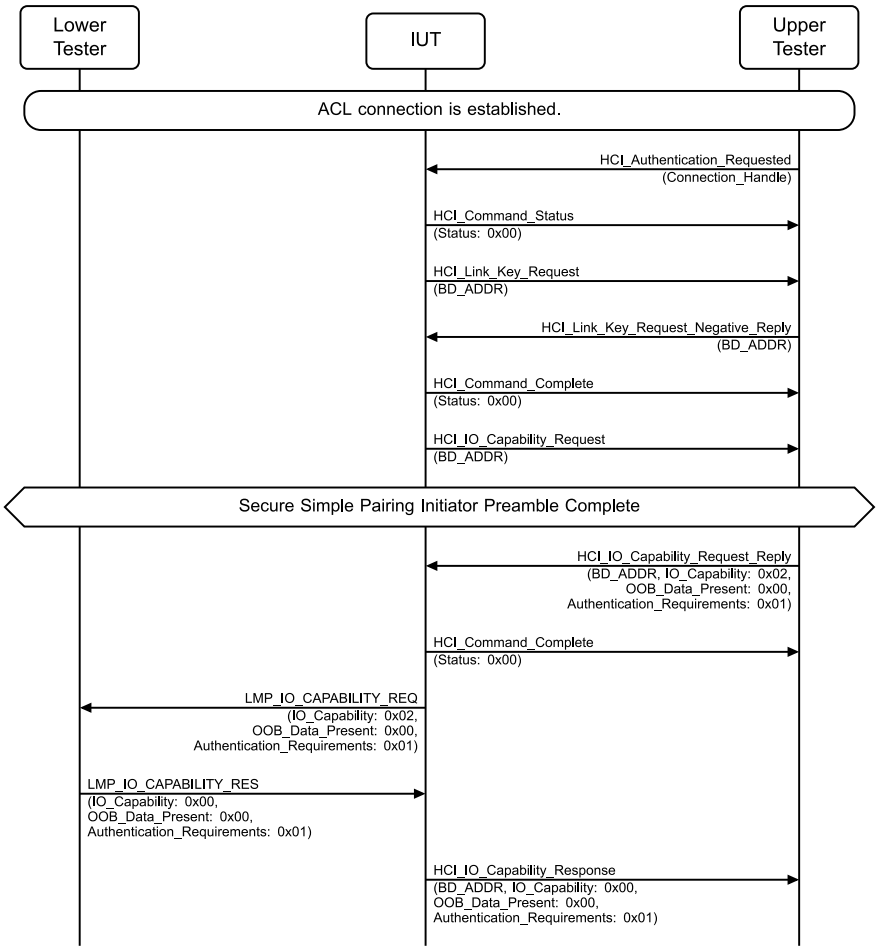


Figure 4.11-31: Passkey Entry – IUT Initiator – Failure on Initiating Side MSC – Page 1 of 2

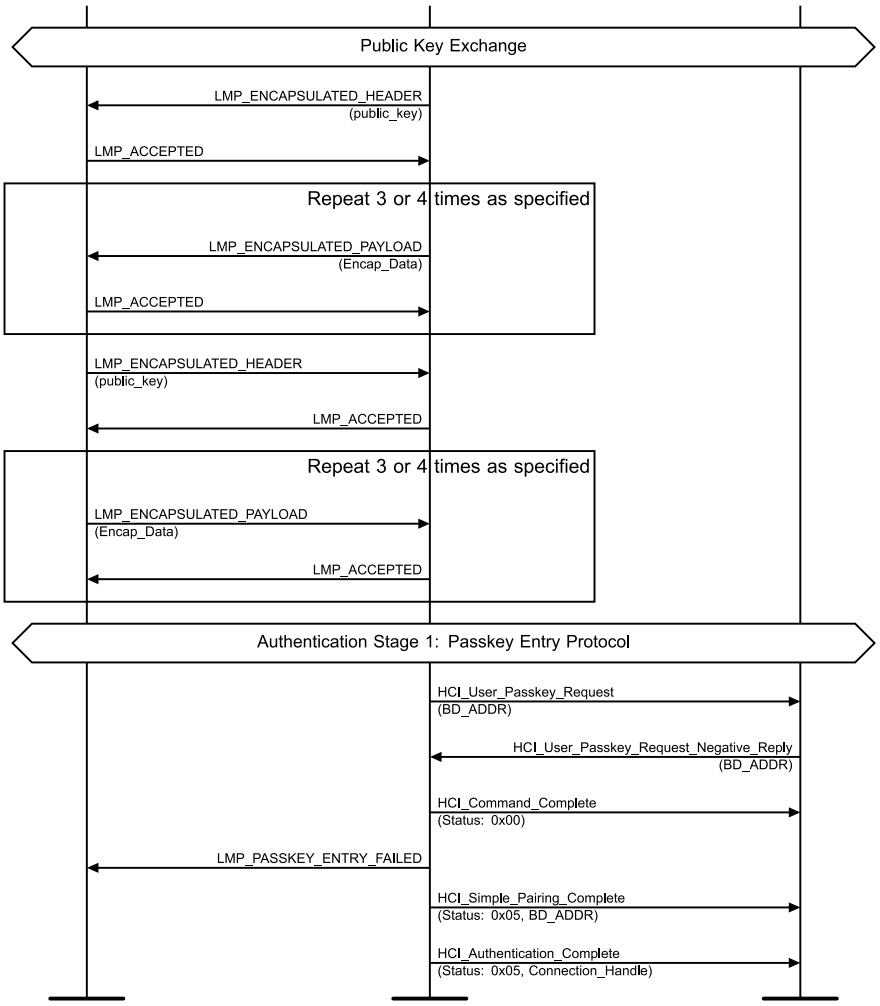


Figure 4.11-32: Passkey Entry – IUT Initiator – Failure on Initiating Side MSC – Page 2 of 2

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-6.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-6.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds to the IUT with an HCI_User_Passkey_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
19. The IUT sends an LMP_PASSKEY_ENTRY_FAILED PDU to the Lower Tester.
20. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  In Step 19, the IUT sends an LMP_PASSKEY_ENTRY_FAILED PDU to the Lower Tester.

  In Step 20, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

4.11.4.2 Passkey Entry – IUT Responder – Failure on Initiating Side

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol and that the IUT responds correctly when the Upper Tester responds that the passkey did not verify correctly.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-7.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-15-C [Passkey Entry - IUT Responder - Failure on Initiating Side]	P-192
LMP/SP/BV-51-C [Passkey Entry – IUT Responder – Failure on Initiating Side, P-256]	P-256

Table 4.11-7: Passkey Entry – IUT Responder – Failure on Initiating Side test cases

- Test Procedure

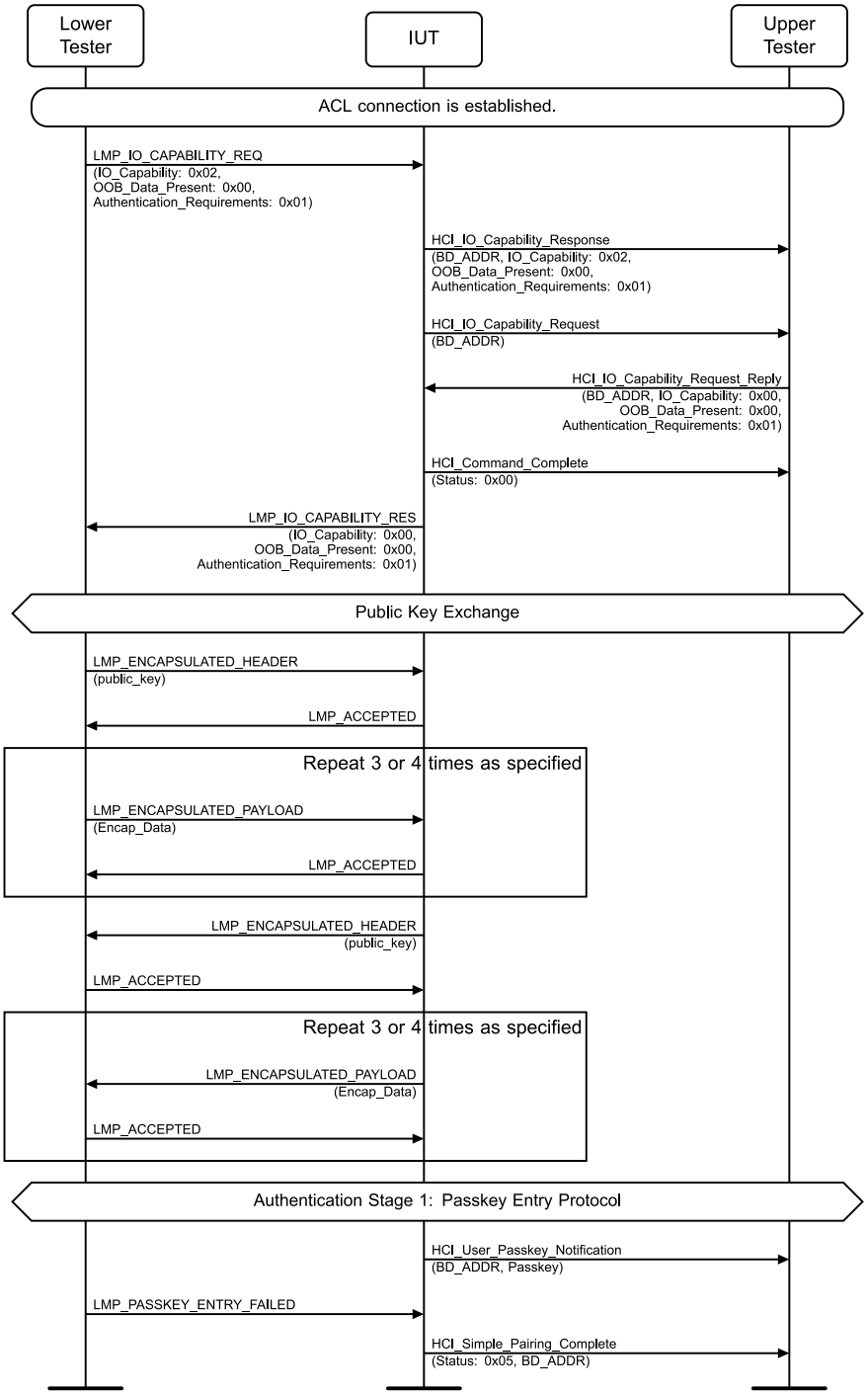


Figure 4.11-33: Passkey Entry – IUT Responder – Failure on Initiating Side MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times if using P-192 or four times if using P-256 as specified in Table 4.11-7.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times if using P-192 or four times if using P-256 as specified in Table 4.11-7.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.
15. The Lower Tester sends an LMP_PASSKEY_ENTRY_FAILED PDU to the IUT.
16. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after receiving the LMP_PASSKEY_ENTRY_FAILED PDU from the Lower Tester.

4.11.4.3 Passkey Entry – IUT Initiator – Failure on Responding Side

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol and that the IUT responds correctly when the responding side fails the passkey entry check step.

- Reference

  [1] 4.2.7

- Initial Condition
  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-8.
  - The IUT is the Initiator.
  - An ACL connection has been established.
- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-16-C [Passkey Entry - IUT Initiator - Failure on Responding Side]	P-192
LMP/SP/BV-52-C [Passkey Entry – IUT Initiator – Failure on Responding Side, P-256]	P-256

Table 4.11-8: Passkey Entry – IUT Initiator – Failure on Responding Side test cases

- Test Procedure

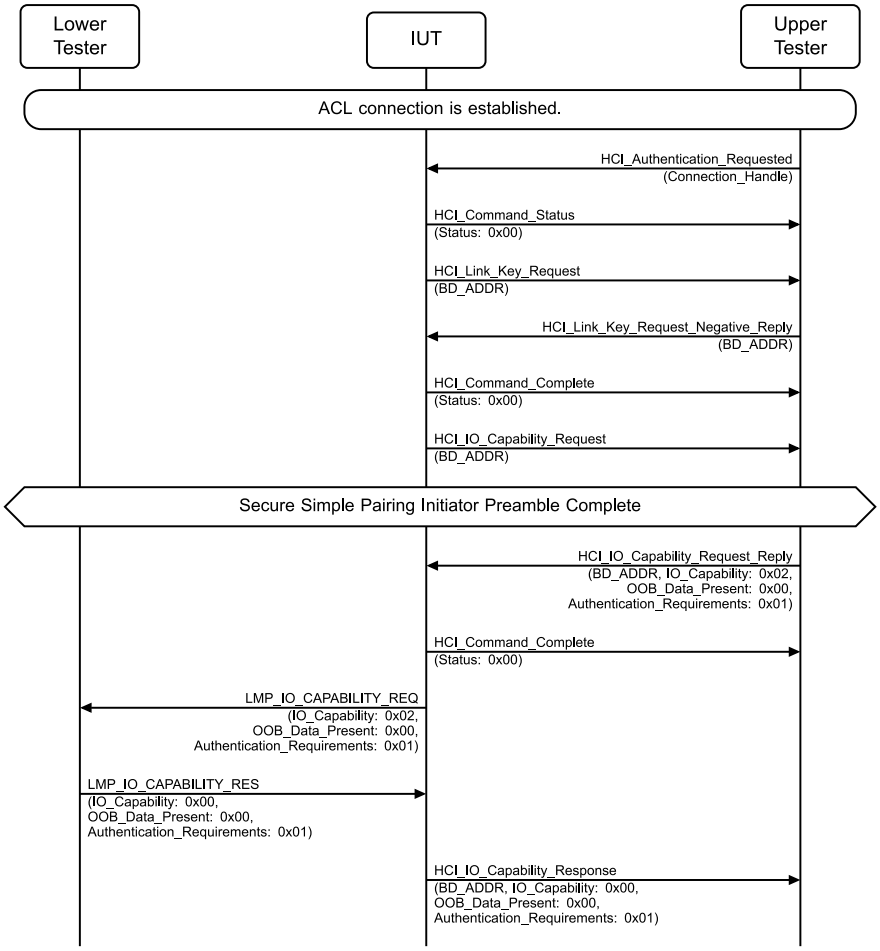


Figure 4.11-34: Passkey Entry - IUT Initiator - Failure on Responding Side MSC – Page 1 of 2

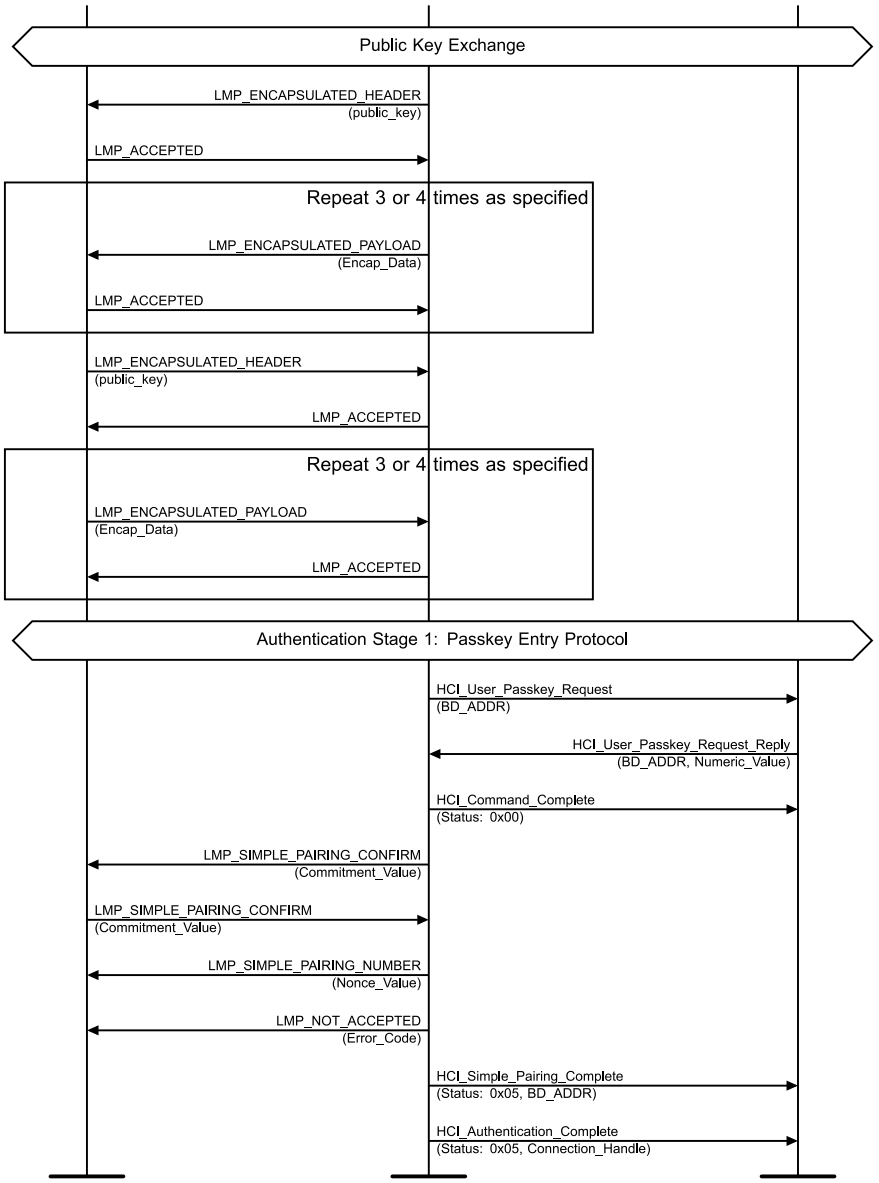


Figure 4.11-35: Passkey Entry - IUT Initiator - Failure on Responding Side MSC – Page 2 of 2

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-8.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-8.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds to the IUT with an HCI_User_Passkey_Request_Reply command with the BD_ADDR and Numeric_Value and receives a successful HCI_Command_Complete event in response.
19. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
20. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
21. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
22. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.
23. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  The IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after the Lower Tester responds to the LMP_SIMPLE_PAIRING_NUMBER PDU with an LMP_NOT_ACCEPTED PDU.

4.11.4.4 Passkey Entry – IUT Responder – Failure on Responding Side

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol and that the IUT responds correctly when the responding side fails the passkey entry check step.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-9.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-17-C [Passkey Entry - IUT Responder - Failure on Responding Side]	P-192
LMP/SP/BV-53-C [Passkey Entry – IUT Responder – Failure on Responding Side, P-256]	P-256

Table 4.11-9: Passkey Entry – IUT Responder – Failure on Responding Side test cases

- Test Procedure

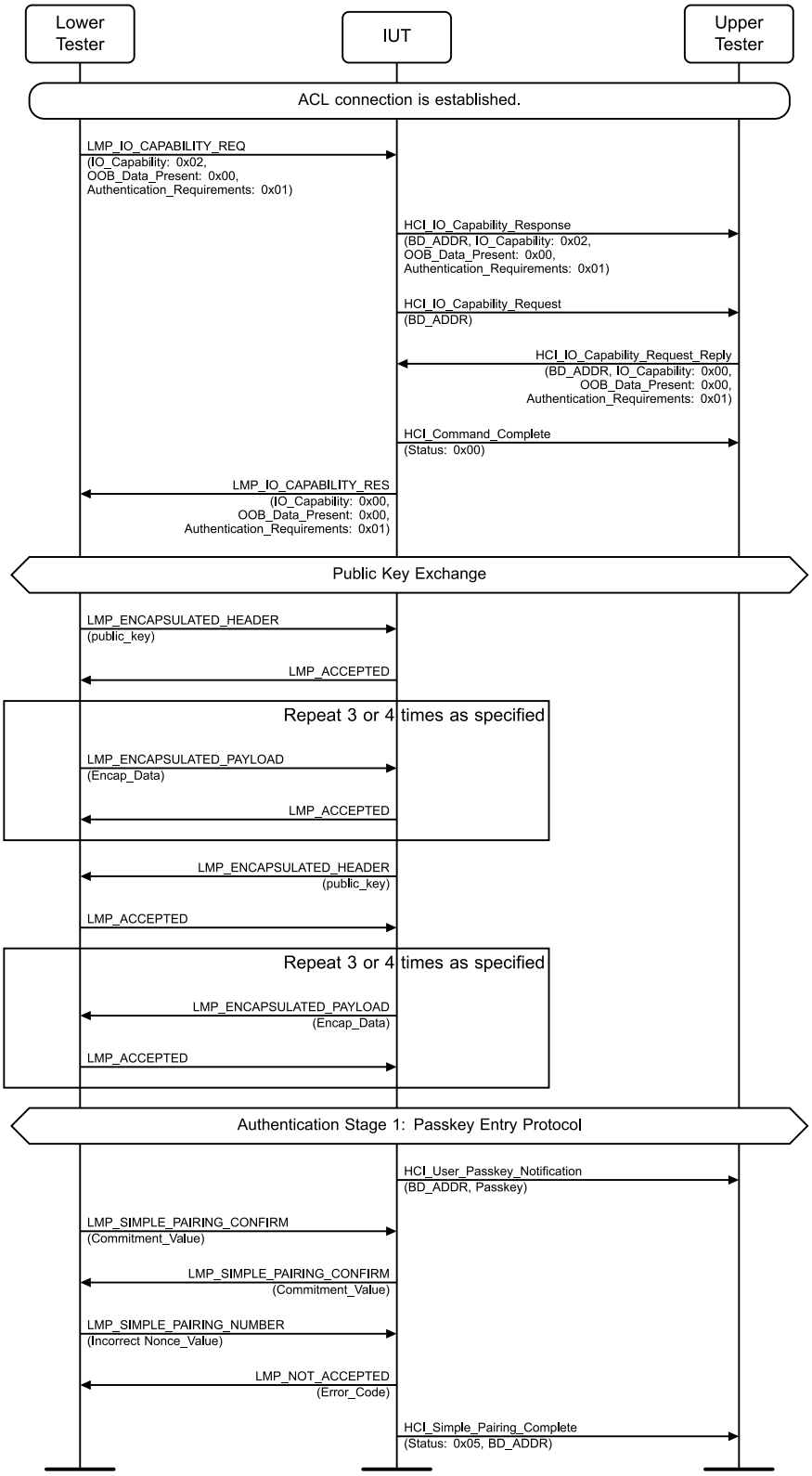


Figure 4.11-36: Passkey Entry - IUT Responder - Failure on Responding Side MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times if using P-192 or four times if using P-256 as specified in Table 4.11-9.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times if using P-192 or four times if using P-256 as specified in Table 4.11-9.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.
15. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
16. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
18. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.
19. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester in response to the LMP_SIMPLE_PAIRING_NUMBER PDU from the Lower Tester.

  In Step 19, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

LMP/SP/BV-48-C [Passkey Entry – IUT Initiator – Success, P-256]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol using the P-256 elliptic curve.

- Reference

  [1] 4.2.7

- Initial Condition
  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

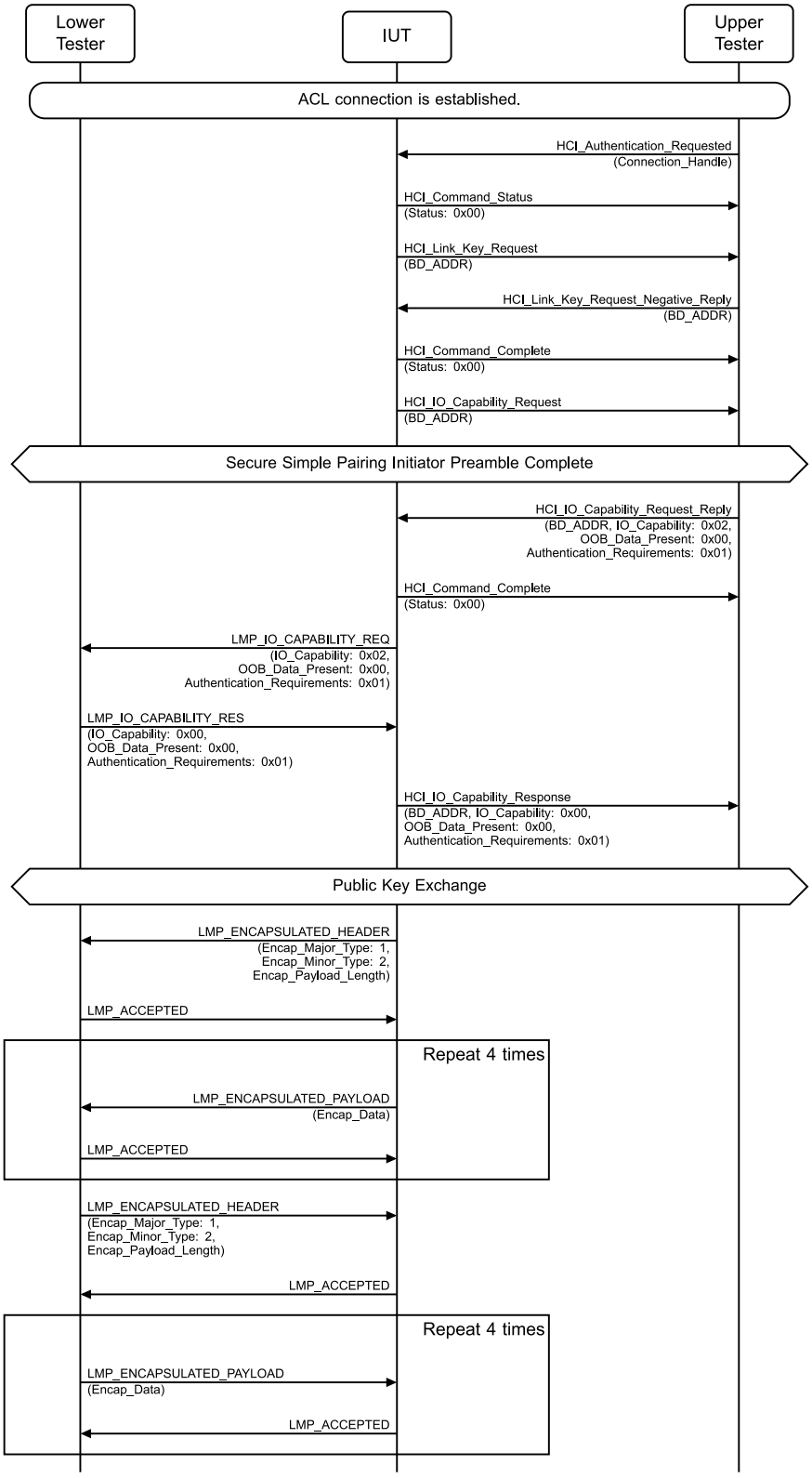


Figure 4.11-37: LMP/SP/BV-48-C [Passkey Entry – IUT Initiator – Success, P-256] MSC – Page 1 of 3

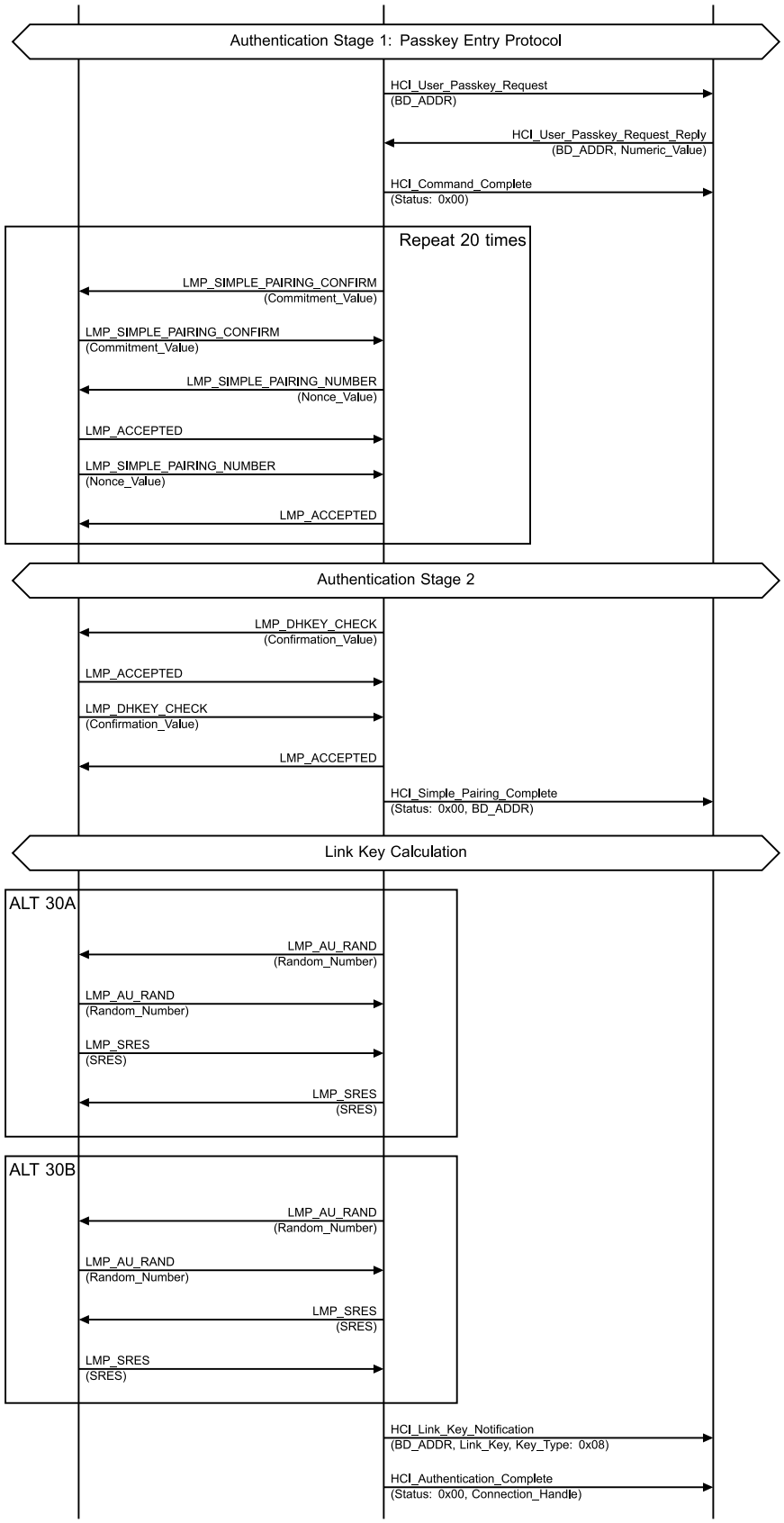


Figure 4.11-38: LMP/SP/BV-48-C [Passkey Entry – IUT Initiator – Success, P-256] MSC – Page 2 of 3

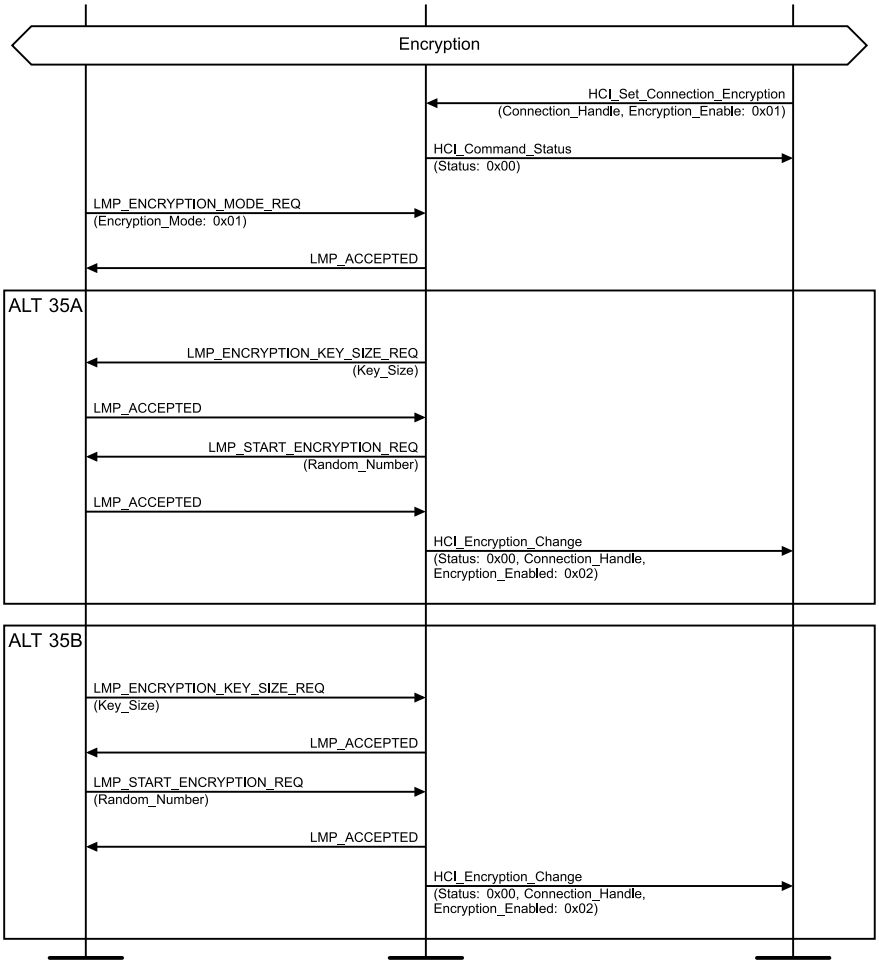


Figure 4.11-39: LMP/SP/BV-48-C [Passkey Entry – IUT Initiator – Success, P-256] MSC – Page 3 of 3

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 four times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 four times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds to the IUT with an HCI_User_Passkey_Request_Reply command with the BD_ADDR and Numeric_Value and receives a successful HCI_Command_Complete event in response.

Repeat Steps 19–24 20 times.

19. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
20. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
21. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
22. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
23. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
24. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
25. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
26. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
27. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
28. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
29. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.

30. Perform either alternative 30A or 30B depending on the IUT's role.
    Alternative 30A (The IUT is the Central):
    - 30A.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
    - 30A.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
    - 30A.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.
    - 30A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES.

    Alternative 30B (The IUT is the Peripheral):
    - 30B.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
    - 30B.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
    - 30B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
    - 30B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES.
31. The IUT sends an HCI_Link_Key_Notification event to the IUT with the BD_ADDR, Link_Key, and Key_Type set to 0x08 (Authenticated Combination Key generated from P-256) and a successful HCI_Authentication_Complete event with the Connection_Handle.
32. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response.
33. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption).
34. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
35. Perform either alternative 35A or 35B depending on the IUT's role.
    Alternative 35A (The IUT is the Central):
    - 35A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
    - 35A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    - 35A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
    - 35A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    - 35A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

    Alternative 35B (The IUT is the Peripheral):
    - 35B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.
    - 35B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    - 35B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
    - 35B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    - 35B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

- Expected Outcome

  Pass verdict

  In Step 25, the IUT sends the LMP_DHKEY_CHECK PDU with a valid Confirmation_Value to the Lower Tester.

  In Step 29, the IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 31, the IUT sends the resulting Link_Key and Key_Type to the Upper Tester in an HCI_Link_Key_Notification event. The Link_Key matches at the Upper and Lower Testers, and the Key_Type is 0x08 (Authenticated Combination Key generated from P-256).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to an LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-49-C [Passkey Entry – IUT Responder – Success, P-256]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol using the P-256 elliptic curve.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Procedure

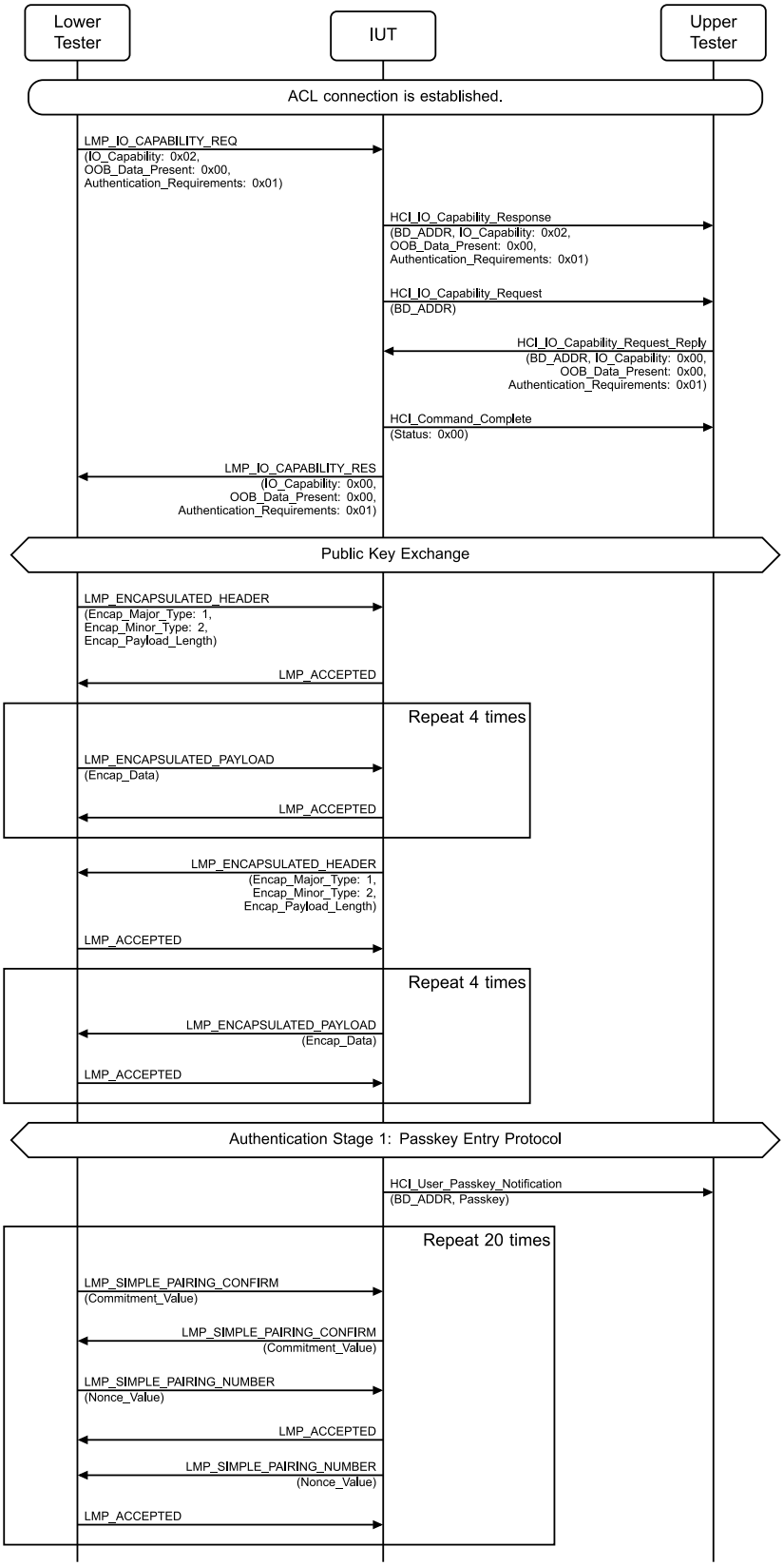


Figure 4.11-40: LMP/SP/BV-49-C [Passkey Entry – IUT Responder – Success, P-256] MSC – Page 1 of 2

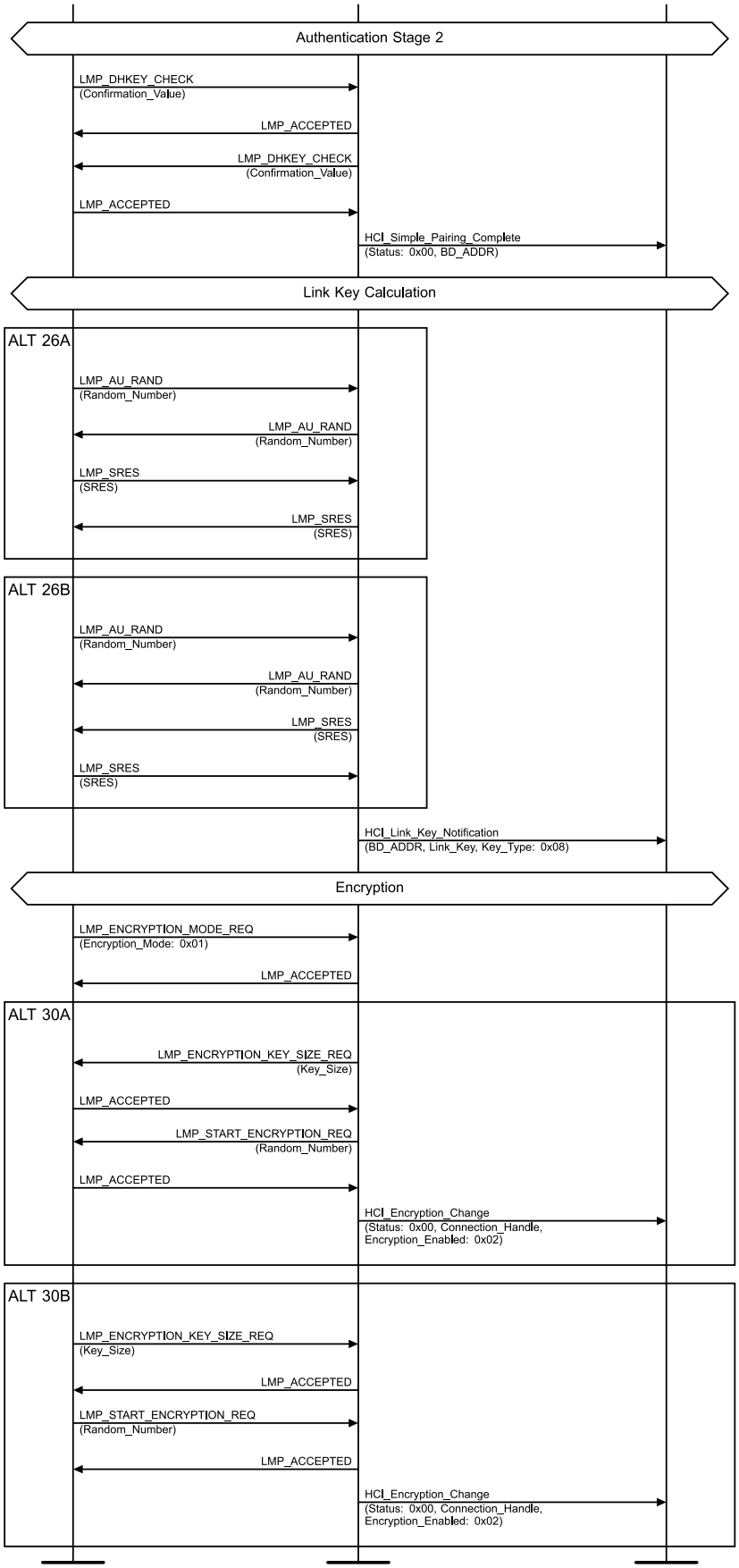


Figure 4.11-41: LMP/SP/BV-49-C [Passkey Entry – IUT Responder – Success, P-256] MSC – Page 2 of 2

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 four times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 four times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.

Repeat Steps 15–20 20 times.

15. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
16. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
17. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
18. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
19. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
20. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.

21. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
22. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
23. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
24. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
25. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
26. Perform either alternative 26A or 26B depending on the IUT's role.

    Alternative 26A (The IUT is the Central):

    26A.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.

    26A.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.

    26A.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.

    26A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES.

    Alternative 26B (The IUT is the Peripheral):

    26B.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.

    26B.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.

    26B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.

    26B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES.
27. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x08 (Authenticated Combination Key generated from P-256).
28. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption).
29. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
30. Perform either alternative 30A or 30B depending on the IUT's role.

    Alternative 30A (The IUT is the Central):

    30A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.

    30A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    30A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

    30A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    30A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

    Alternative 30B (The IUT is the Peripheral):

    30B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

    30B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    30B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

    30B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

30B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

- Expected Outcome

  Pass verdict

  In Step 14, the IUT sends the HCI_User_Passkey_Notification event to the Upper Tester.

  In Step 23, the IUT sends the LMP_DHKEY_CHECK PDU with a valid Confirmation_Value to the Lower Tester.

  In Step 25, the IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 27, the IUT sends the resulting Link_Key and Key_Type to the Upper Tester in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x08 (Authenticated Combination Key generated from P-256).

- Notes

  If the Commitment_Value calculated by the Lower Tester does not match the Commitment_Value sent by the IUT, the Lower Tester sends an LMP_NOT_ACCEPTED PDU with the Authentication_Failure Error_Code.

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-67-C [Passkey Entry – IUT Initiator, Verify Random Passkeys]

- Test Purpose

  Verify that the IUT randomly generates unique passkeys.

- Reference

  [1] 4.2.7

  [16] 7.2.3

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Initiator.

- Test Procedure

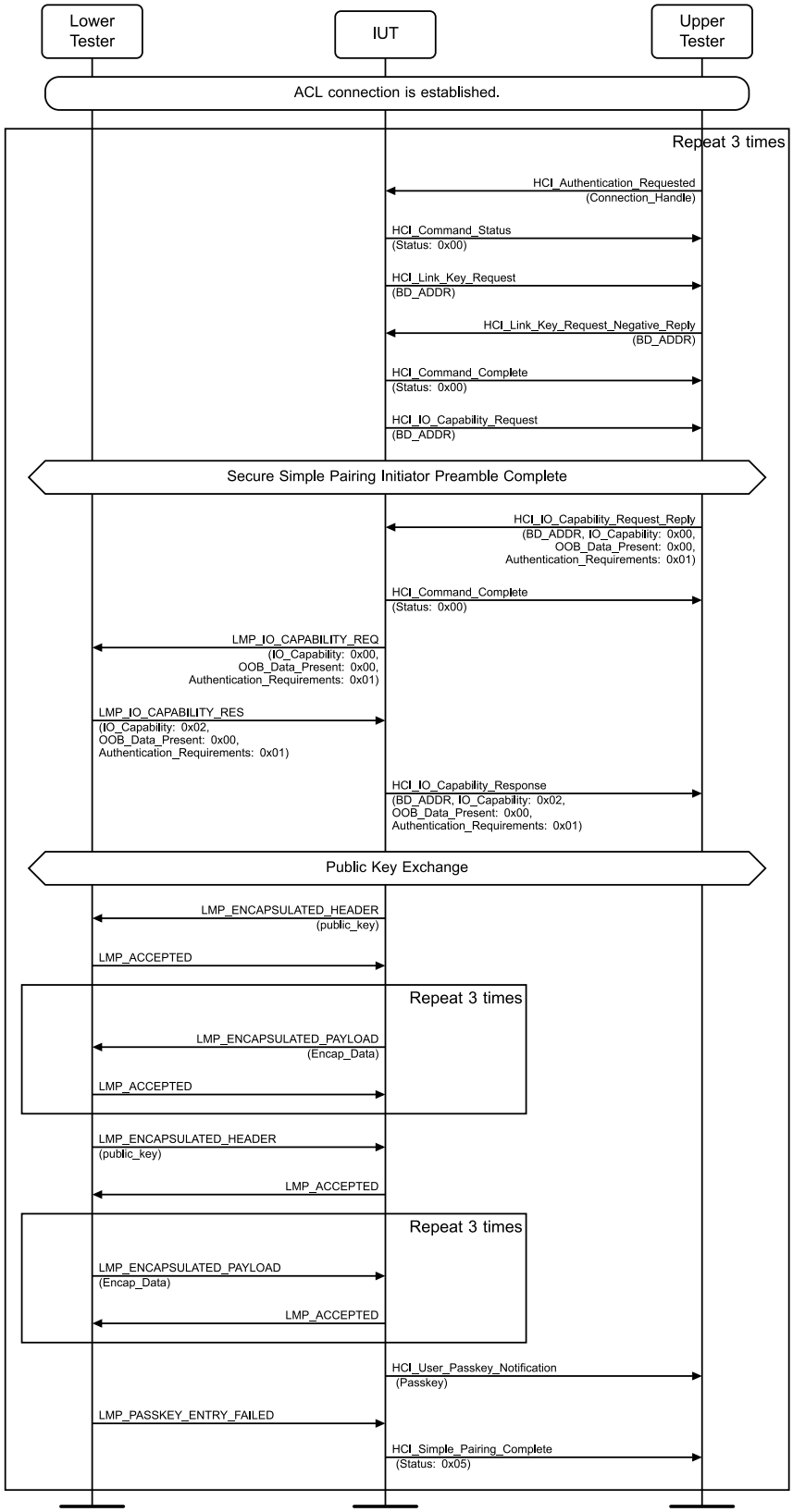


Figure 4.11-42: LMP/SP/BV-67-C [Passkey Entry – IUT Initiator, Verify Random Passkeys] MSC

Repeat the test procedure three times. In each repeat, the Lower Tester uses the same public key.

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with a Passkey.
18. The Lower Tester sends an LMP_PASSKEY_ENTRY_FAILED PDU to the IUT.
19. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05.

- Expected Outcome

  Pass verdict

  In Step 17, the IUT sends different passkeys to the Upper Tester.

LMP/SP/BV-68-C [Passkey Entry – IUT Responder – Verify Random Passkeys]

- Test Purpose

  Verify that the IUT randomly generates unique passkeys.

- Reference

  [1] 4.2.7

  [16] 7.2.3

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Responder.

- Test Procedure

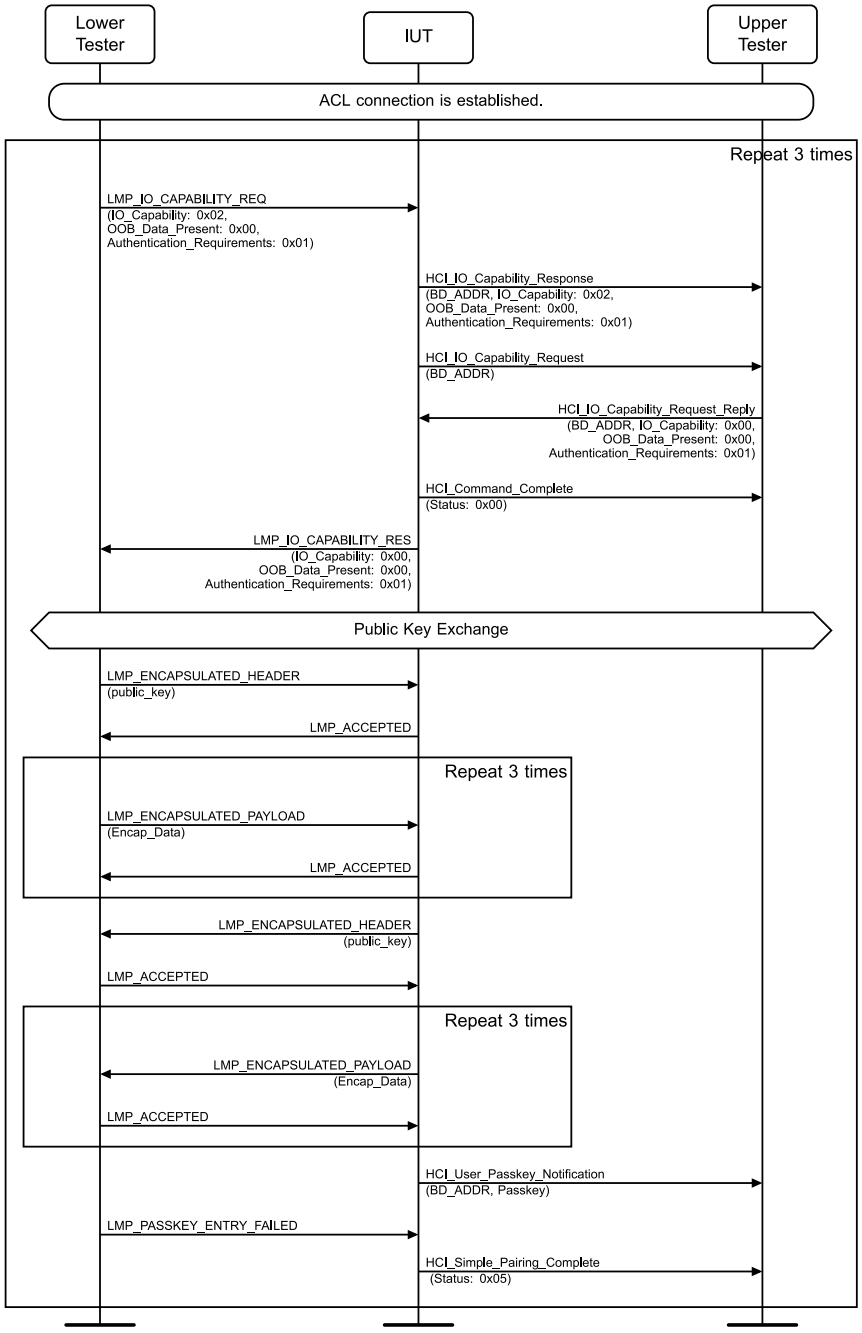


Figure 4.11-43: LMP/SP/BV-68-C [Passkey Entry – IUT Responder, Verify Random Passkeys] MSC

Repeat the test procedure three times. In each repeat, the Lower Tester uses the same public key.

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.

4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.
15. The Lower Tester sends an LMP_PASSKEY_ENTRY_FAILED PDU to the IUT.
16. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05.

- Expected Outcome

  Pass verdict

  In Step 14, the IUT sends different passkeys to the Upper Tester.

4.11.5 Out-of-Band procedures

Note: In all the test cases in Section 4.11.5 Out-of-Band procedures, it does not matter whether the IUT is the Central or the Peripheral.

4.11.5.1 OOB Protocol – IUT Initiator – Success

- Test Purpose

  Verify that the IUT supports the OOB protocol when the IUT or the Lower Tester have OOB_Auth_Data.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Initiator.
  - If the Lower Tester has OOB data as noted in Table 4.11-10, the Upper Tester has sent the HCI_Read_Local_OOB_Data command to the IUT and received a successful HCI_Command_Complete event with the C and R in response before the ACL connection is established between the IUT and the Lower Tester.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	IUT OOB	Lower Tester OOB	Random Key
LMP/SP/BV-18-C [OOB Protocol - IUT Initiator - IUT with OOB_Auth_Data – Success, v6.2 and earlier]	Yes	No	May be repeated
LMP/SP/BV-20-C [OOB Protocol - IUT Initiator – Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier]	No	Yes	May be repeated
LMP/SP/BV-22-C [OOB Protocol - IUT Initiator - IUT and Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier]	Yes	Yes	May be repeated
LMP/SP/BV-70-C [OOB Protocol - IUT Initiator - IUT with OOB_Auth_Data – Success, v6.3 and later]	Yes	No	Unique
LMP/SP/BV-71-C [OOB Protocol - IUT Initiator – Lower Tester with OOB_Auth_Data – Success, v6.3 and later]	No	Yes	Unique
LMP/SP/BV-72-C [OOB Protocol - IUT Initiator - IUT and Lower Tester with OOB_Auth_Data – Success, v6.3 and later]	Yes	Yes	Unique

Table 4.11-10: OOB Protocol – IUT Initiator – Success test cases

- Test Procedure

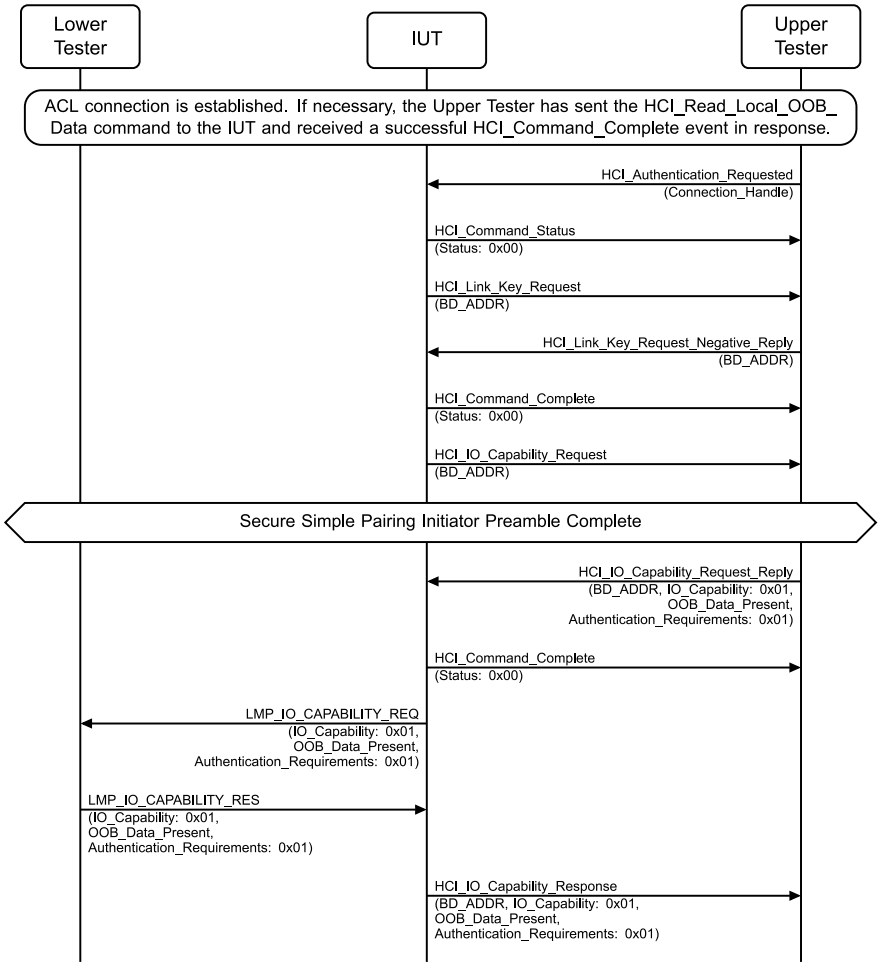


Figure 4.11-44: OOB Protocol – IUT Initiator – Success MSC – Page 1 of 3

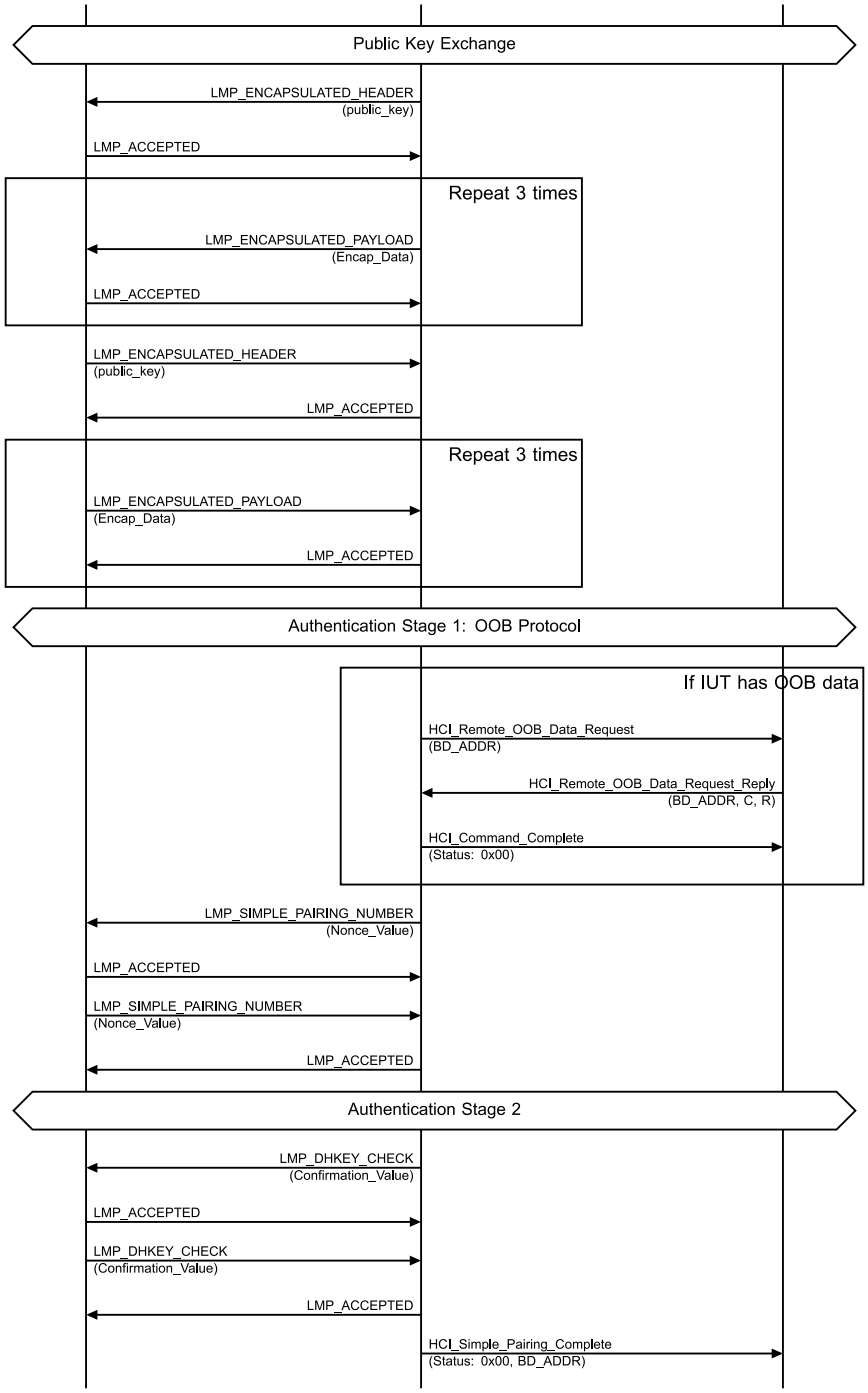


Figure 4.11-45: OOB Protocol – IUT Initiator – Success MSC – Page 2 of 3

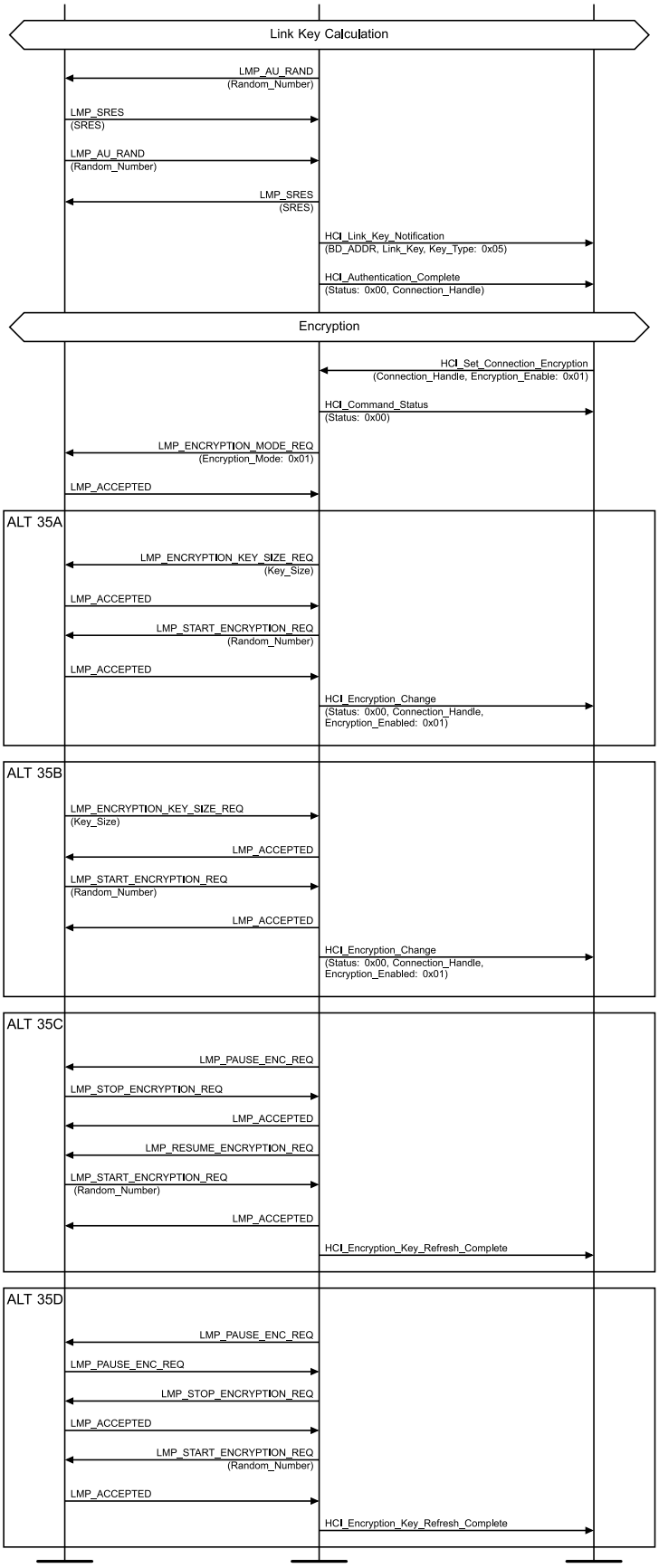


Figure 4.11-46: OOB Protocol – IUT Initiator – Success MSC – Page 3 of 3

Repeat the Test Procedure five times.

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x01 (P-192 OOB authentication data from remote device present) if the IUT has OOB data specified in Table 4.11-10, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the IUT has OOB data specified in Table 4.11-10, otherwise set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the Lower Tester has OOB data specified in Table 4.11-10, otherwise set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x01 (P-192 OOB authentication data from remote device present) if the Lower Tester has OOB data specified in Table 4.11-10, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.

Execute Steps 17–18 only if the IUT has OOB data specified in Table 4.11-10.

17. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds with an HCI_Remote_OOB_Data_Request_Reply command with the BD_ADDR, C, and R and receives an HCI_Command_Complete event in response.
19. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
20. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
21. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
22. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
23. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
24. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
25. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
26. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
27. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
28. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
29. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
30. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
31. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192) and a successful HCI_Authentication_Complete event with the Connection_Handle to the IUT.
32. On the first procedure iteration, the Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response; otherwise, skip Steps 33 and 34 and perform alternative 35C or 35D depending on the IUT's role.
33. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
34. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
35. Perform either alternative 35A or 35B depending on the IUT's role.

    Alternative 35A (The IUT is the Central):

    35A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.

    35A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    35A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

    35A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    35A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

Alternative 35B (The IUT is the Peripheral):

35B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

35B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

35B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

35B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

35B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

Alternative 35C (The IUT is the Peripheral):

35C.1. The IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.

35C.2. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.

35C.3. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.

35C.4. The IUT sends an LMP_RESUME_ENCRYPTION_REQ PDU to the Lower Tester.

35C.5. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.

35C.6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

35C.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.

Alternative 35D (The IUT is the Central):

35D.1. The IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.

35D.2. The Lower Tester sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the IUT.

35D.3. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.

35D.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.

35D.5. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.

35D.6. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

35D.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.

36. The Upper Tester sends an HCI_Delete_Stored_Link_Key command to the IUT with BD_ADDR set to the Lower Tester BD_ADDR and Delete_All set to 0x00 and receives a successful HCI_Command_Complete event in response.

- Expected Outcome

  Pass verdict

  The IUT always generates a unique public key.

  If the IUT has OOB data specified in Table 4.11-10: In Step 17, the IUT sends the HCI_Remote_OOB_Data_Request event to the Upper Tester.

  If the IUT does not have OOB data specified in Table 4.11-10: The IUT does not send the HCI_Remote_OOB_Data_Request event to the Upper Tester.

  In Step 23, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 27, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

If specified in Table 4.11-10, the random number sent by the IUT in Step 28 is always unique.

In Step 31, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to an LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

4.11.5.2 OOB Protocol – IUT Responder – Success

- Test Purpose

  Verify that the IUT supports the OOB protocol when the IUT or the Lower Tester has OOB_Auth_Data.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Responder.
  - If the Lower Tester has OOB data as noted in Table 4.11-11, the Upper Tester has sent the HCI_Read_Local_OOB_Data command to the IUT and received a successful HCI_Command_Complete event with the C and R in response before the ACL connection is established between the IUT and the Lower Tester.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	IUT OOB	Lower Tester OOB	Random Key
LMP/SP/BV-19-C [OOB Protocol - IUT Responder - IUT with OOB_Auth_Data – Success, v6.2 and earlier]	Yes	No	May be repeated
LMP/SP/BV-21-C [OOB Protocol - IUT Responder – Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier]	No	Yes	May be repeated
LMP/SP/BV-23-C [OOB Protocol - IUT Responder - IUT and Lower Tester with OOB_Auth_Data – Success, v6.2 and earlier]	Yes	Yes	May be repeated
LMP/SP/BV-73-C [OOB Protocol - IUT Responder - IUT with OOB_Auth_Data – Success, v6.3 and later]	Yes	No	Unique
LMP/SP/BV-74-C [OOB Protocol - IUT Responder – Lower Tester with OOB_Auth_Data – Success, v6.3 and later]	No	Yes	Unique
LMP/SP/BV-75-C [OOB Protocol - IUT Responder - IUT and Lower Tester with OOB_Auth_Data – Success, v6.3 and later]	Yes	Yes	Unique

Table 4.11-11: OOB Protocol – IUT Responder – Success test cases

- Test Procedure

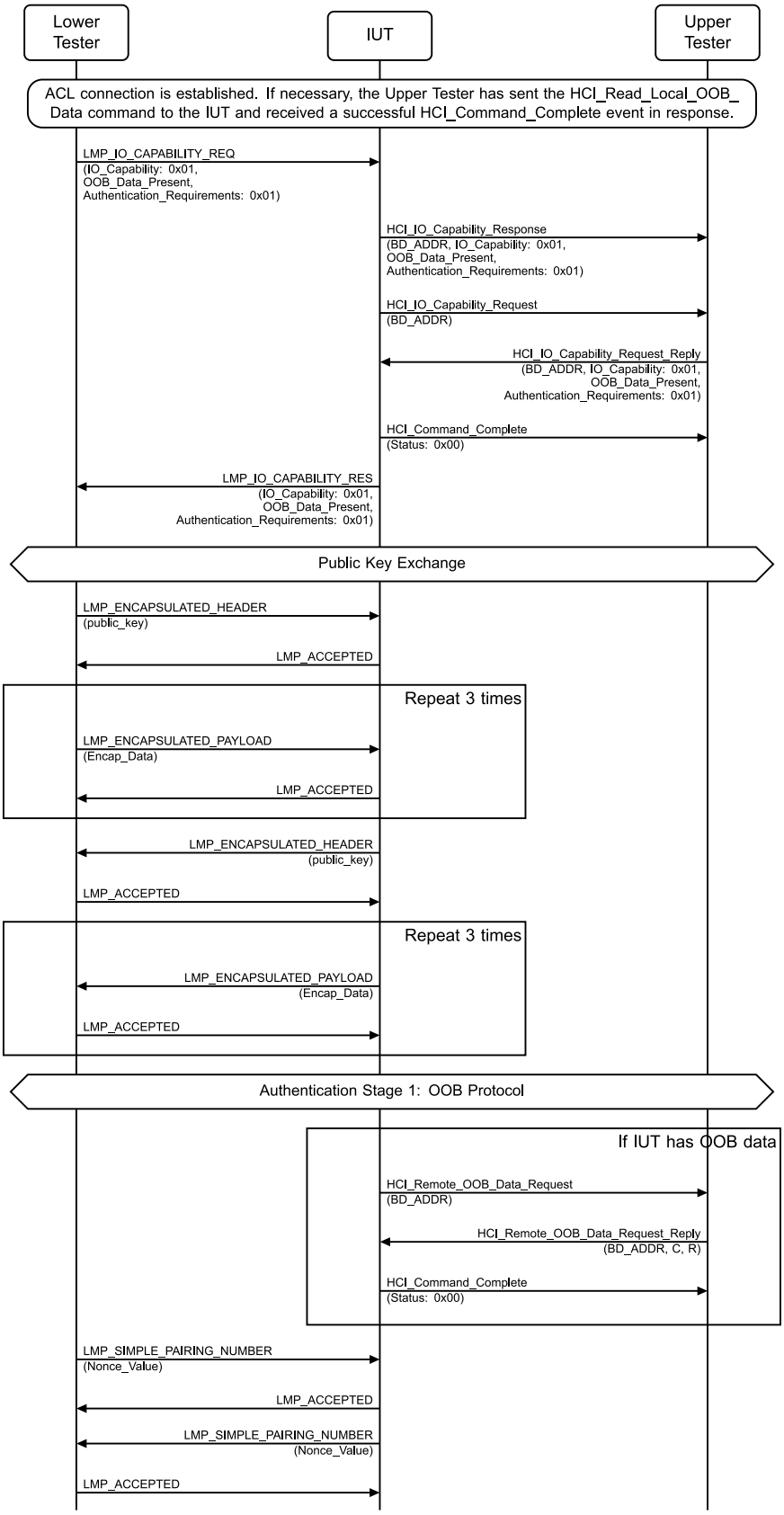


Figure 4.11-47: OOB Protocol – IUT Responder – Success MSC – Page 1 of 3

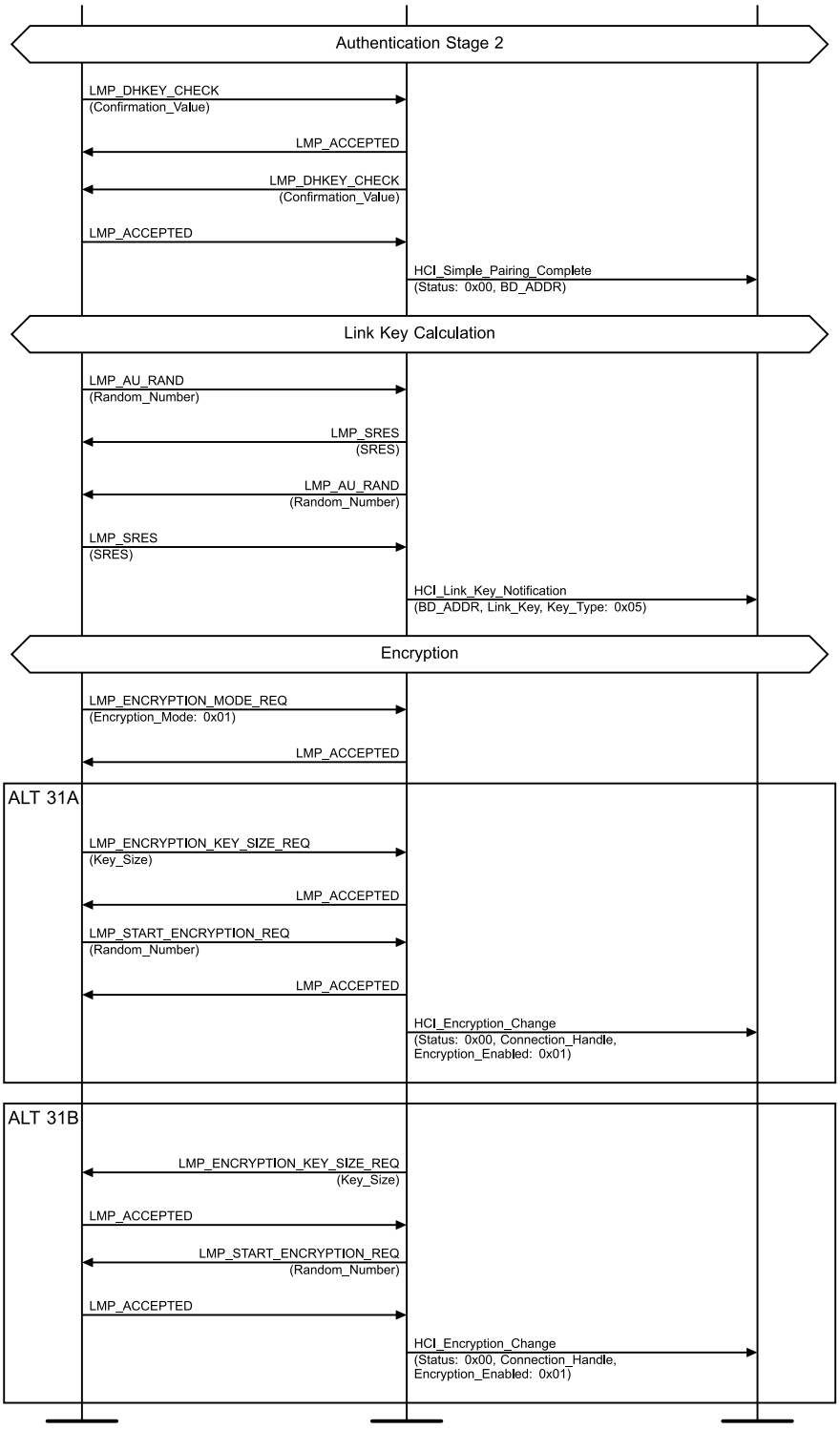


Figure 4.11-48: OOB Protocol – IUT Responder – Success MSC – Page 2 of 3

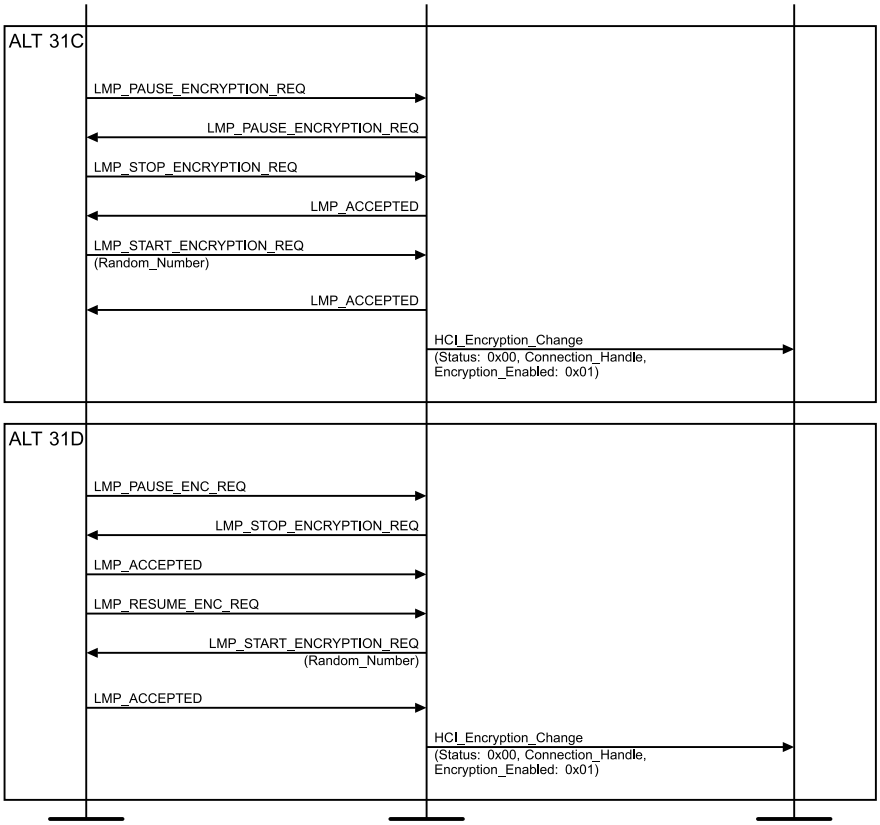


Figure 4.11-49: OOB Protocol – IUT Responder – Success MSC – Page 3 of 3

Repeat the Test Procedure five times.

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the Lower Tester has OOB data specified in Table 4.11-11, otherwise set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x01 (P-192 OOB authentication data from remote device present) if the Lower Tester has OOB data specified in Table 4.11-11, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x01 (P-192 OOB authentication data from remote device present) if the IUT has OOB data specified in Table 4.11-11, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the IUT has OOB data specified in Table 4.11-11, otherwise set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.

7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.

Execute Steps 14–15 only if the IUT has OOB data specified in Table 4.11-11.

14. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.
15. The Upper Tester responds with an HCI_Remote_OOB_Data_Request_Reply command with the BD_ADDR, C, and R and receives an HCI_Command_Complete event in response.
16. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
18. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
22. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
23. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
24. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
25. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
26. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
27. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.
28. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192).
29. On the first procedure iteration, the Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption); otherwise, skip Steps 30 and 31 and perform alternative 31C or 31D depending on the IUT's role.
30. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.

31. Perform either alternative 31A or 31B depending on the IUT's role.
    Alternative 31A (The IUT is the Peripheral):
    - 31A.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.
    - 31A.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    - 31A.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
    - 31A.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    - 31A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

    Alternative 31B (The IUT is the Central):
    - 31B.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
    - 31B.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
    - 31B.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
    - 31B.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    - 31B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

    Alternative 31C (The IUT is the Peripheral):
    - 31C.1. The Lower Tester sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the IUT.
    - 31C.2. The IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
    - 31C.3. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
    - 31C.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
    - 31C.5. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
    - 31C.6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    - 31C.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.

    Alternative 31D (The IUT is the Central):
    - 31D.1. The Lower Tester sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the IUT.
    - 31D.2 The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
    - 31D.3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
    - 31D.4. The Lower Tester sends an LMP_RESUME_ENCRYPTION_REQ PDU to the IUT.
    - 31D.5. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
    - 31D.6. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
    - 31D.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle.
32. The Upper Tester sends an HCI_Delete_Stored_Link_Key command to the IUT with BD_ADDR set to the Lower Tester BD_ADDR and Delete_All set to 0x00 and receives a successful HCI_Command_Complete event in response.

- Expected Outcome

  Pass verdict

  The IUT always generates a unique public key.

  If the IUT has OOB data specified in Table 4.11-11: In Step 14, the IUT sends the HCI_Remote_OOB_Data_Request event to the Upper Tester.

  If the IUT does not have OOB data specified in Table 4.11-11: The IUT does not send the HCI_Remote_OOB_Data_Request event to the Upper Tester.

  In Step 22, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 24, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

  If specified in Table 4.11-11, the random number sent by the IUT in Step 26 is always unique.

  In Step 28, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

4.11.5.3 OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Failure

- Test Purpose

  Verify that the IUT supports the OOB protocol when the IUT has OOB_Auth_Data that does not match the Lower Tester.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-12.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-24-C [OOB Protocol - IUT Initiator - IUT with OOB_Auth_Data – Failure]	P-192
LMP/SP/BV-60-C [OOB Protocol – IUT Initiator - IUT with OOB_Auth_Data – Failure, P-256]	P-256

Table 4.11-12: OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Failure test cases

- Test Procedure

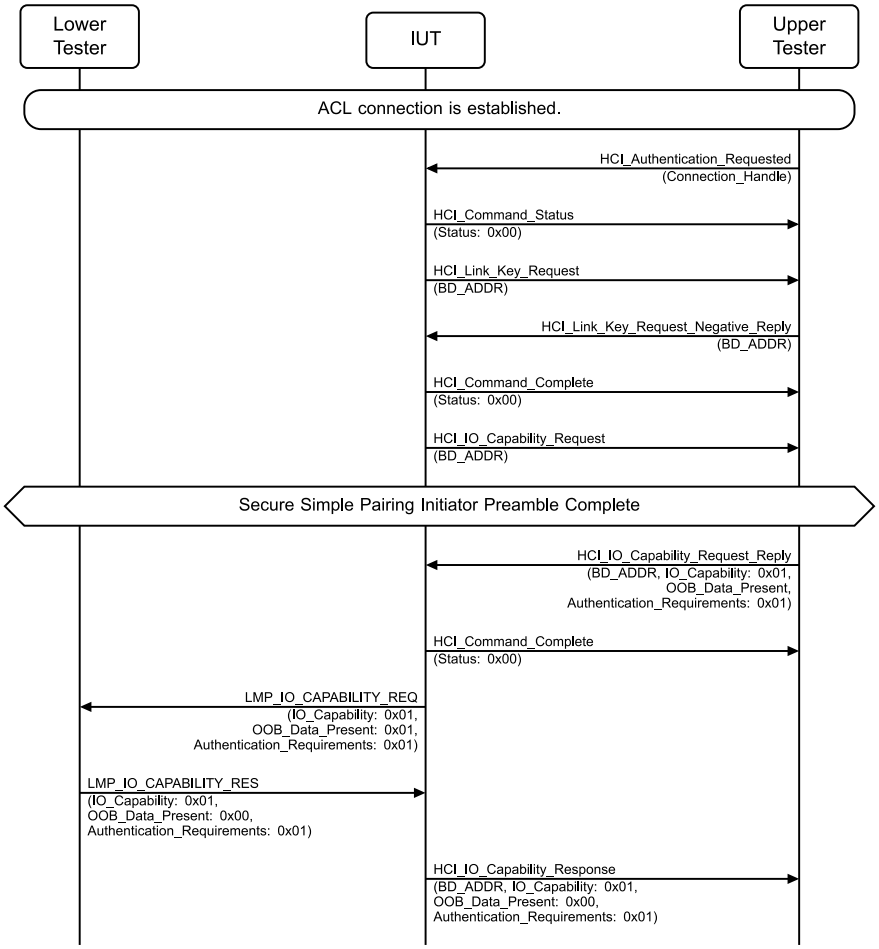


Figure 4.11-50: OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Failure MSC – Page 1 of 2

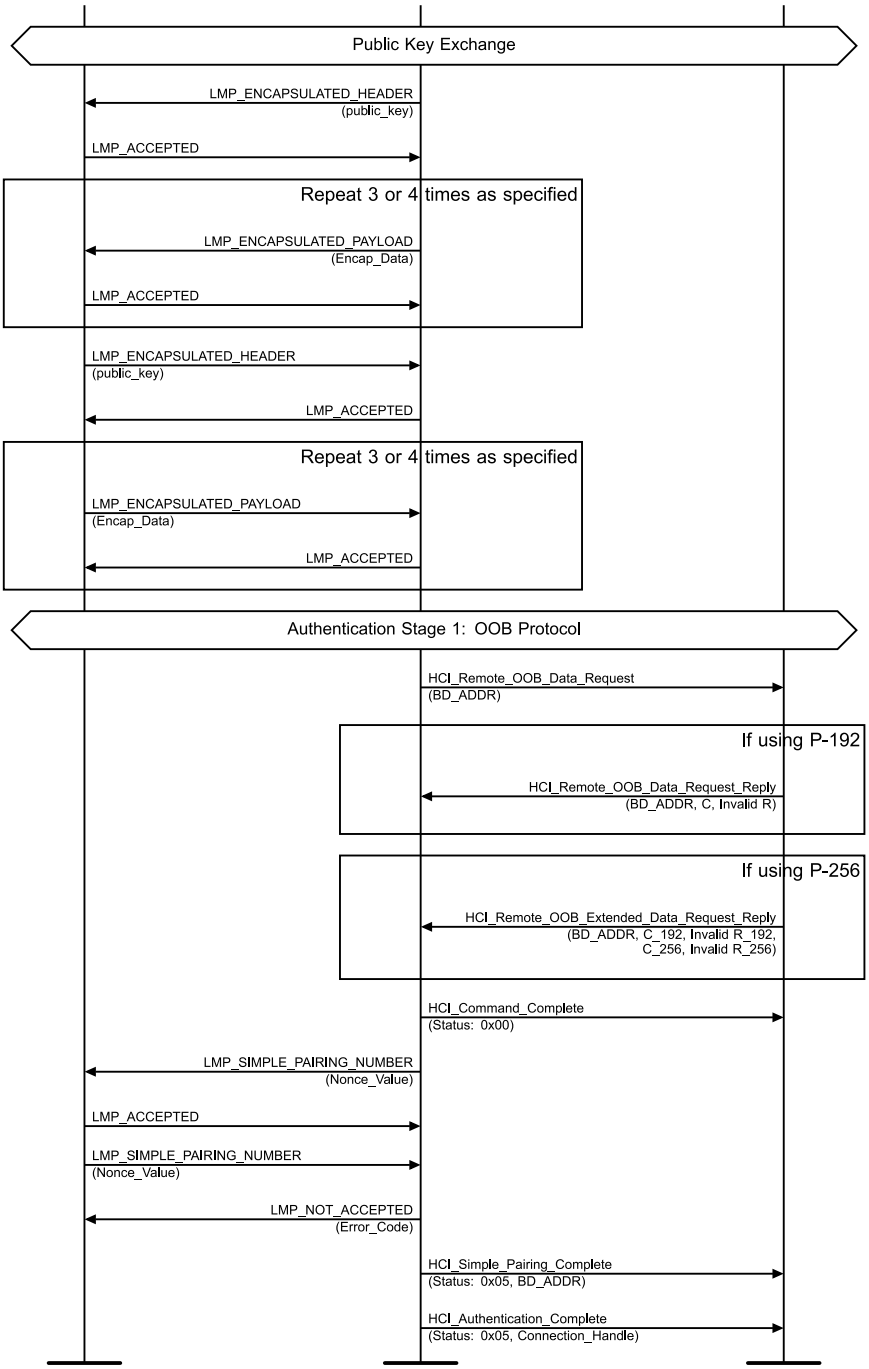


Figure 4.11-51: OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Failure MSC – Page 2 of 2

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x03 (P-192 and P-256 OOB authentication data from remote device present) if using P-256 as

specified in Table 4.11-12, otherwise set to 0x01 (P-192 OOB authentication data from remote device present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.

6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-12.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-12.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds with an HCI_Remote_OOB_Data_Request_Reply command with the BD_ADDR, C, and an invalid R if using P-192 or an HCI_Remote_OOB_Extended_Data_Request_Reply command with the BD_ADDR, C_192, invalid R_192, C_256, and invalid R_256 if using P-256 as specified in Table 4.11-12 and receives an HCI_Command_Complete event in response.
19. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
20. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
21. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
22. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.

23. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  The IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after sending the LMP_NOT_ACCEPTED PDU to the Lower Tester.

4.11.5.4 OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Failure

- Test Purpose

  Verify that the IUT supports the OOB protocol when the IUT has OOB_Auth_Data that does not match the Lower Tester.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section if using P-192 or the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings if using P-256 as specified in Table 4.11-13.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-25-C [OOB Protocol - IUT Responder - IUT with OOB_Auth_Data – Failure]	P-192
LMP/SP/BV-61-C [OOB Protocol – IUT Responder - IUT with OOB_Auth_Data – Failure, P-256]	P-256

Table 4.11-13: OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Failure test cases

- Test Procedure

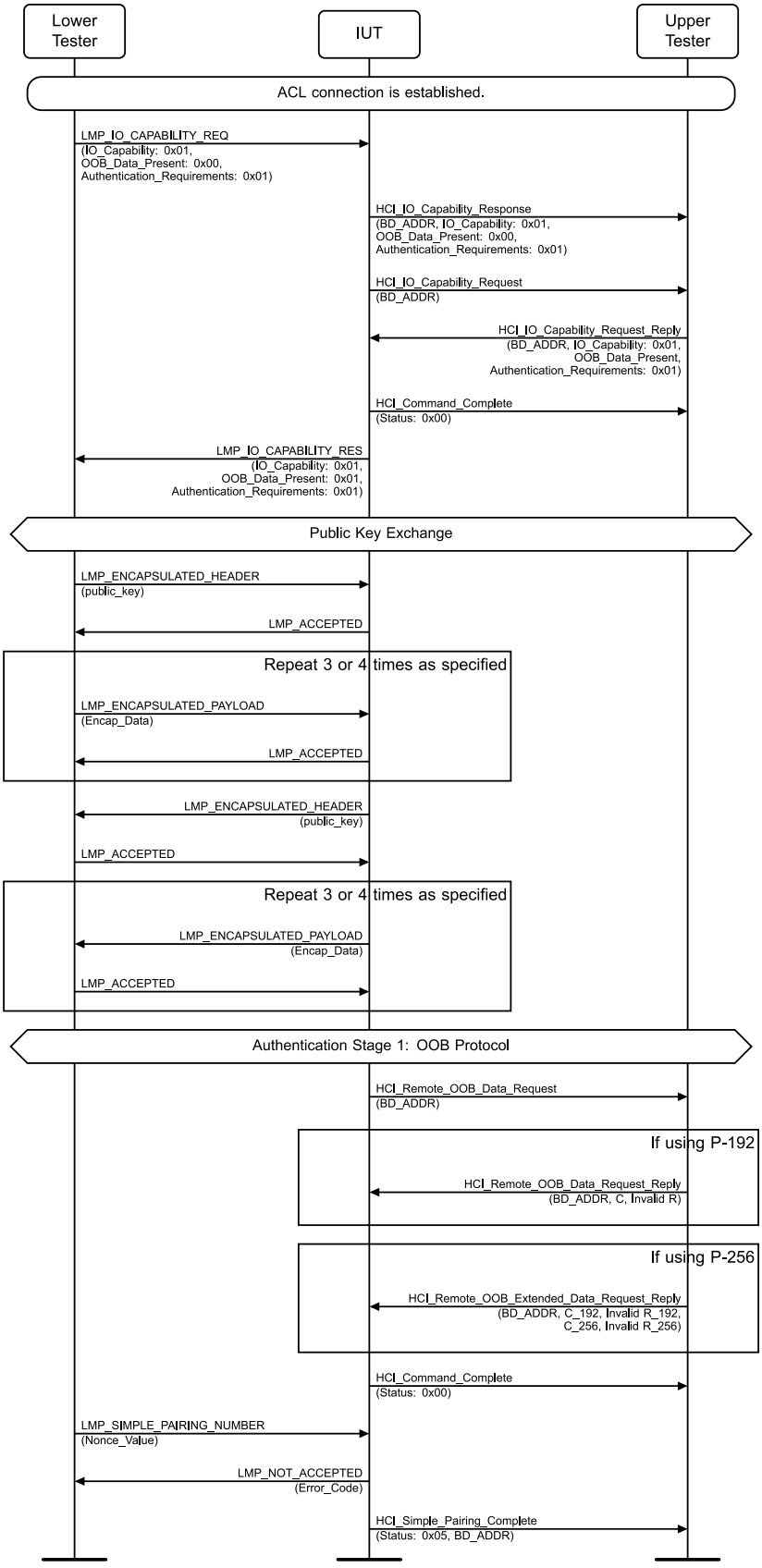


Figure 4.11-52: OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Failure MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x03 (P-192 and P-256 OOB authentication data from remote device present) if using P-256 as specified in Table 4.11-13, otherwise set to 0x01 (P-192 OOB authentication data from remote device present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times if using P-192 or four times if using P-256 as specified in Table 4.11-13.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times if using P-192 or four times if using P-256 as specified in Table 4.11-13.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.
15. The Upper Tester responds with an HCI_Remote_OOB_Data_Request_Reply command with the BD_ADDR, C, and an invalid R if using P-192 or an HCI_Remote_OOB_Extended_Data_Request_Reply command with the BD_ADDR, C_192, invalid R_192, C_256, and invalid R_256 if using P-256 as specified in Table 4.11-13 and receives an HCI_Command_Complete event in response.
16. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.

17. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.
18. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after sending the LMP_NOT_ACCEPTED PDU to the Lower Tester.

4.11.5.5 OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Failure

- Test Purpose

  Verify that the IUT supports the OOB protocol when the Lower Tester has OOB_Auth_Data that does not match the IUT.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Initiator.
  - The Upper Tester has sent the HCI_Read_Local_OOB_Data command to the IUT and received a successful HCI_Command_Complete event with the C and R if using P-192 or with C_192, R_192, C_256, and R_256 if using P-256 as specified in Table 4.11-14 in response before the ACL connection is established between the IUT and the Lower Tester.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-26-C [OOB Protocol - IUT Initiator – Lower Tester with OOB_Auth_Data – Failure]	P-192
LMP/SP/BV-62-C [OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Failure, P-256]	P-256

Table 4.11-14: OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Failure test cases

- Test Procedure

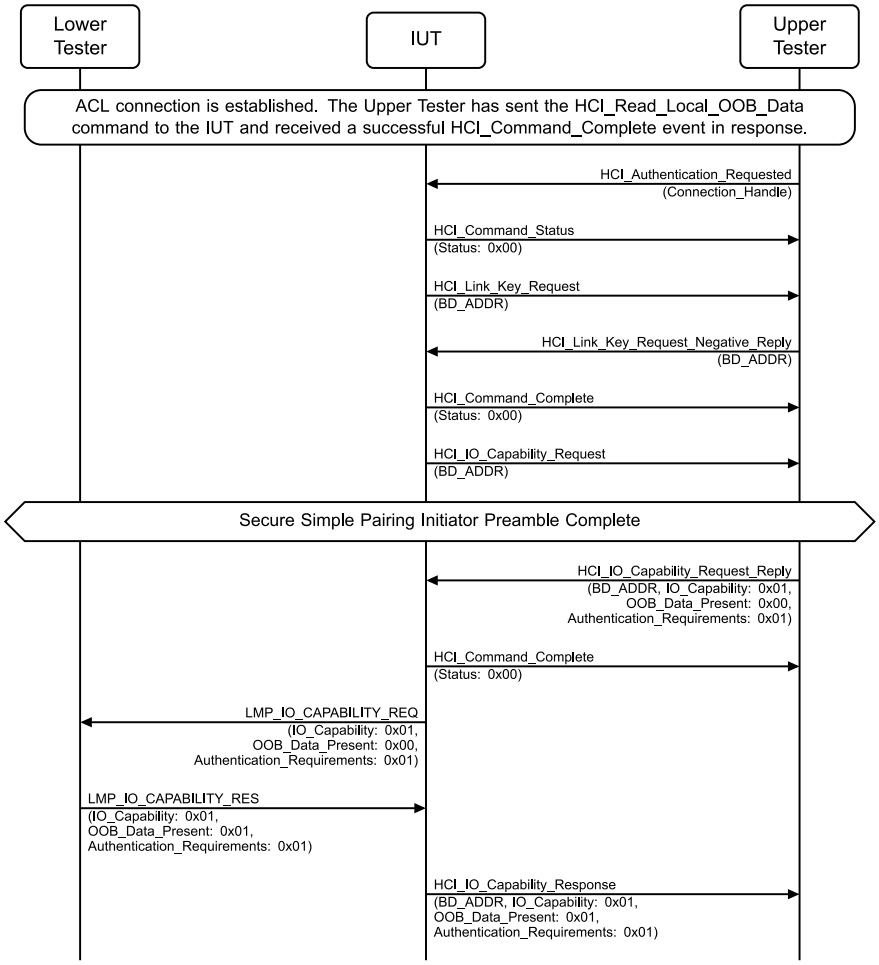


Figure 4.11-53: OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Failure MSC – Page 1 of 2

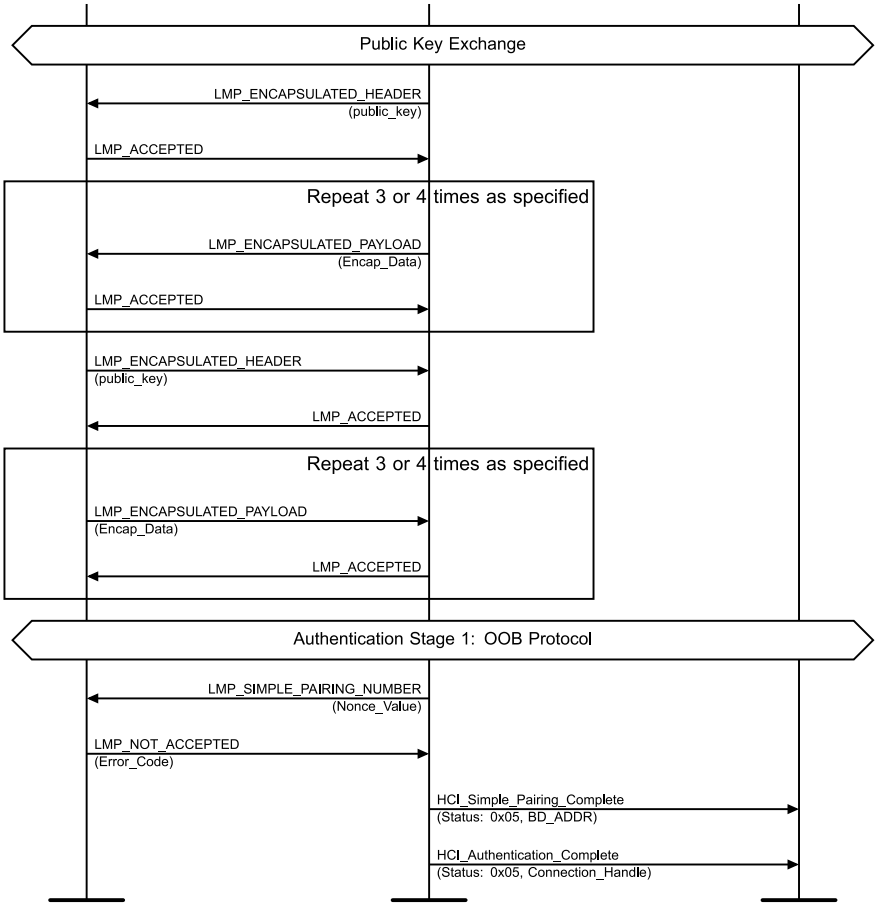


Figure 4.11-54: OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Failure MSC – Page 2 of 2

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x01 (P-192 OOB authentication data from remote device present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-14.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-14.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
18. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.
19. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  The IUT does not send an HCI_Remote_OOB_Data_Request event to the Upper Tester.

  The IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after receiving the LMP_NOT_ACCEPTED PDU from the Lower Tester.

4.11.5.6 OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Failure

- Test Purpose

  Verify that the IUT supports the OOB protocol when the Lower Tester has OOB_Auth_Data that does not match the IUT.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Responder.
  - The Upper Tester has sent the HCI_Read_Local_OOB_Data command to the IUT and received a successful HCI_Command_Complete event with the C and R if using P-192 or with C_192, R_192, C_256, and R_256 if using P-256 as specified in Table 4.11-15 in response before the ACL connection is established between the IUT and the Lower Tester.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	Elliptic Curve
LMP/SP/BV-27-C [OOB Protocol - IUT Responder – Lower Tester with OOB_Auth_Data – Failure]	P-192
LMP/SP/BV-63-C [OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Failure, P-256]	P-256

Table 4.11-15: OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Failure test cases

- Test Procedure

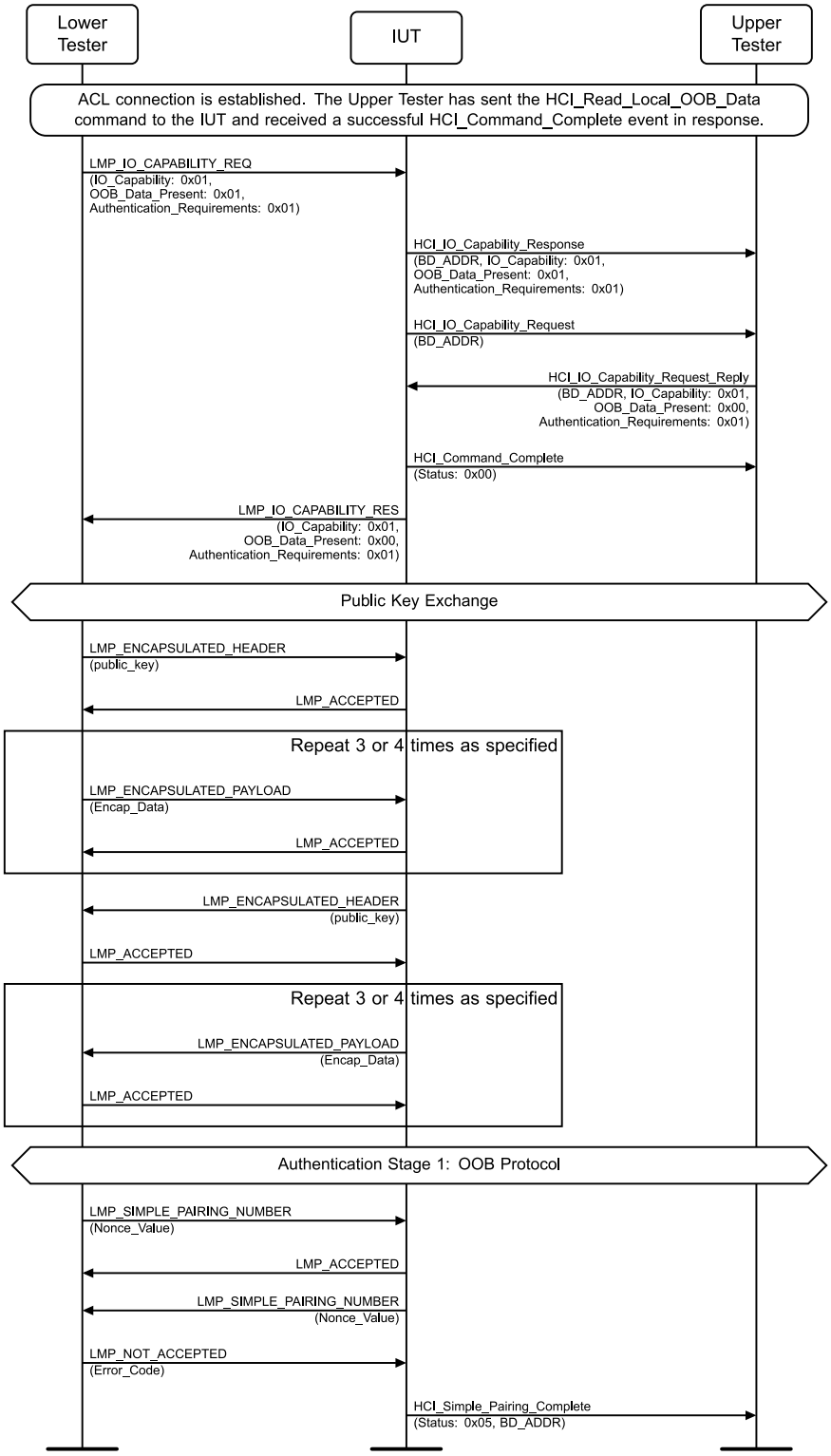


Figure 4.11-55: OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Failure MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x01 (P-192 OOB authentication data from remote device present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times if using P-192 or four times if using P-256 as specified in Table 4.11-15.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times if using P-192 or four times if using P-256 as specified in Table 4.11-15.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
15. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
16. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
17. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.
18. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT does not send an HCI_Remote_OOB_Data_Request event to the Upper Tester.

  The IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after receiving the LMP_NOT_ACCEPTED PDU from the Lower Tester.

4.11.5.7 OOB Protocol – IUT Initiator – Success, P-256

- Test Purpose

  Verify that the IUT supports the OOB protocol using the P-256 elliptic curve when the IUT or the Lower Tester has OOB_Auth_Data.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Initiator.
  - If the Lower Tester has OOB data as noted in Table 4.11-16, the Upper Tester has sent the HCI_Read_Local_OOB_Data command to the IUT and received a successful HCI_Command_Complete event with the C and R in response before the ACL connection is established between the IUT and the Lower Tester.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	IUT OOB	Lower Tester OOB	Random Key
LMP/SP/BV-54-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Success, P-256, v6.2 and earlier]	Yes	No	May be repeated
LMP/SP/BV-56-C [OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier]	No	Yes	May be repeated
LMP/SP/BV-58-C [OOB Protocol – IUT Initiator – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier]	Yes	Yes	May be repeated
LMP/SP/BV-76-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Success, P-256, v6.3 and later]	Yes	No	Unique
LMP/SP/BV-77-C [OOB Protocol – IUT Initiator – Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later]	No	Yes	Unique
LMP/SP/BV-78-C [OOB Protocol – IUT Initiator – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later]	Yes	Yes	Unique

Table 4.11-16: OOB Protocol – IUT Initiator – Success, P-256 test cases

- Test Procedure

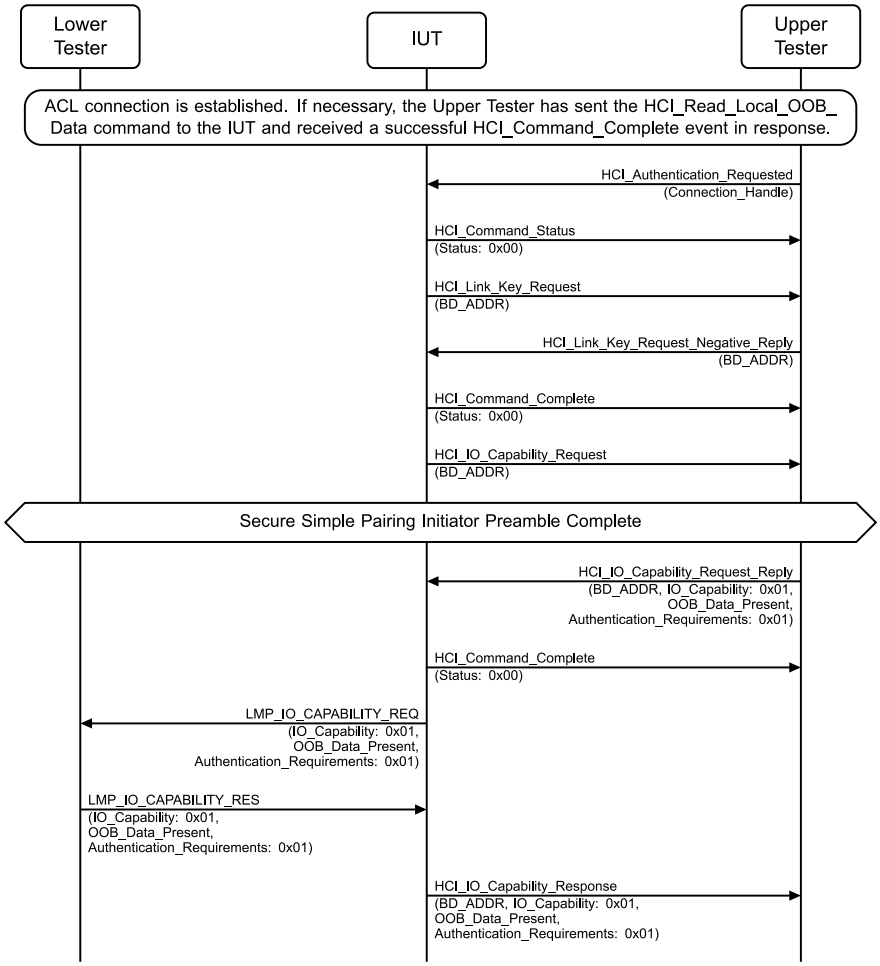


Figure 4.11-56: OOB Protocol – IUT Initiator – Success, P-256 MSC – Page 1 of 4

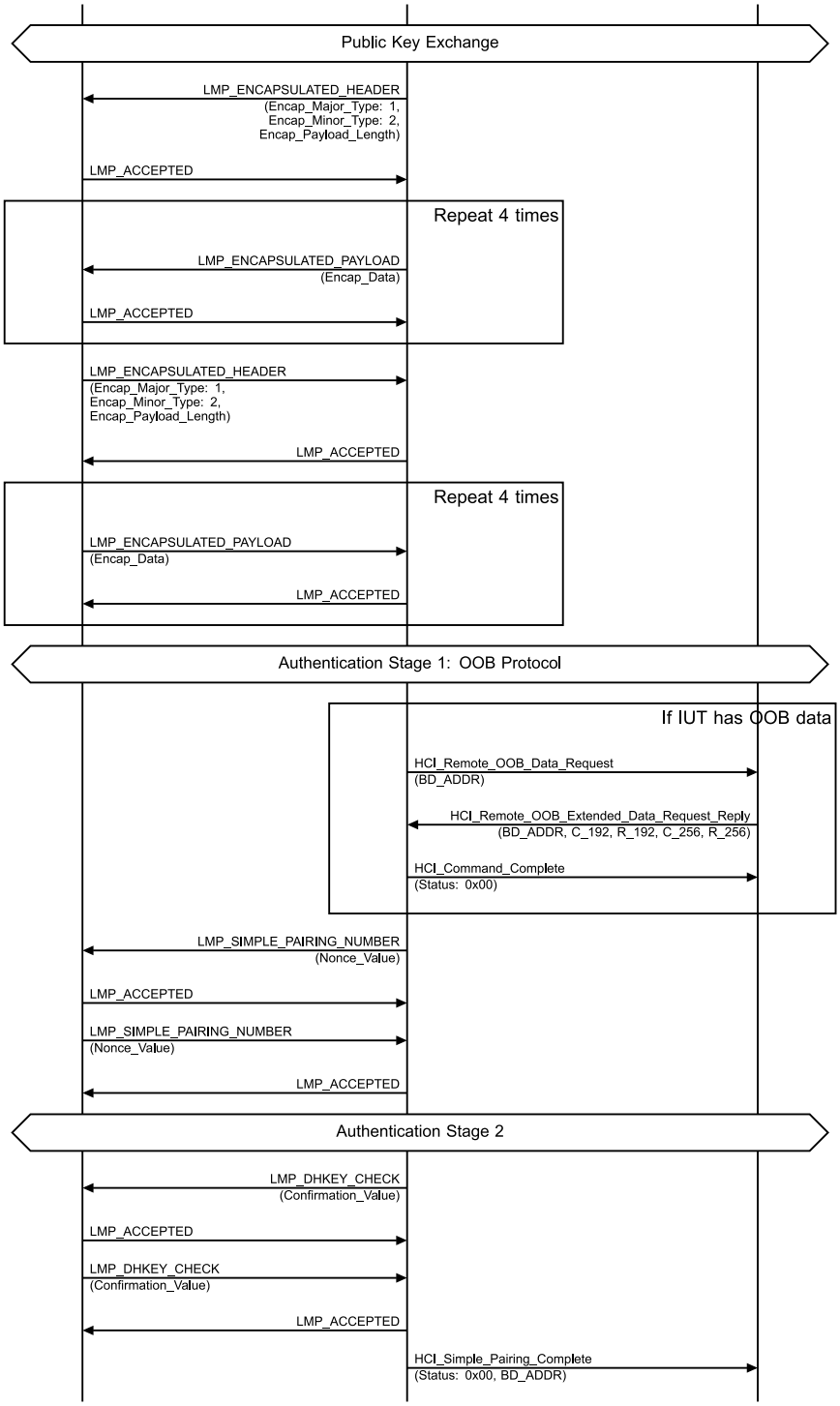


Figure 4.11-57: OOB Protocol – IUT Initiator – Success, P-256 MSC – Page 2 of 4

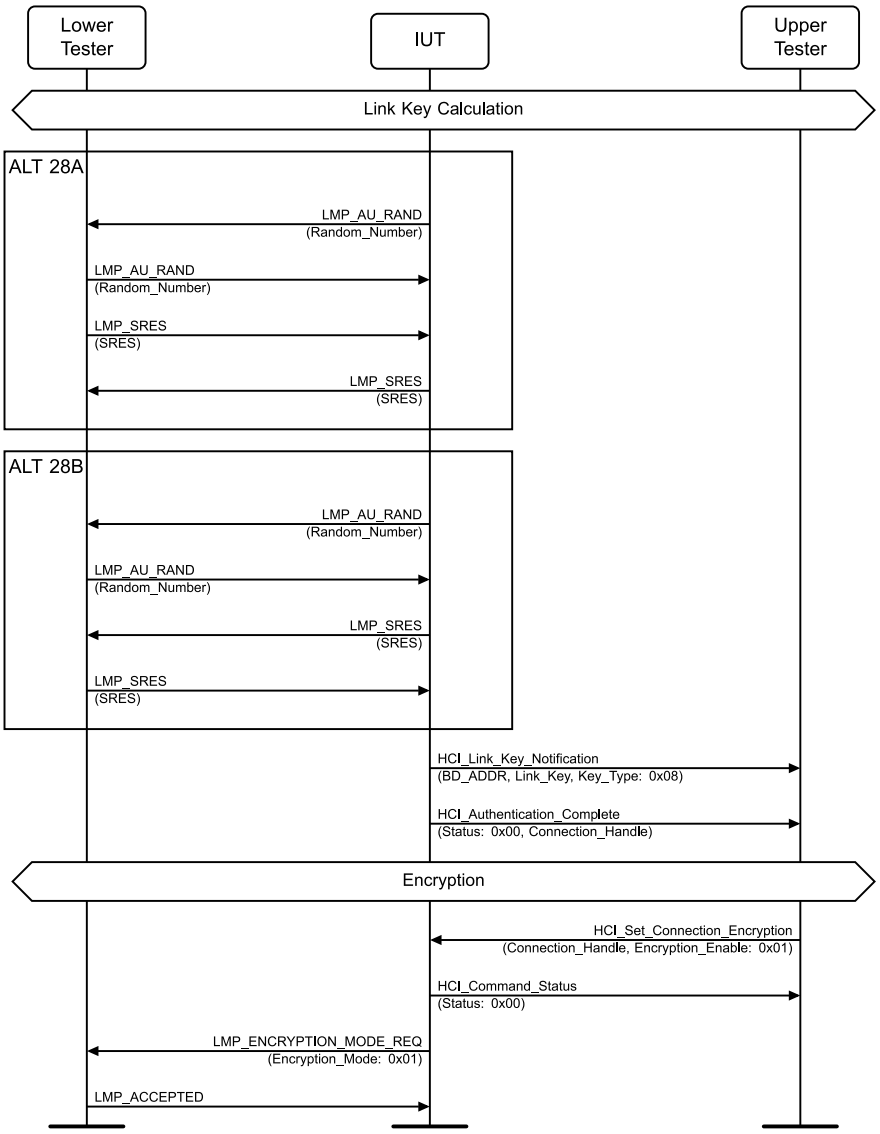


Figure 4.11-58: OOB Protocol – IUT Initiator – Success, P-256 MSC – Page 3 of 4

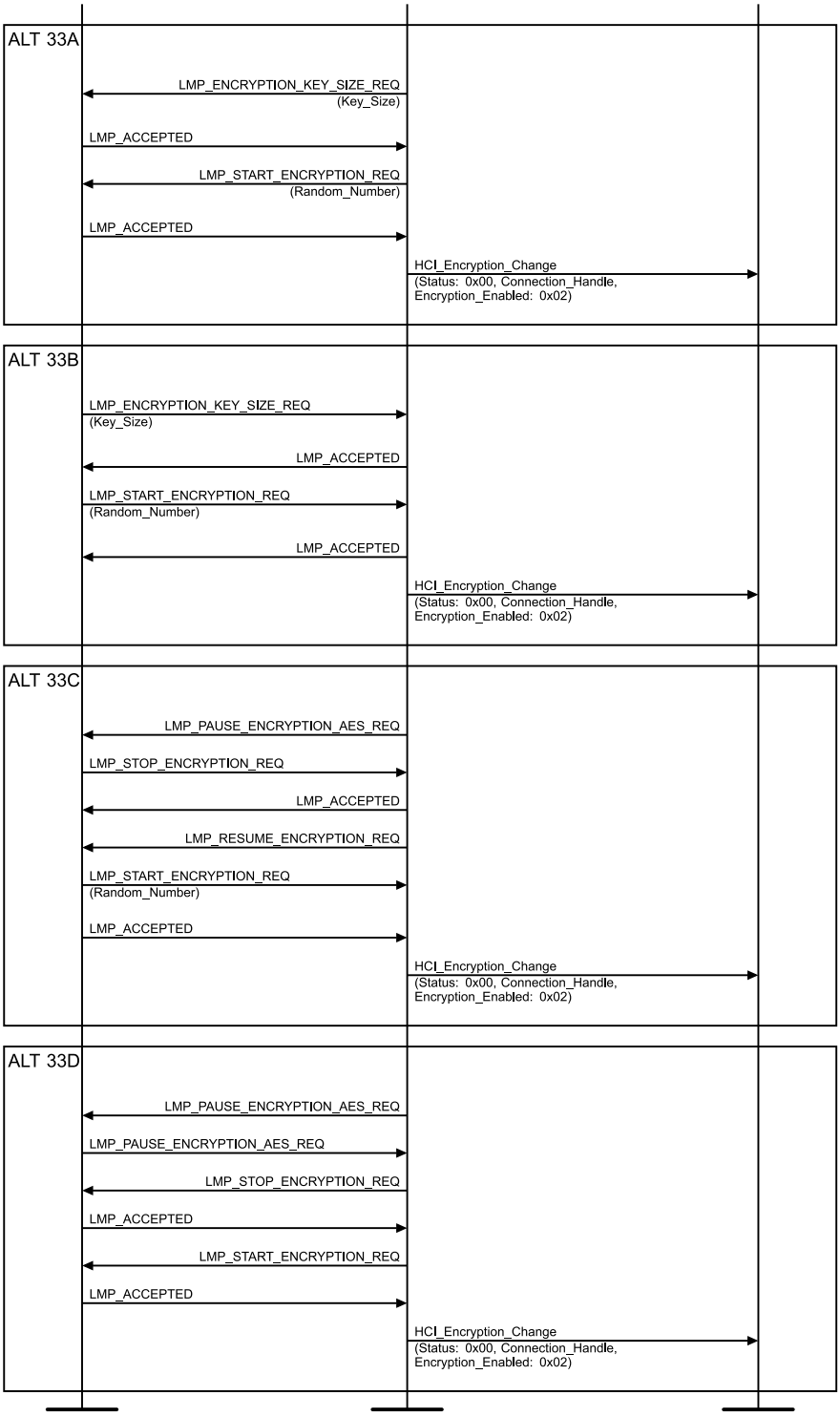


Figure 4.11-59: OOB Protocol – IUT Initiator – Success, P-256 MSC – Page 4 of 4

Repeat the Test Procedure five times.

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.

3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x03 (P-192 and P-256 OOB authentication data from remote device present) if the IUT has OOB data specified in Table 4.11-16, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the IUT has OOB data specified in Table 4.11-16, otherwise set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the Lower Tester has OOB data specified in Table 4.11-16, otherwise set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x03 (P-192 and P-256 OOB authentication data from remote device present) if the Lower Tester has OOB data specified in Table 4.11-16, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 four times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 four times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.

Execute Steps 17–18 only if the IUT has OOB data specified in Table 4.11-16.

17. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.

18. The Upper Tester responds with an HCI_Remote_OOB_Extended_Data_Request_Reply command with the BD_ADDR, C_192, R_192, C_256, and R_256 and receives an HCI_Command_Complete event in response.
19. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
20. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
21. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
22. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
23. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
24. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
25. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
26. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
27. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
28. Perform either alternative 28A or 28B depending on the IUT's role.
    Alternative 28A (The IUT is the Central):
    - 28A.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
    - 28A.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
    - 28A.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.
    - 28A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES.

    Alternative 28B (The IUT is the Peripheral):
    - 28B.1. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
    - 28B.2. The Lower Tester responds to the IUT with an LMP_AU_RAND PDU with a Random_Number.
    - 28B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
    - 28B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES.
29. The IUT sends an HCI_Link_Key_Notification event to the Upper Tester with the BD_ADDR, Link_Key, and Key_Type set to 0x08 (Authenticated Combination Key generated from P-256) and a successful HCI_Authentication_Complete event with the Connection_Handle.
30. On the first procedure iteration, the Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response; otherwise, skip Steps 31 and 32 and perform alternative 33C or 33D depending on the IUT's role.
31. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
32. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
33. Perform either alternative 33A or 33B depending on the IUT's role.
    Alternative 33A (The IUT is the Central):
    - 33A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
    - 33A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

33A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
33A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
33A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

Alternative 33B (The IUT is the Peripheral):

33B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.
33B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
33B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
33B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
33B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

Alternative 33C (The IUT is the Peripheral, AES Encryption):

33C.1. The IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester.
33C.2. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
33C.3. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
33C.4. The IUT sends an LMP_RESUME_ENCRYPTION_REQ PDU to the Lower Tester.
33C.5. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
33C.6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
33C.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

Alternative 33D (The IUT is the Central, AES Encryption):

33D.1. The IUT sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the Lower Tester.
33D.2. The Lower Tester sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the IUT.
33D.3. The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
33D.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
33D.5. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
33D.6. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
33D.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

34. The Upper Tester sends an HCI_Delete_Stored_Link_Key command to the IUT with BD_ADDR set to the Lower Tester BD_ADDR and Delete_All set to 0x00 and receives a successful HCI_Command_Complete event in response.

- Expected Outcome

  Pass verdict

  If the IUT has OOB data specified in Table 4.11-16: In Step 17, the IUT sends the HCI_Remote_OOB_Data_Request event to the Upper Tester.

If the IUT does not have OOB data specified in Table 4.11-16: The IUT does not send the HCI_Remote_OOB_Data_Request event to the Upper Tester.

In Step 23, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

In Step 27, the IUT sends the successful HCI_Simple_Pairing_Complete event to the Upper Tester.

If specified in Table 4.11-16, the random number sent by the IUT in Step 28A.2 or 28B.2 is always unique.

In Step 29, the IUT sends the resulting Link_Key and Key_Type to the Upper Tester in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x08 (Authenticated Combination Key generated from P-256).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to an LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

4.11.5.8 OOB Protocol – IUT Responder – Success, P-256

- Test Purpose

  Verify that the IUT supports the OOB protocol using the P-256 elliptic curve when the IUT or the Lower Tester has OOB_Auth_Data.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Responder.
  - If the Lower Tester has OOB data noted in Table 4.11-17, the Upper Tester has sent the HCI_Read_Local_OOB_Data command to the IUT and received a successful HCI_Command_Complete event with the C and R in response before the ACL connection is established between the IUT and the Lower Tester.
  - An ACL connection has been established.

- Test Case Configuration

Test Case	IUT OOB	Lower Tester OOB	Random Key
LMP/SP/BV-55-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Success, P-256, v6.2 and earlier]	Yes	No	May be repeated
LMP/SP/BV-57-C [OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier]	No	Yes	May be repeated
LMP/SP/BV-59-C [OOB Protocol – IUT Responder – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.2 and earlier]	Yes	Yes	May be repeated

Test Case	IUT OOB	Lower Tester OOB	Random Key
LMP/SP/BV-79-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Success, P-256, v6.3 and later]	Yes	No	Unique
LMP/SP/BV-80-C [OOB Protocol – IUT Responder – Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later]	No	Yes	Unique
LMP/SP/BV-81-C [OOB Protocol – IUT Responder – IUT and Lower Tester with OOB_Auth_Data – Success, P-256, v6.3 and later]	Yes	Yes	Unique

Table 4.11-17: OOB Protocol – IUT Responder – Success, P-256 test cases

- Test Procedure

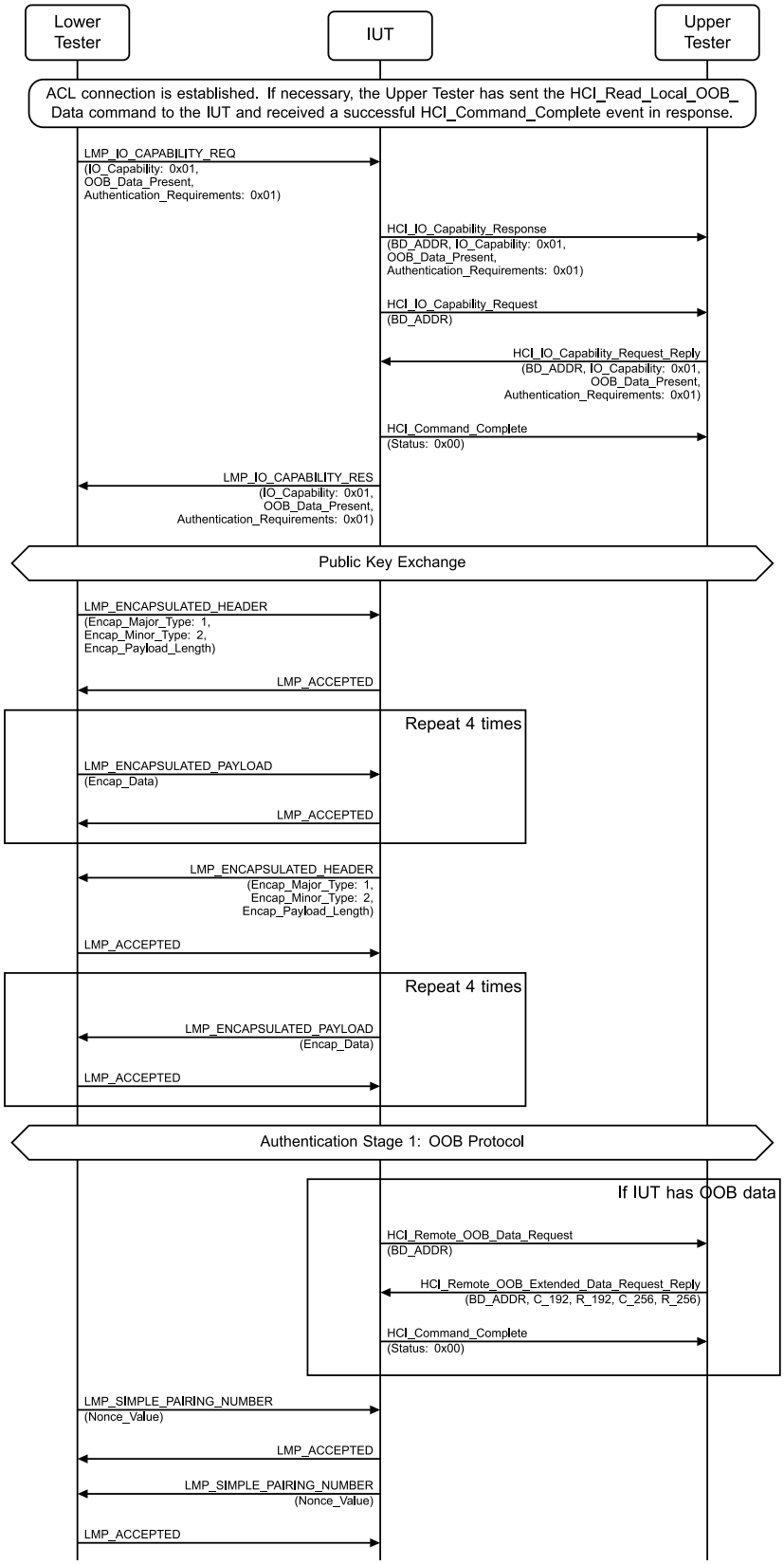


Figure 4.11-60: OOB Protocol – IUT Responder – Success, P-256 MSC – Page 1 of 3

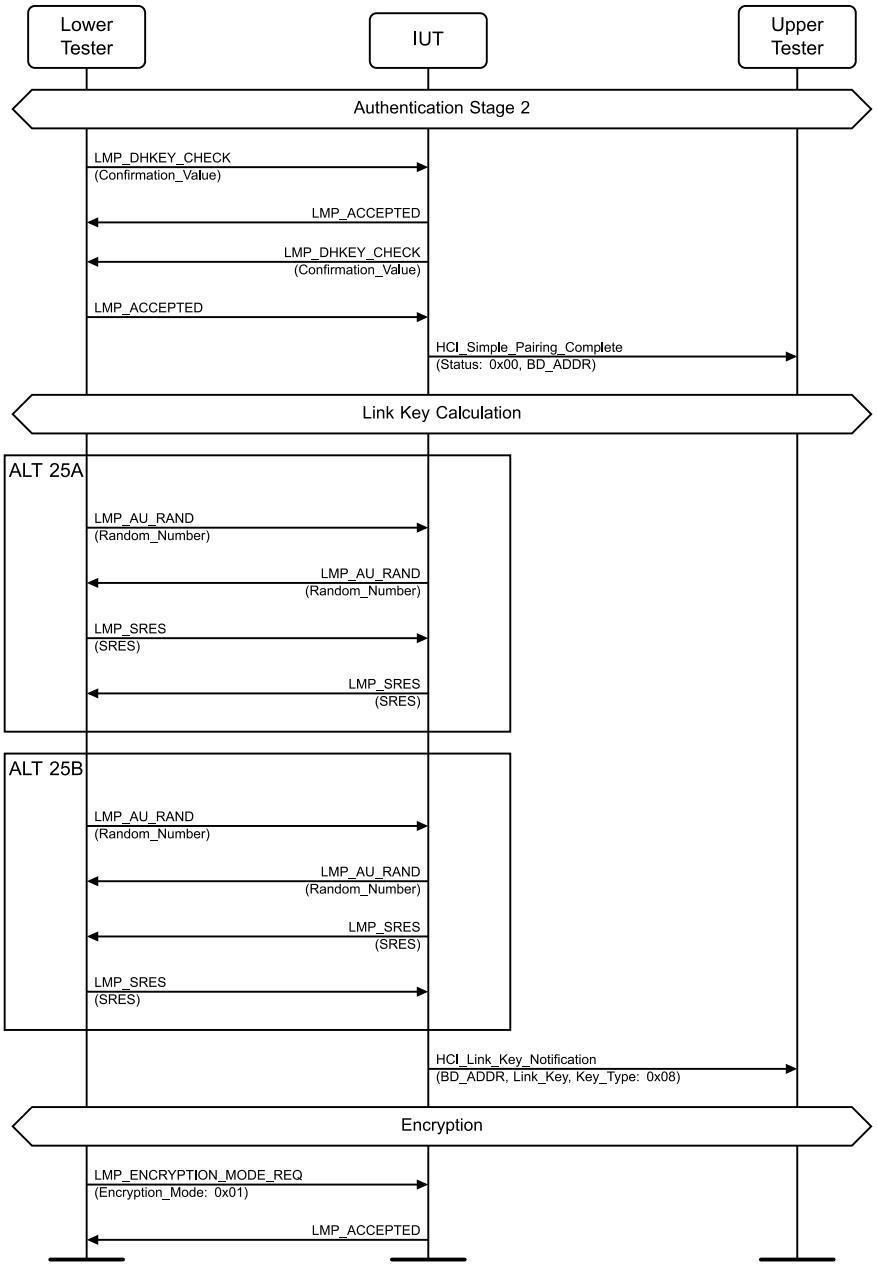


Figure 4.11-61: OOB Protocol – IUT Responder – Success, P-256 MSC – Page 2 of 3

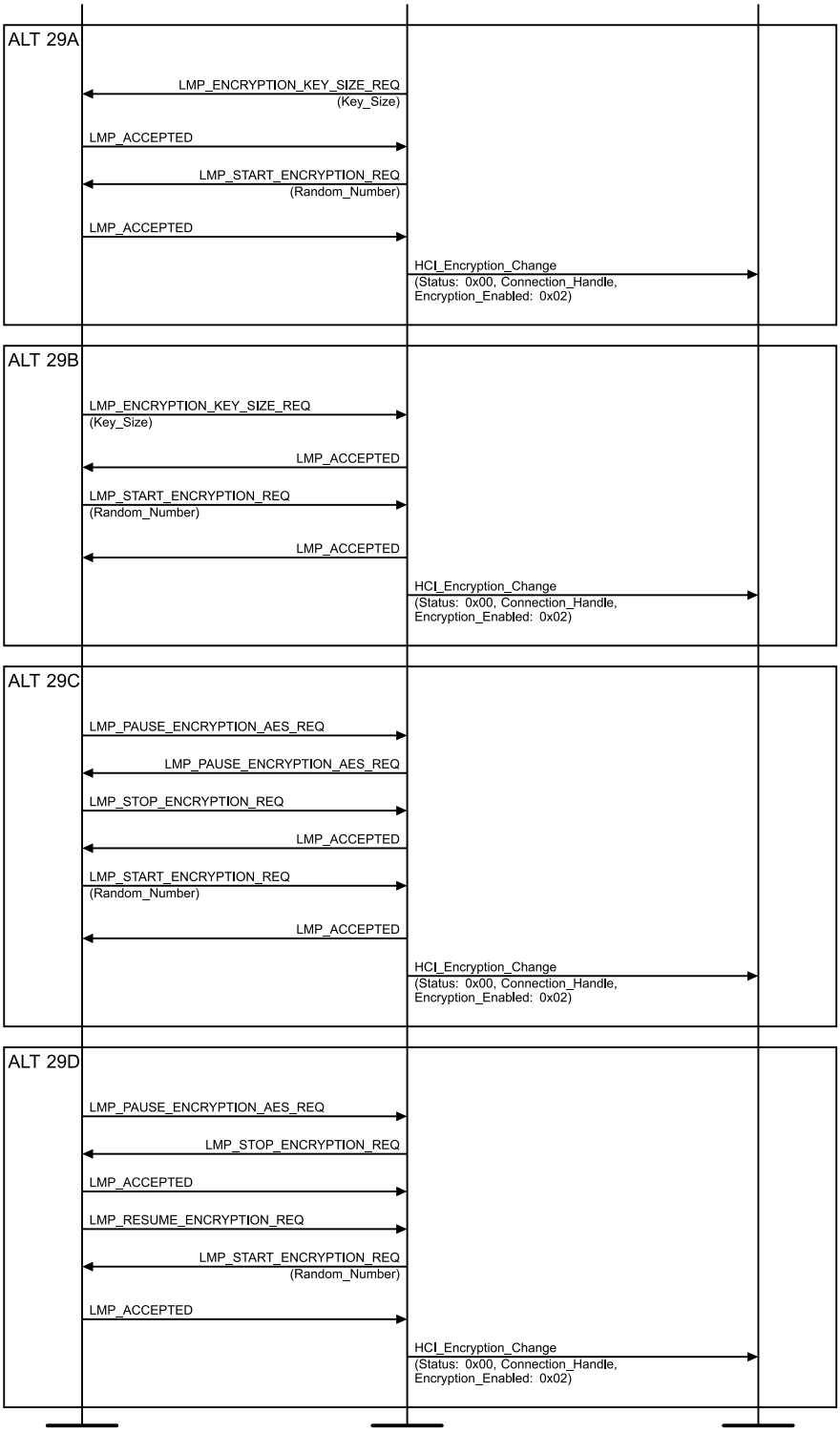


Figure 4.11-62: OOB Protocol – IUT Responder – Success, P-256 MSC – Page 3 of 3

Repeat the Test Procedure five times.

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the Lower Tester has OOB data specified in Table 4.11-17, otherwise set to 0x00 (No OOB

Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x03 (P-192 and P-256 OOB authentication data from remote device present) if the Lower Tester has OOB data specified in Table 4.11-17, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x03 (P-192 and P-256 OOB authentication data from remote device present) if the IUT has OOB data specified in Table 4.11-17, otherwise set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received) if the IUT has OOB data specified in Table 4.11-17, otherwise set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 four times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with Encap_Major_Type set to 1, Encap_Minor_Type set to 2, and Encap_Payload_Length.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 four times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.

Execute Steps 14–15 only if the IUT has OOB data specified in Table 4.11-17.

14. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.
15. The Upper Tester responds with an HCI_Remote_OOB_Extended_Data_Request_Reply command with the BD_ADDR, C_192, R_192, C_256, and R_256 and receives an HCI_Command_Complete event in response.
16. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.

18. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
19. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
22. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
23. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
24. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
25. Perform either alternative 25A or 25B depending on the IUT's role.

    Alternative 25A (The IUT is the Central):

    25A.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.

    25A.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.

    25A.3. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.

    25A.4. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES.

    Alternative 25B (The IUT is the Peripheral):

    25B.1. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.

    25B.2. The IUT responds to the Lower Tester with an LMP_AU_RAND PDU with a Random_Number.

    25B.3. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.

    25B.4. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES.
26. The IUT sends an HCI_Link_Key_Notification event with the BD_ADDR, Link_Key, and Key_Type set to 0x08 (Authenticated Combination Key generated from P-256).
27. On the first procedure iteration, the Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption); otherwise, skip Step 28 and perform alternative 29C or 29D depending on the IUT's role.
28. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
29. Perform either alternative 29A or 29B depending on the IUT's role.

    Alternative 29A (The IUT is the Central):

    29A.1. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.

    29A.2. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

    29A.3. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.

    29A.4. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.

    29A.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

    Alternative 29B (The IUT is the Peripheral):

    29B.1. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.

    29B.2. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.

29B.3. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
29B.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
29B.5. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

Alternative 29C (The IUT is the Peripheral, AES Encryption):

29C.1. The Lower Tester sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the IUT.
29C.2. The IUT sends an LMP_PAUSE_ENCRYPTION_REQ PDU to the Lower Tester.
29C.3. The Lower Tester sends an LMP_STOP_ENCRYPTION_REQ PDU to the IUT.
29C.4. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
29C.5. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
29C.6. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
29C.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

Alternative 29D (The IUT is the Central, AES Encryption):

29D.1. The Lower Tester sends an LMP_PAUSE_ENCRYPTION_AES_REQ PDU to the IUT.
29D.2 The IUT sends an LMP_STOP_ENCRYPTION_REQ PDU to the Lower Tester.
29D.3. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_STOP_ENCRYPTION_REQ PDU Opcode.
29D.4. The Lower Tester sends an LMP_RESUME_ENCRYPTION_REQ PDU to the IUT.
29D.5. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester.
29D.6. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
29D.7. The IUT sends a successful HCI_Encryption_Key_Refresh_Complete event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x02.

30. The Upper Tester sends an HCI_Delete_Stored_Link_Key command to the IUT with BD_ADDR set to the Lower Tester BD_ADDR and Delete_All set to 0x00 and receives a successful HCI_Command_Complete event in response.

- Expected Outcome

  Pass verdict

  If the IUT has OOB data specified in Table 4.11-17: In Step 14, the IUT sends the HCI_Remote_OOB_Data_Request event to the Upper Tester.

  If the IUT does not have OOB data specified in Table 4.11-17: The IUT does not send the HCI_Remote_OOB_Data_Request event to the Upper Tester.

  In Step 22, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 24, the IUT sends the successful HCI_Simple_Pairing_Complete event to the Upper Tester.

  If specified in Table 4.11-17, the random number sent by the IUT in Step 25A.2 or 25B.2 is always unique.

In Step 26, the IUT sends the resulting Link Key and Key Type to the Upper Tester in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x08 (Authenticated Combination Key generated from P-256).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to an LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

4.11.6 Test Mode procedures

Note: In all the test cases in Section 4.11.6 Test Mode procedures, it does not matter whether the IUT is the Central or the Peripheral.

LMP/SP/BV-28-C [Secure Simple Pairing Debug Mode - Fixed Private Key]

- Test Purpose

  Verify that the IUT supports the Secure Simple Pairing debug mode when the fixed private/public key pair is used.

- Reference

  [1] 4.2.7

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

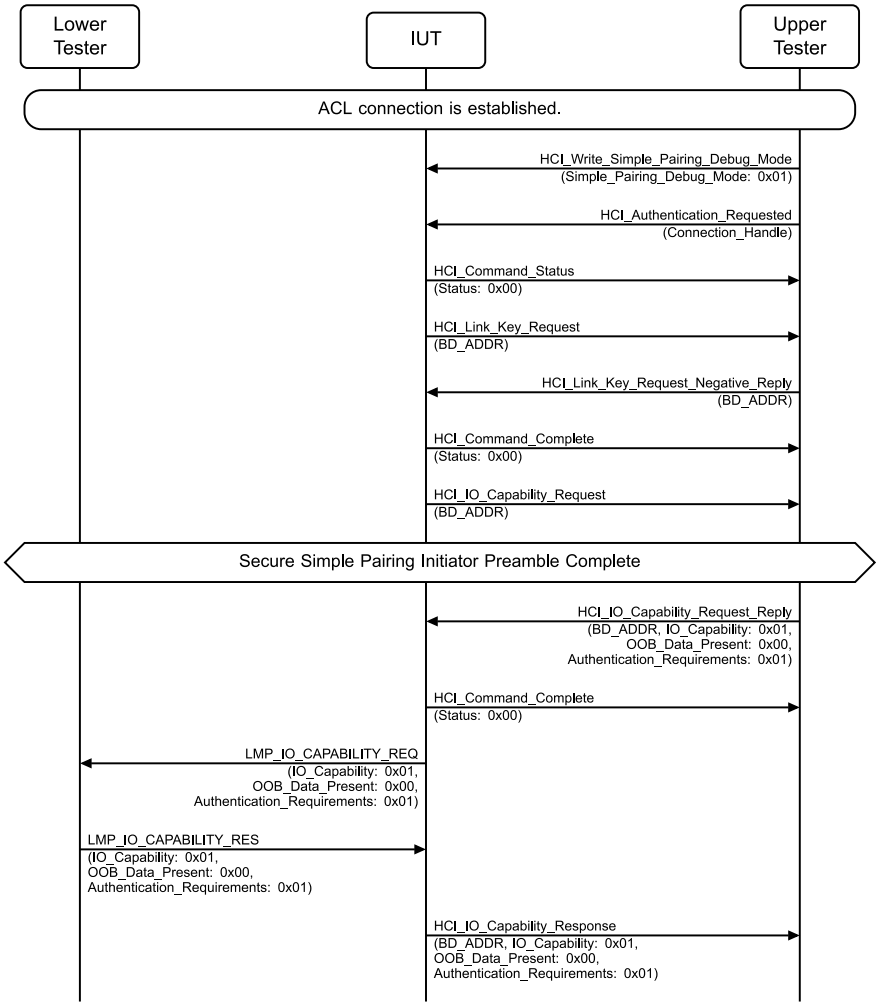


Figure 4.11-63: LMP/SP/BV-28-C [Secure Simple Pairing Debug Mode - Fixed Private Key] MSC – Page 1 of 3

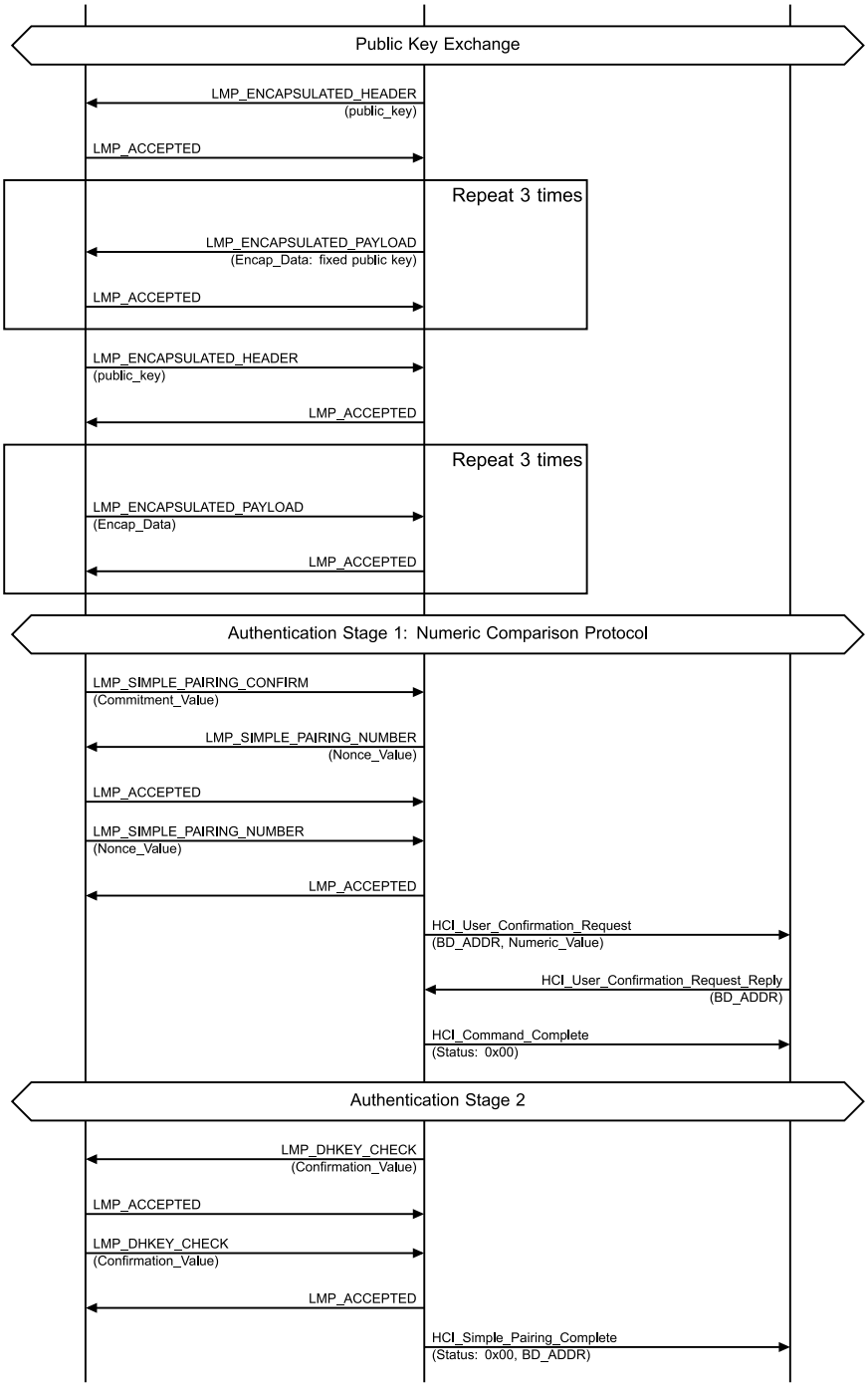


Figure 4.11-64: LMP/SP/BV-28-C [Secure Simple Pairing Debug Mode - Fixed Private Key] MSC – Page 2 of 3

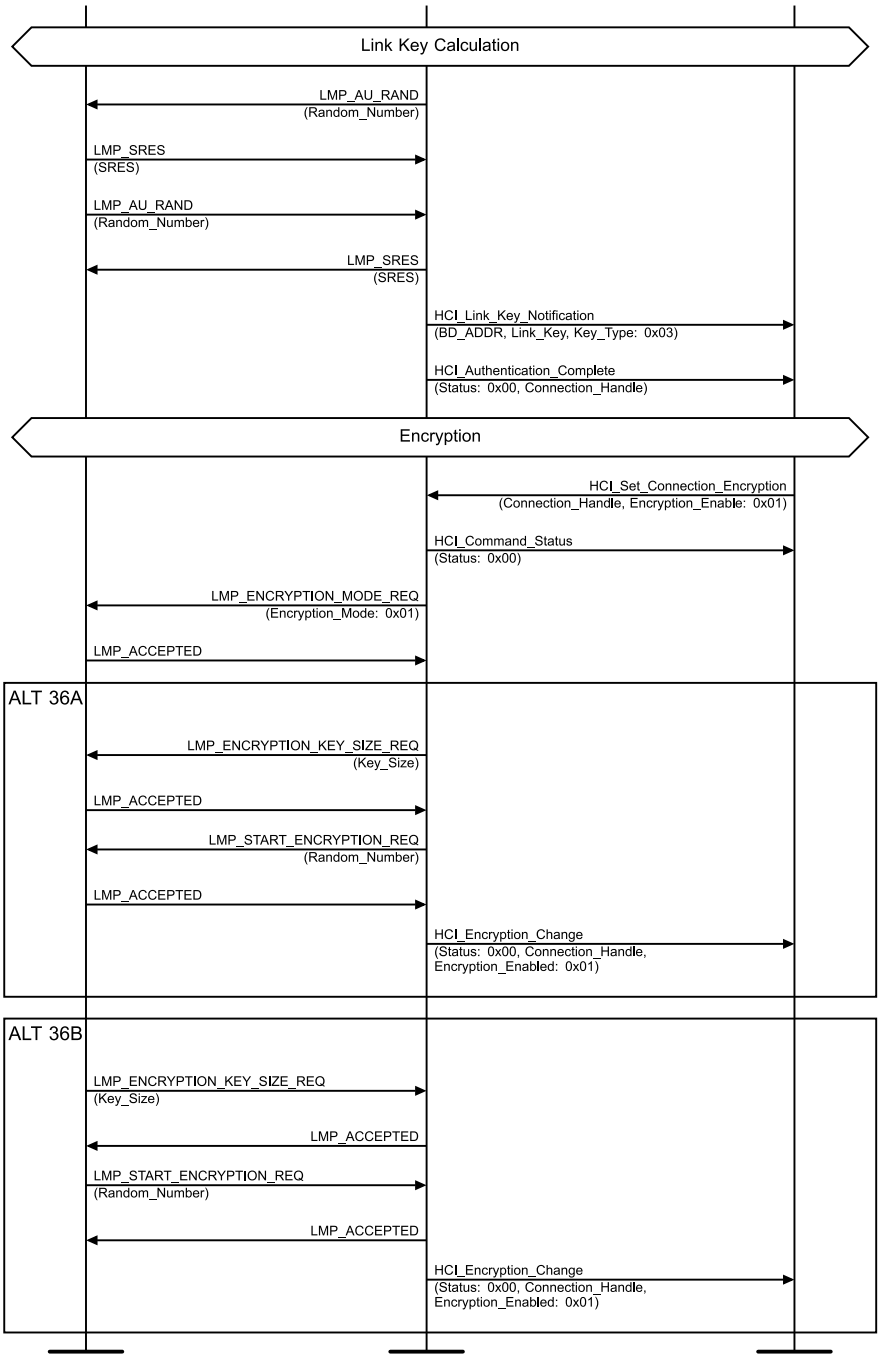


Figure 4.11-65: LMP/SP/BV-28-C [Secure Simple Pairing Debug Mode - Fixed Private Key] MSC – Page 3 of 3

1. The Upper Tester sends an HCI_Write_Simple_Pairing_Debug_Mode command to the IUT with Simple_Pairing_Debug_Mode set to 0x01 (Enabled).
2. Execute the test procedure steps of LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success] once, except that in Step 11 the IUT sends the fixed public key, and in Step 32 the Key_Type is set to 0x03 (Debug Combination Key).

- Expected Outcome

  Pass verdict

  The IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x03 (Debug Combination Key).

LMP/SP/BV-29-C [Secure Simple Pairing Debug Mode - Responding Device Uses Fixed Private Key]

- Test Purpose

  Verify that the IUT reports the Key_Type as Debug Combination Key when the remote device uses the Secure Simple Pairing debug mode with a fixed private/public key pair.

- Reference

  [1] 4.2.7

- Initial Condition
  - See the “Default settings” section.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

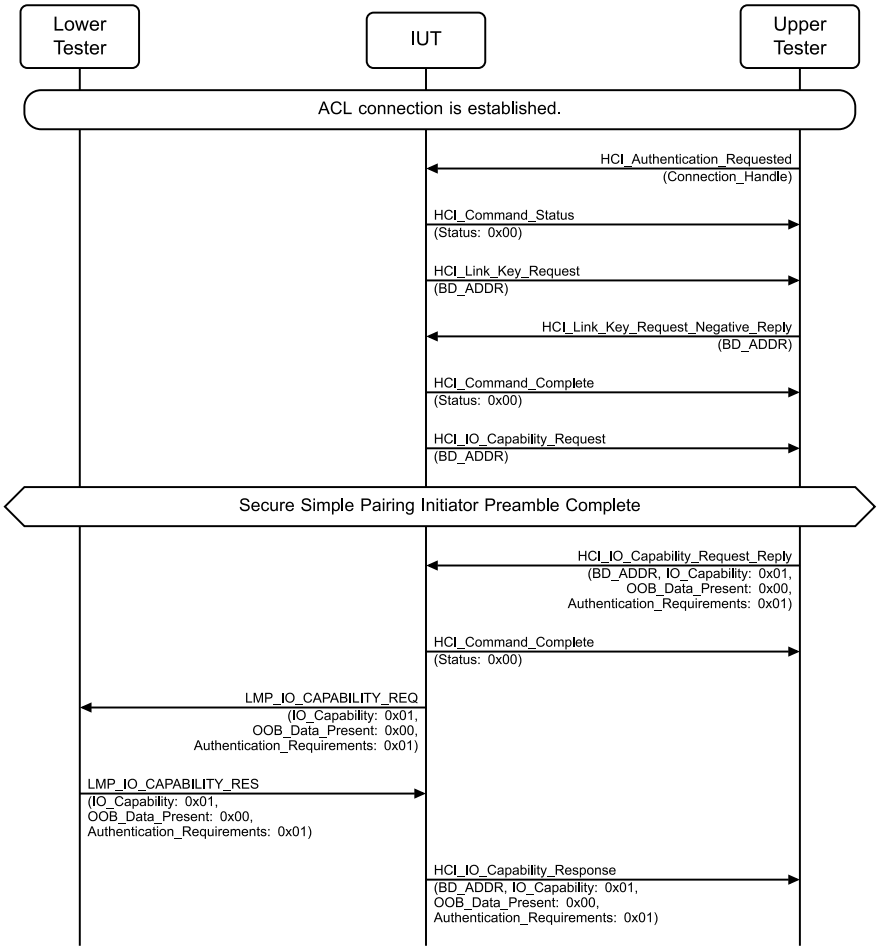


Figure 4.11-66: LMP/SP/BV-29-C [Secure Simple Pairing Debug Mode - Responding Device Uses Fixed Private Key] MSC – Page 1 of 3

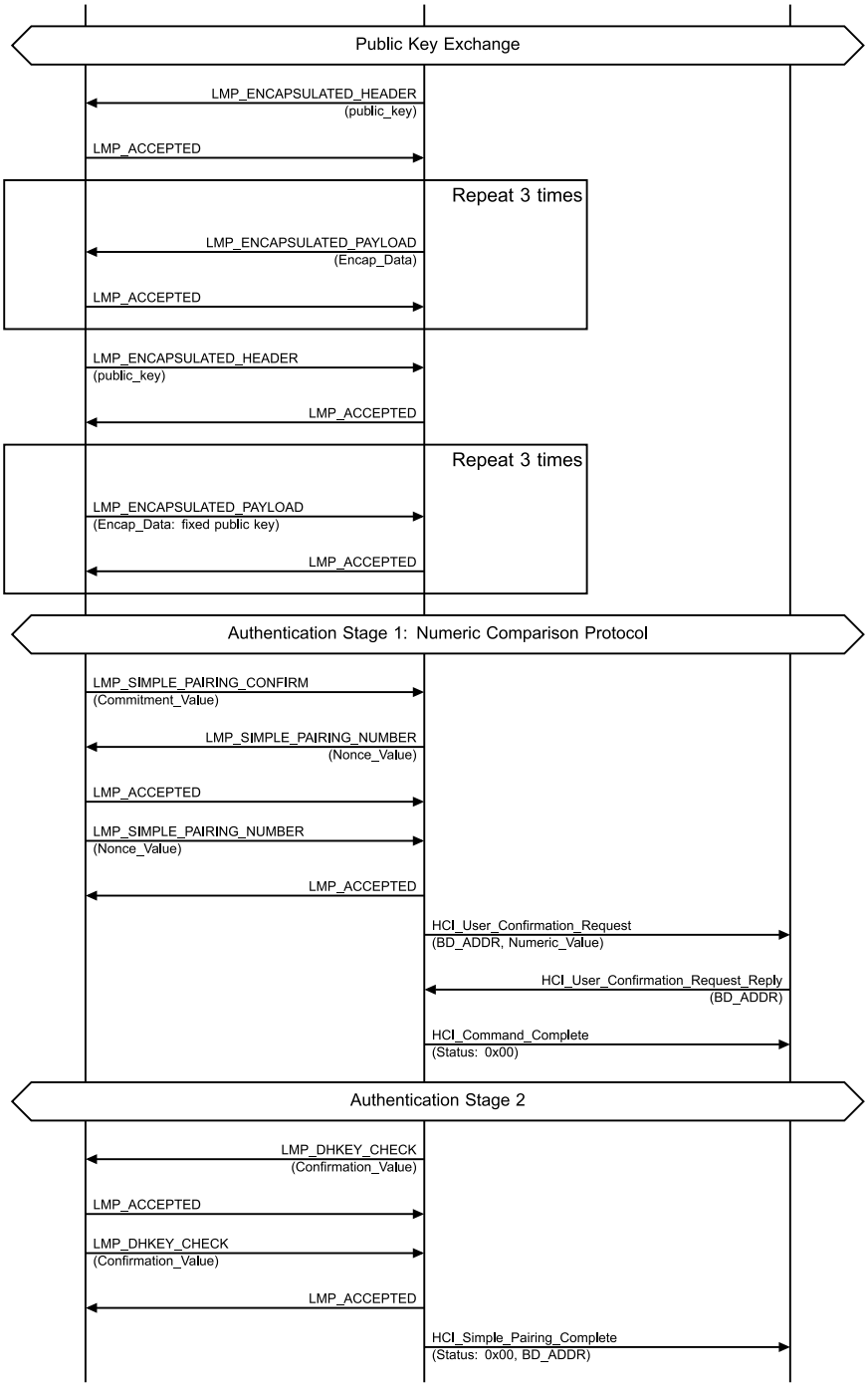


Figure 4.11-67: LMP/SP/BV-29-C [Secure Simple Pairing Debug Mode - Responding Device Uses Fixed Private Key] MSC – Page 2 of 3

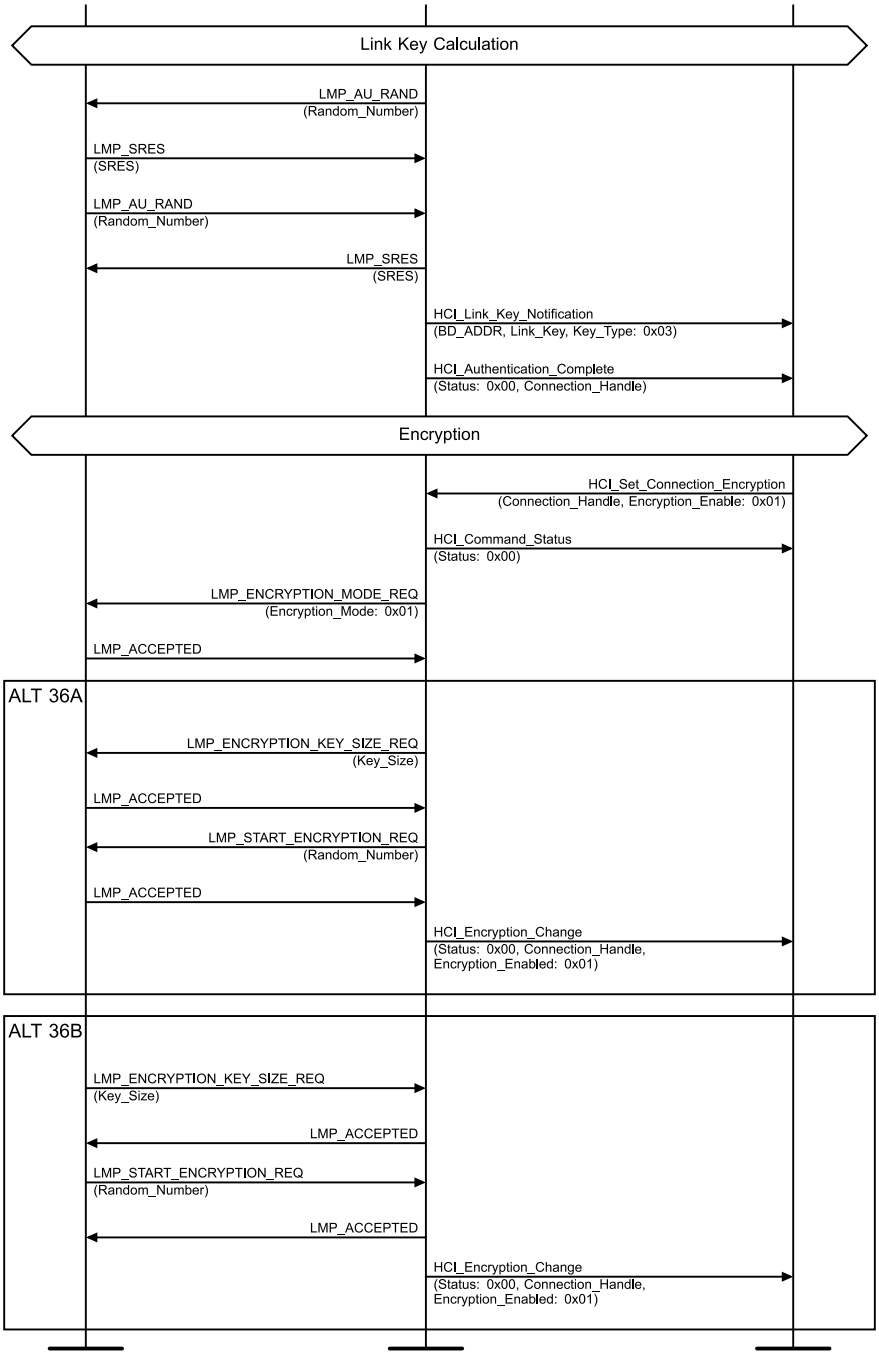


Figure 4.11-68: LMP/SP/BV-29-C [Secure Simple Pairing Debug Mode - Responding Device Uses Fixed Private Key] MSC – Page 3 of 3

1. Execute the test procedure steps of LMP/SP/BV-06-C [Numeric Comparison - IUT Initiator – Success] once, except that in Step 15 the Lower Tester sends the fixed public key, and in Step 32 the Key_Type is set to 0x03 (Debug Combination Key).

- Expected Outcome

  Pass verdict

  The IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x03 (Debug Combination Key).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-64-C [Secure Simple Pairing Debug Mode – Fixed Private Key, P256]

- Test Purpose

  Verify that the IUT supports the Secure Simple Pairing debug mode when the fixed P-256 private/public key pair is used and reports the Key_Type as Debug Combination Key.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

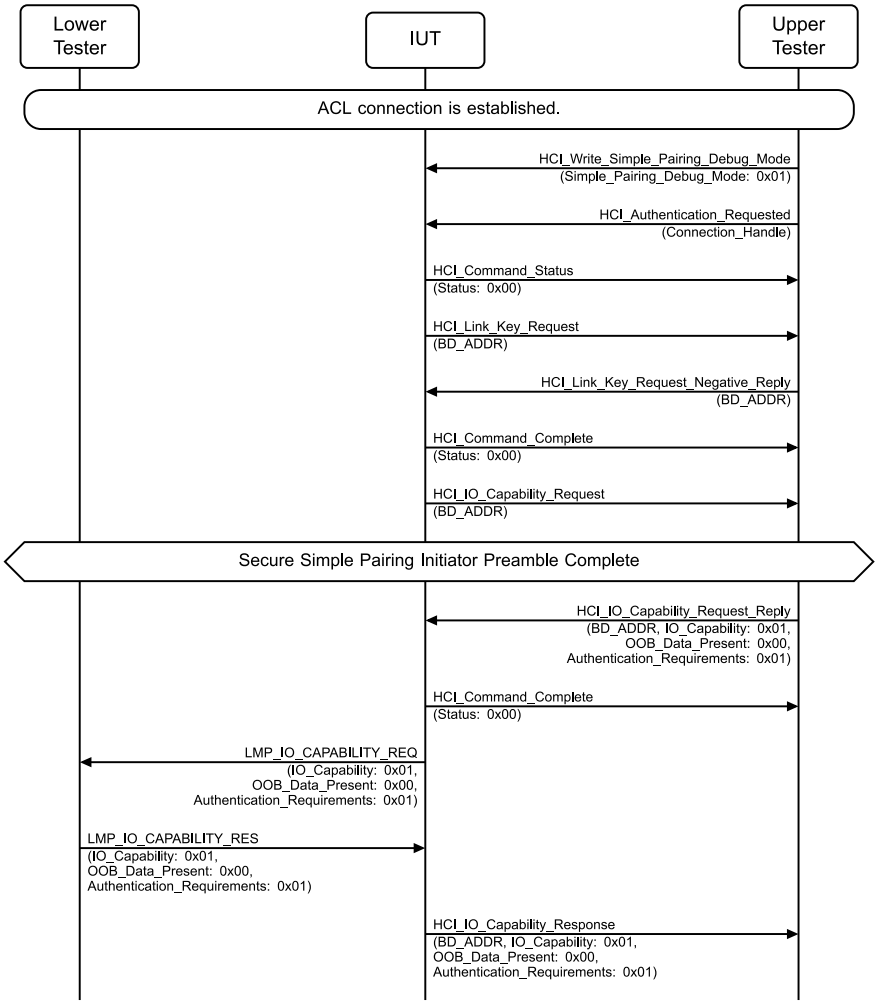


Figure 4.11-69: LMP/SP/BV-64-C [Secure Simple Pairing Debug Mode – Fixed Private Key, P256] MSC – Page 1 of 3

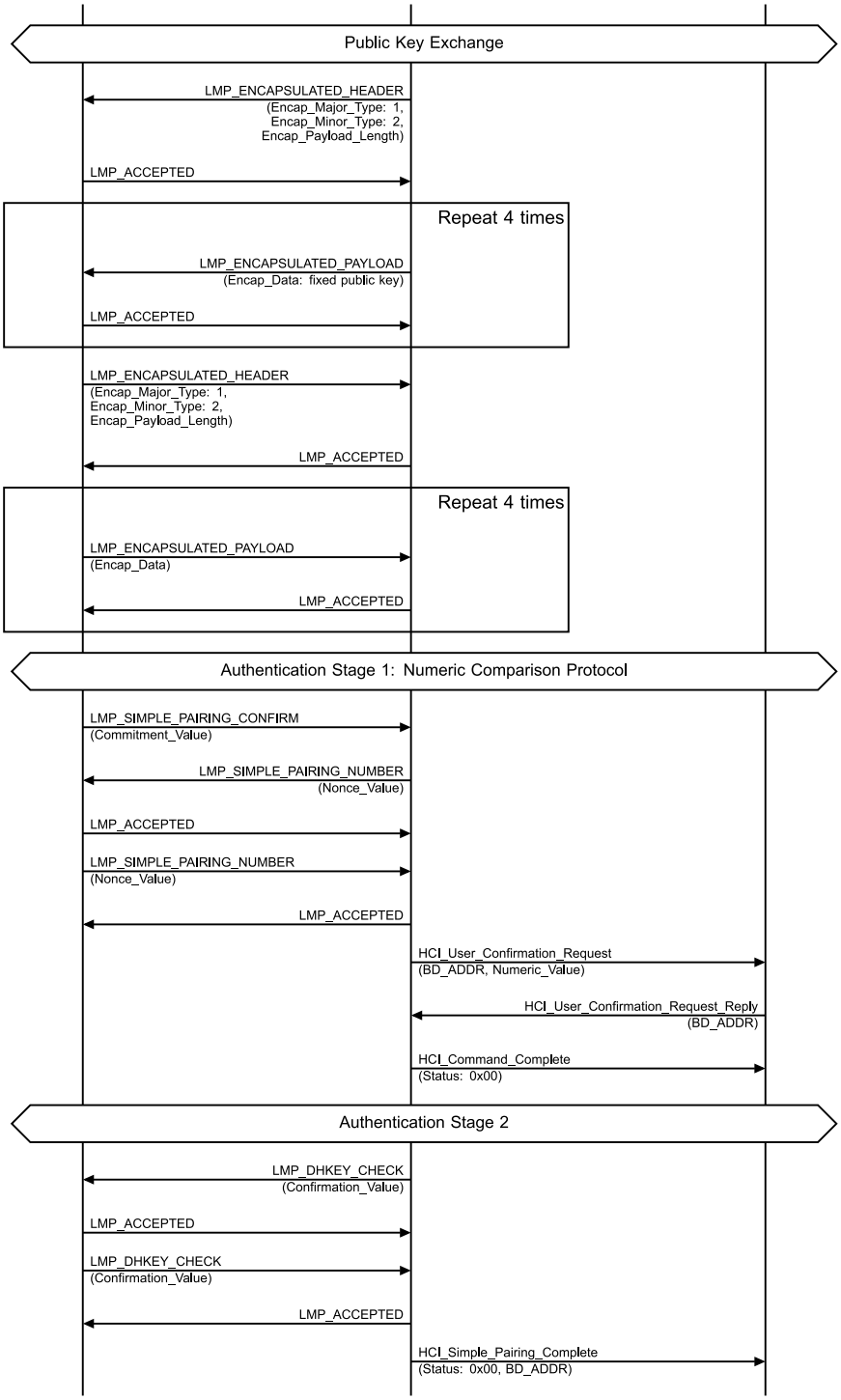


Figure 4.11-70: LMP/SP/BV-64-C [Secure Simple Pairing Debug Mode – Fixed Private Key, P256] MSC – Page 2 of 3

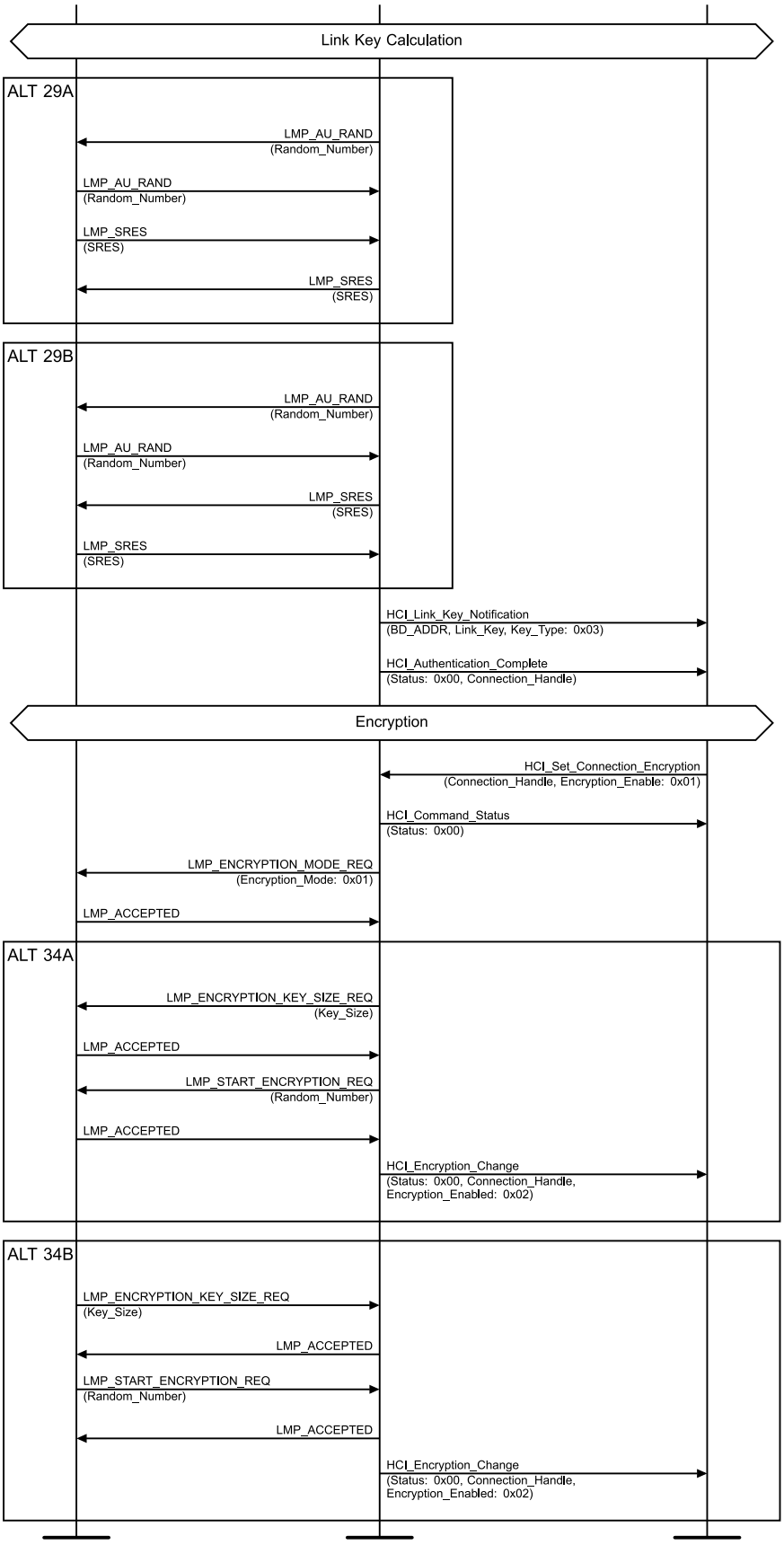


Figure 4.11-71: LMP/SP/BV-64-C [Secure Simple Pairing Debug Mode – Fixed Private Key, P256] MSC – Page 3 of 3

1. The Upper Tester sends an HCI_Write_Simple_Pairing_Debug_Mode command to the IUT with Simple_Pairing_Debug_Mode set to 0x01 (Enabled).
2. Execute the test procedure steps of LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256] once, except that in Step 11 the IUT sends the fixed public key, and in Step 30 the Key_Type is set to 0x03 (Debug Combination Key).

- Expected Outcome

  Pass verdict

  The IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x03 (Debug Combination Key).

LMP/SP/BV-65-C [Secure Simple Pairing Debug Mode – Responding Device Uses Fixed Private Key, P256]

- Test Purpose

  Verify that the IUT reports the Key_Type as Debug Combination Key when the remote device uses the Secure Simple Pairing debug mode and the fixed P-256 private/public key pair is used.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Baseband assumptions” section and the “Secure Simple Pairing P-256” default settings.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

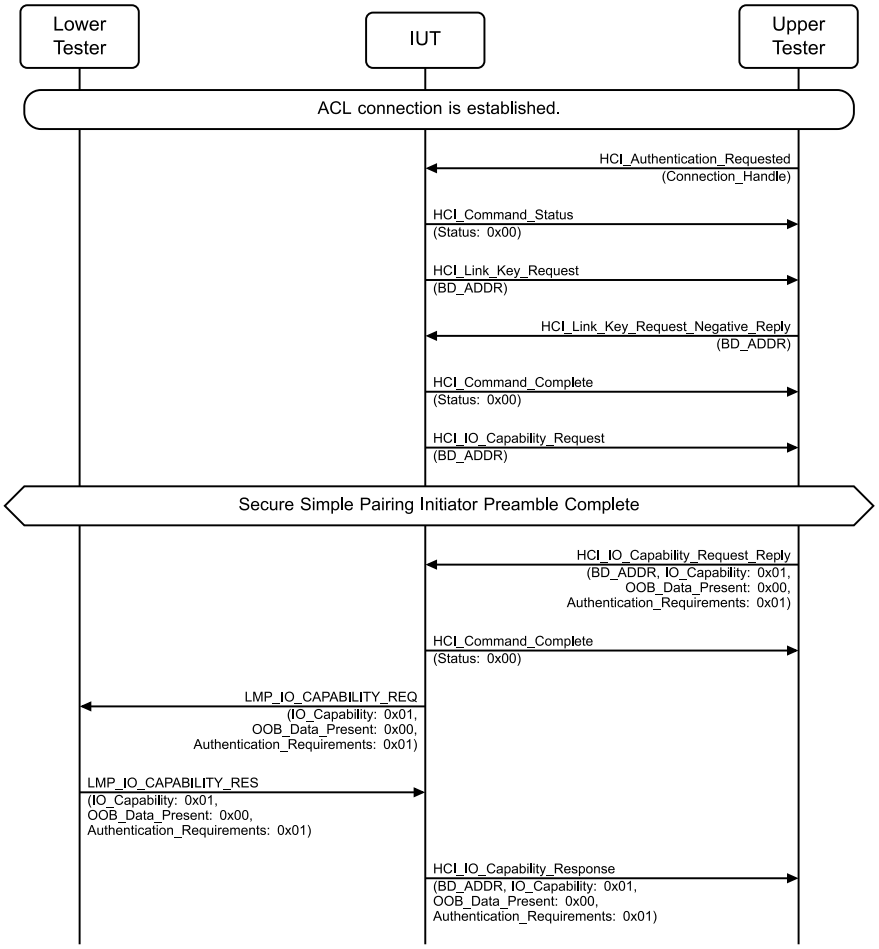


Figure 4.11-72: LMP/SP/BV-65-C [Secure Simple Pairing Debug Mode – Responding Device Uses Fixed Private Key, P256] MSC – Page 1 of 3

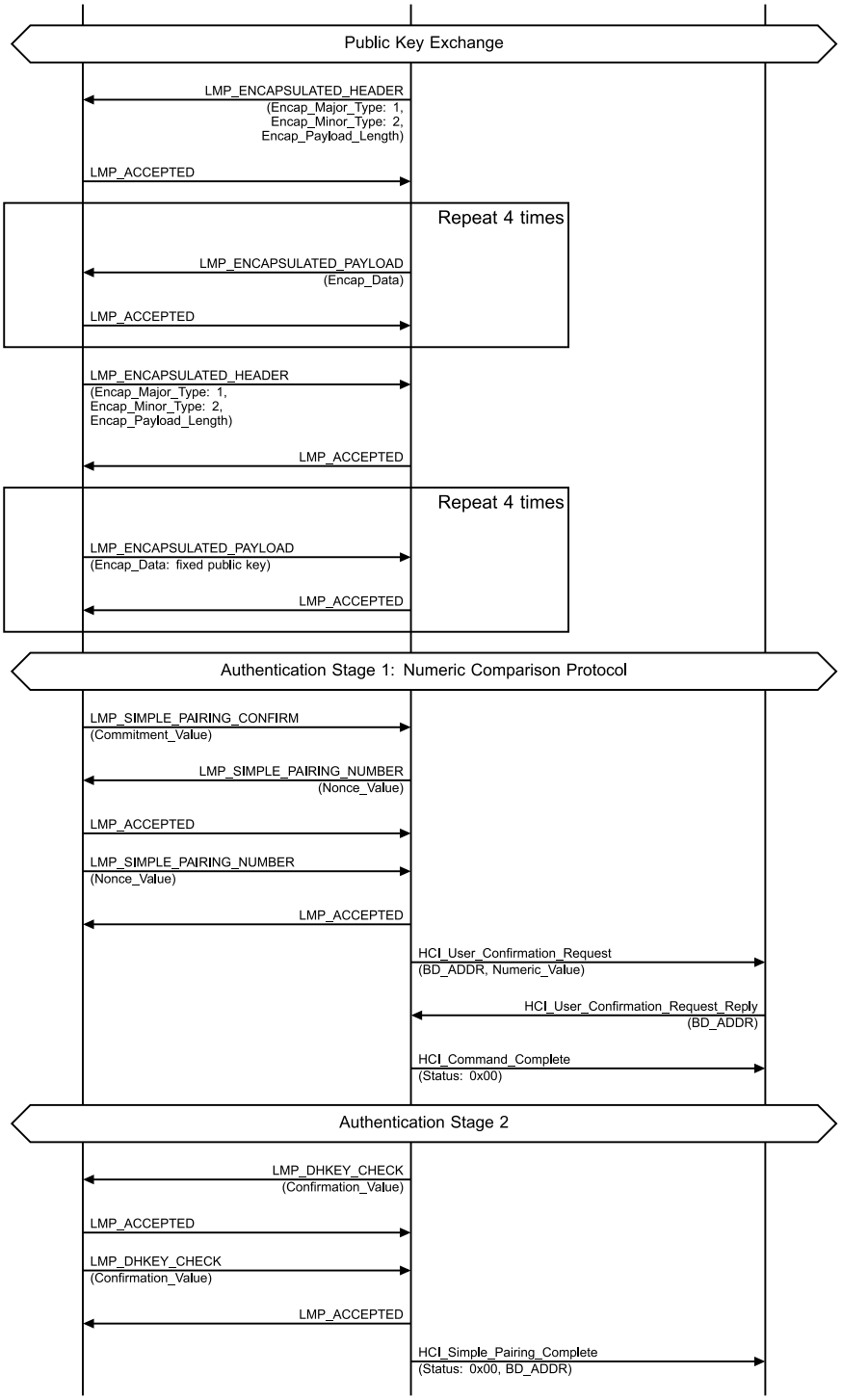


Figure 4.11-73: LMP/SP/BV-65-C [Secure Simple Pairing Debug Mode – Responding Device Uses Fixed Private Key, P256] MSC – Page 2 of 3

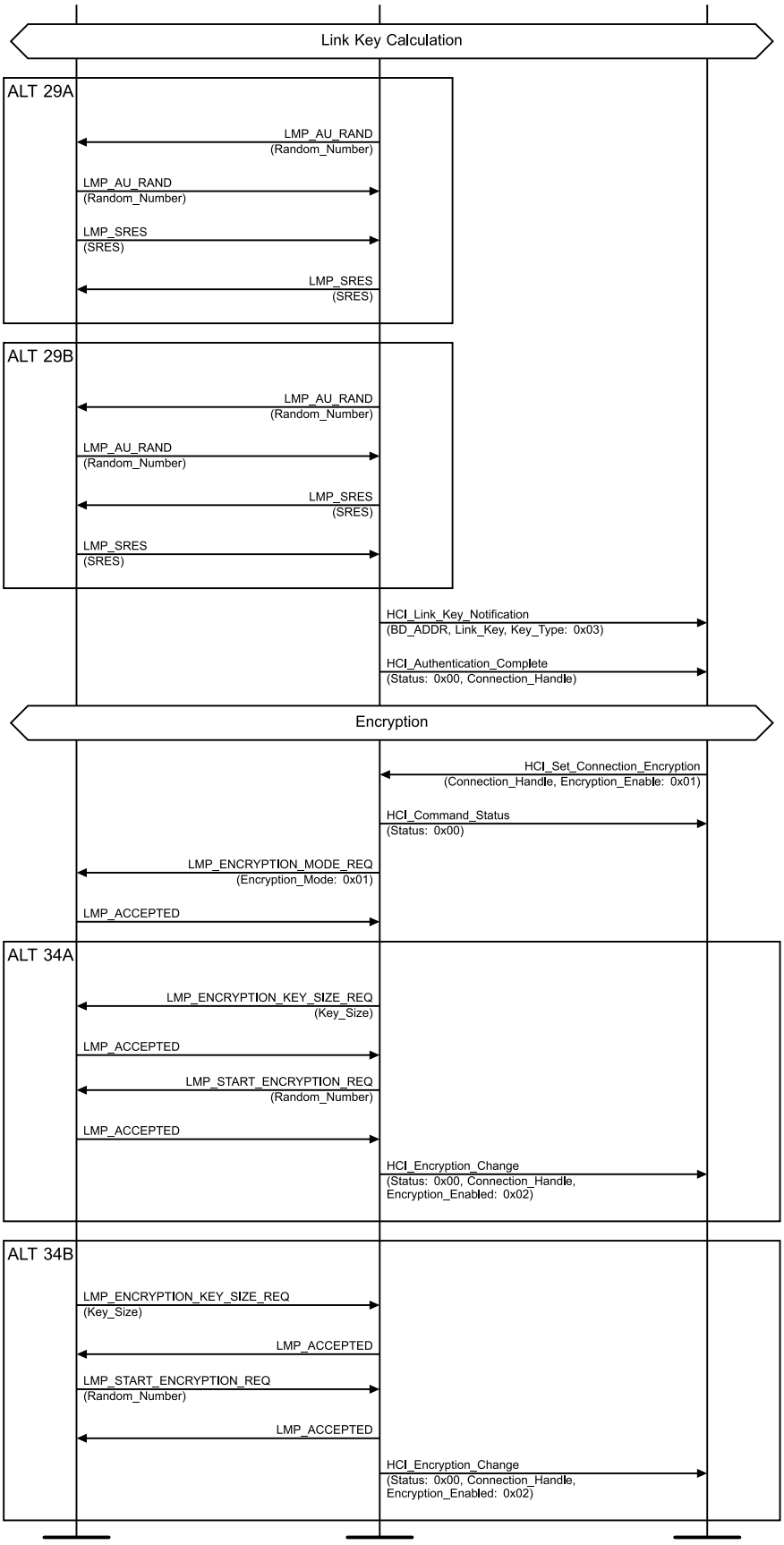


Figure 4.11-74: LMP/SP/BV-65-C [Secure Simple Pairing Debug Mode – Responding Device Uses Fixed Private Key, P256] MSC – Page 3 of 3

1. Execute the test procedure steps of LMP/SP/BV-41-C [Numeric Comparison – IUT Initiator – Success, P-256] once, except that in Step 15 the Lower Tester sends the fixed public key, and in Step 30 the Key_Type is set to 0x03 (Debug Combination Key).

- Expected Outcome

  Pass verdict

  The IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches the Upper and Lower Testers, and the Key_Type is 0x03 (Debug Combination Key).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

4.11.7 Simple Pairing Failure procedures

LMP/SP/BV-30-C [Secure Simple Pairing Failed - IUT Responder]

- Test Purpose

  Verify that the IUT responds correctly when the IO capability exchange procedure fails.

- Reference

  [8] 7.7.45

- Initial Condition

  - See the “Default settings” section.
  - The Lower Tester supports Secure Simple Pairing.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Procedure

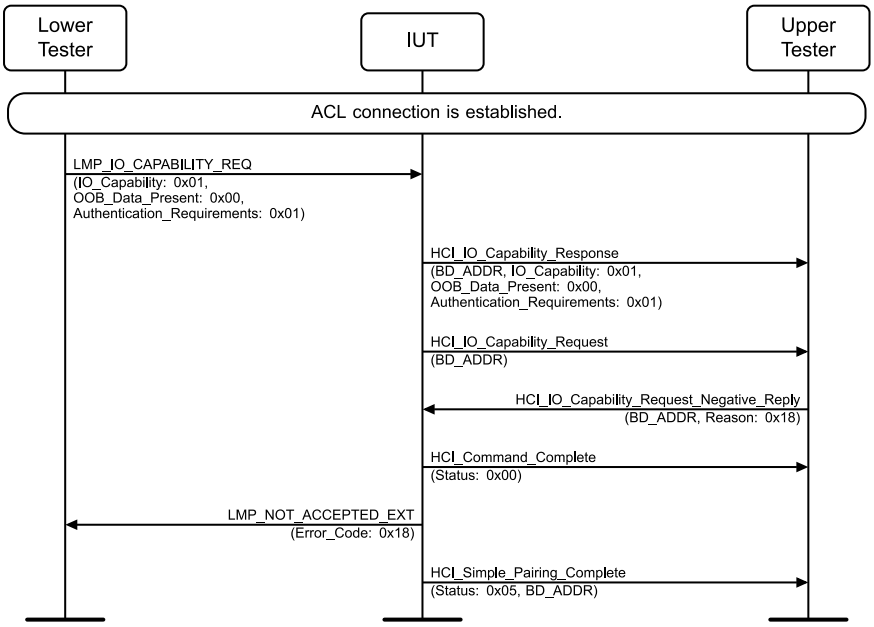


Figure 4.11-75: LMP/SP/BV-30-C [Secure Simple Pairing Failed - IUT Responder] MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Negative_Reply command with the BD_ADDR and Reason set to 0x18 (Pairing not Allowed) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with the LMP_IO_CAPABILITY_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x18 (Pairing not Allowed).
6. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  In Step 5, the IUT sends the LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with the Error_Code provided by the local Host.

  In Step 6, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

LMP/SP/BV-31-C [Secure Simple Pairing Capable Controller–Host rejects Secure Simple Pairing–IUT Initiator]

- Test Purpose

  Verify that the LM on the IUT accepts the rejection of Secure Simple Pairing when the Host sends an HCI_IO_Capability_Request_Negative_Reply command.

- Reference

  [8] 7.7.50

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Initiator and a Secure Simple Pairing–capable controller.
  - The Upper Tester is a Secure Simple Pairing–capable Host.
  - The Lower Tester is a Secure Simple Pairing–enabled Responder.
  - An ACL connection has been established.

- Test Procedure

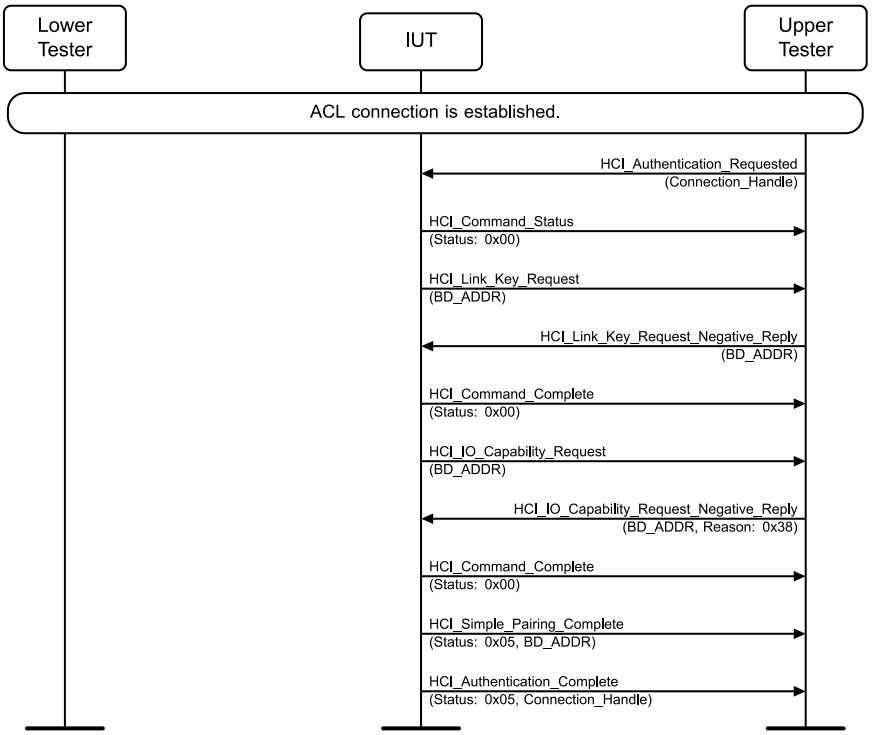


Figure 4.11-76: LMP/SP/BV-31-C [Secure Simple Pairing Capable Controller–Host rejects Secure Simple Pairing–IUT Initiator] MSC

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.

5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Negative_Reply command with the BD_ADDR and Reason set to 0x38 (Host Busy - Pairing) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  The IUT responds to the HCI_IO_Capability_Request_Negative_Reply command from the Upper Tester with an HCI_Command_Complete event.

  The IUT also sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

LMP/SP/BV-32-C [Secure Simple Pairing Capable Controller–Host rejects Secure Simple Pairing–IUT Responder]

- Test Purpose

  Verify that the LM on the IUT rejects Secure Simple Pairing when the Host sends an HCI_IO_Capability_Request_Negative_Reply command.

- Reference

  [8] 7.71.36

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Responder and a Secure Simple Pairing–capable controller.
  - The Upper Tester is a Secure Simple Pairing–capable Host.
  - The Lower Tester is a Secure Simple Pairing–enabled Initiator.
  - An ACL connection has been established.

- Test Procedure

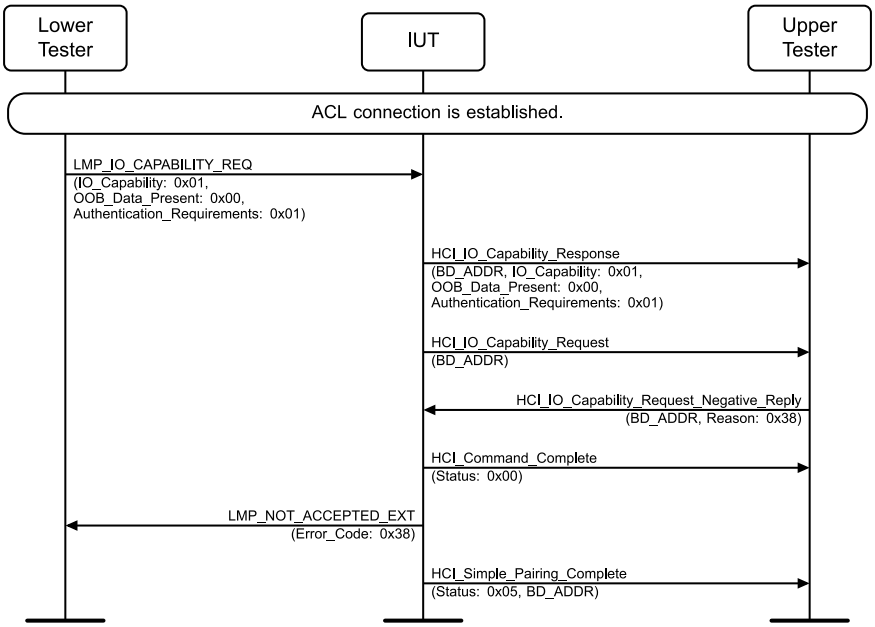


Figure 4.11-77: LMP/SP/BV-32-C [Secure Simple Pairing Capable Controller–Host rejects Secure Simple Pairing–IUT Responder] MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Negative_Reply command with the BD_ADDR and Reason set to 0x38 (Host Busy - Pairing) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with the LMP_IO_CAPABILITY_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x38 (Host Busy - Pairing).
6. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  In Step 5, the IUT responds to the Lower Tester's LMP_IO_CAPABILITY_REQ PDU with an LMP_NOT_ACCEPTED_EXT PDU with Error_Code set to 0x38 (Host Busy - Pairing).

  In Step 6, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

LMP/SP/BV-33-C [Passkey Entry with Keypress notification - IUT Initiator - Success]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

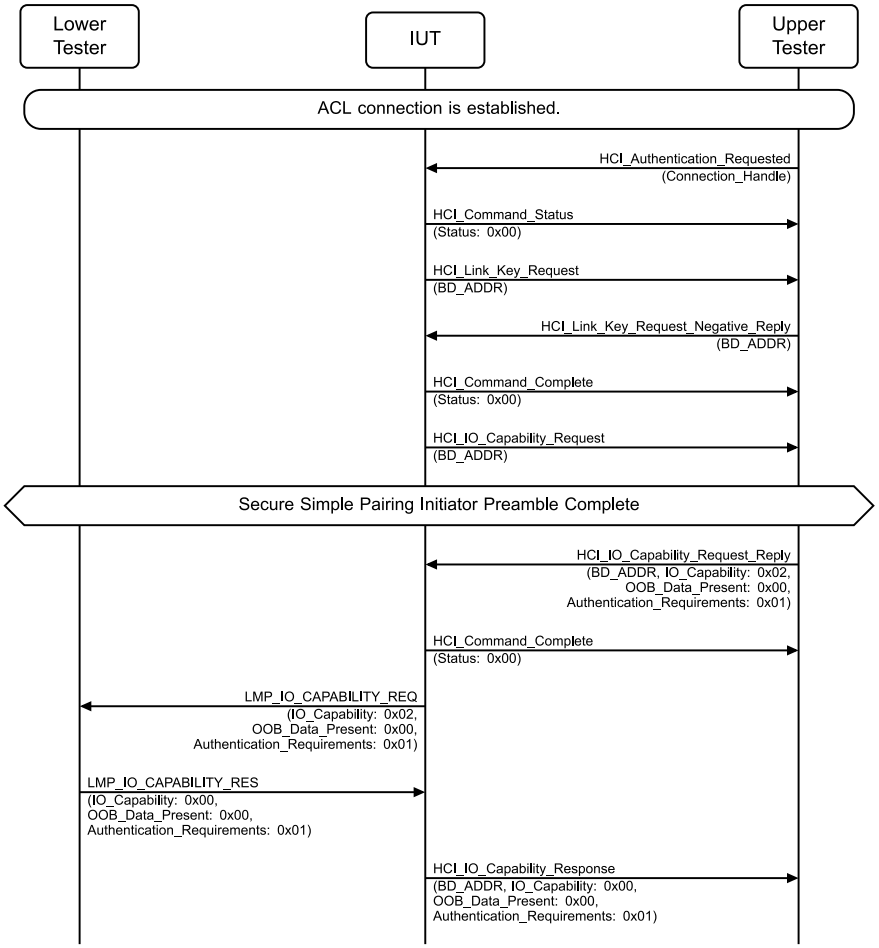


Figure 4.11-78: LMP/SP/BV-33-C [Passkey Entry with Keypress notification - IUT Initiator - Success] MSC – Page 1 of 3

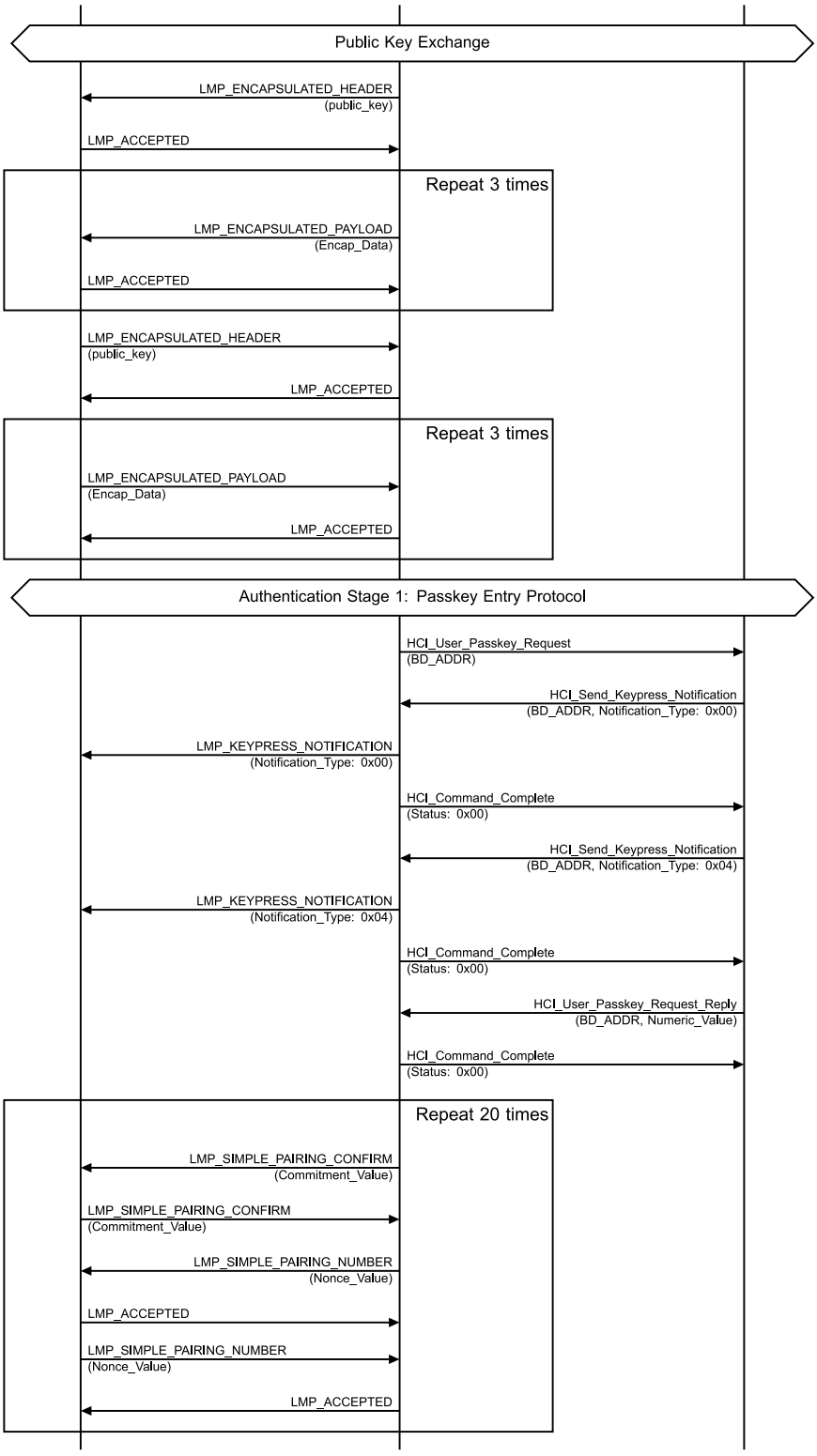


Figure 4.11-79: LMP/SP/BV-33-C [Passkey Entry with Keypress notification - IUT Initiator - Success] MSC – Page 2 of 3

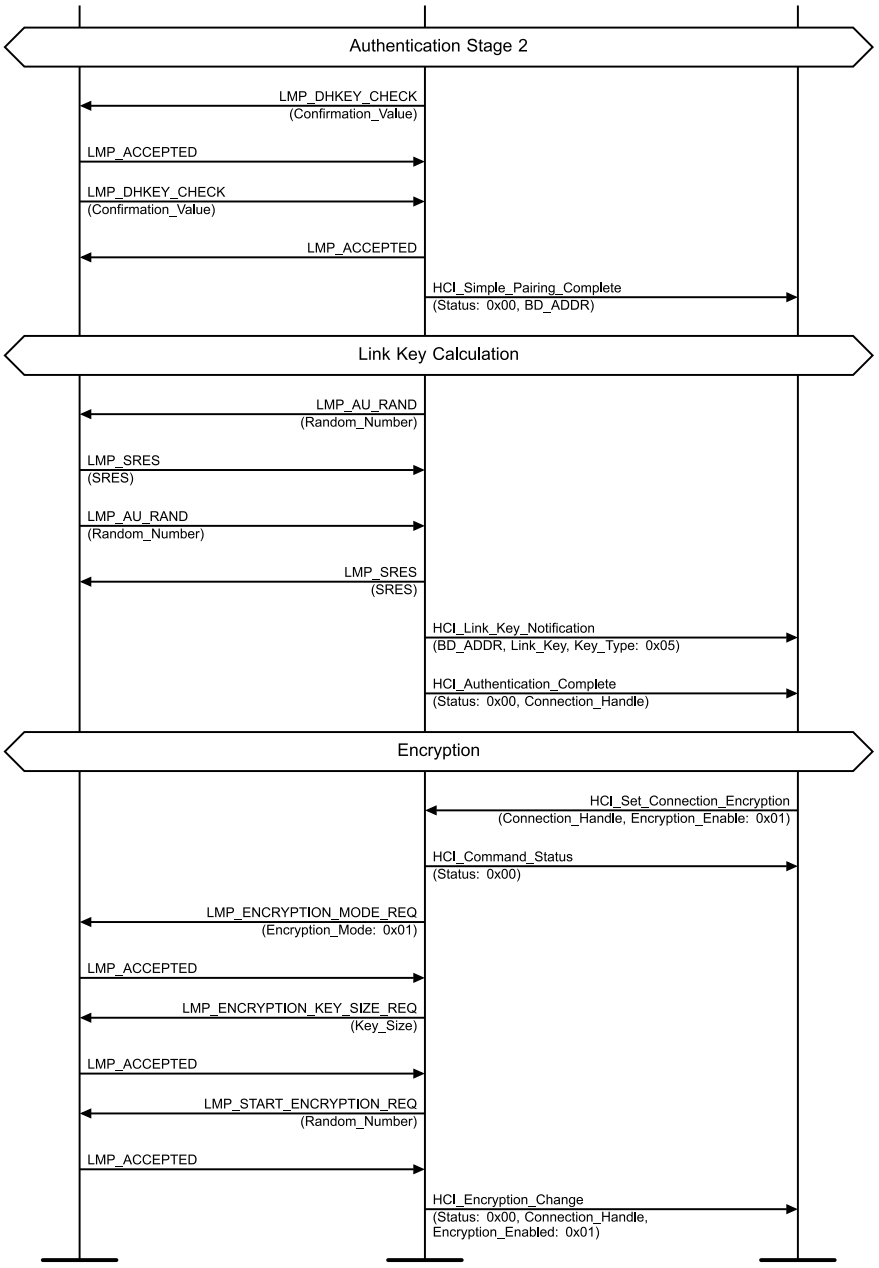


Figure 4.11-80: LMP/SP/BV-33-C [Passkey Entry with Keypress notification - IUT Initiator - Success] MSC – Page 3 of 3

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.

6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds to the IUT with an HCI_Send_Keypress_Notification command with the BD_ADDR and Notification_Type set to 0x00 (Passkey entry started).
19. The IUT sends an LMP_KEYPRESS_NOTIFICATION PDU to the Lower Tester with Notification_Type set to 0x00 (Passkey entry started).
20. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
21. The Upper Tester sends an HCI_Send_Keypress_Notification command to the IUT with the BD_ADDR and Notification_Type set to 0x04 (Passkey entry completed).
22. The IUT sends an LMP_KEYPRESS_NOTIFICATION PDU to the Lower Tester with Notification_Type set to 0x04 (Passkey entry completed).
23. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
24. The Upper Tester responds to the IUT with an HCI_User_Passkey_Request_Reply command with the BD_ADDR and Numeric_Value and receives a successful HCI_Command_Complete event in response.

Repeat Steps 25–30 20 times.

25. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
26. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.

27. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
28. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
29. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
30. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
31. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
32. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
33. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
34. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
35. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
36. The IUT sends an LMP_AU_RAND PDU to the Lower Tester with a Random_Number.
37. The Lower Tester responds to the IUT with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
38. The IUT sends an LMP_SRES PDU to the Lower Tester with the SRES.
39. The IUT sends an HCI_Link_Key_Notification event to the IUT with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192) and a successful HCI_Authentication_Complete event with the Connection_Handle.
40. The Upper Tester sends an HCI_Set_Connection_Encryption command to the IUT with the Connection_Handle and Encryption_Enable set to 0x01 (On) and receives a successful HCI_Command_Status event in response.
41. The IUT sends an LMP_ENCRYPTION_MODE_REQ PDU to the Lower Tester with Encryption_Mode set to 0x01 (Encryption).
42. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
43. The IUT sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the Lower Tester with the Key_Size.
44. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
45. The IUT sends an LMP_START_ENCRYPTION_REQ PDU to the Lower Tester with a Random_Number.
46. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
47. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

- Expected Outcome

  Pass verdict

  In Step 31, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 35, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 39, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

- Notes

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-34-C [Passkey Entry with Keypress notification - IUT Responder – Success]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Procedure

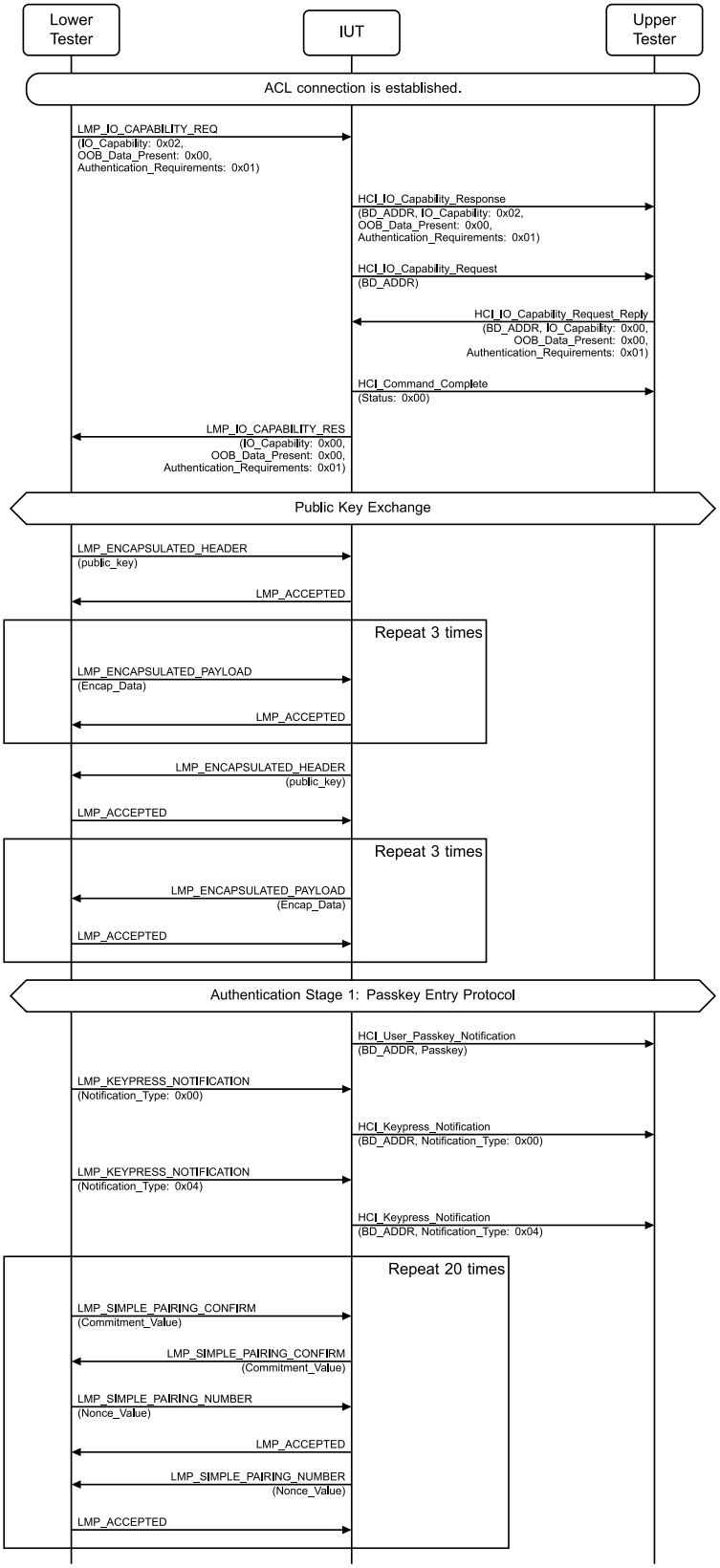


Figure 4.11-81: LMP/SP/BV-34-C [Passkey Entry with Keypress notification - IUT Responder – Success] MSC – Page 1 of 2

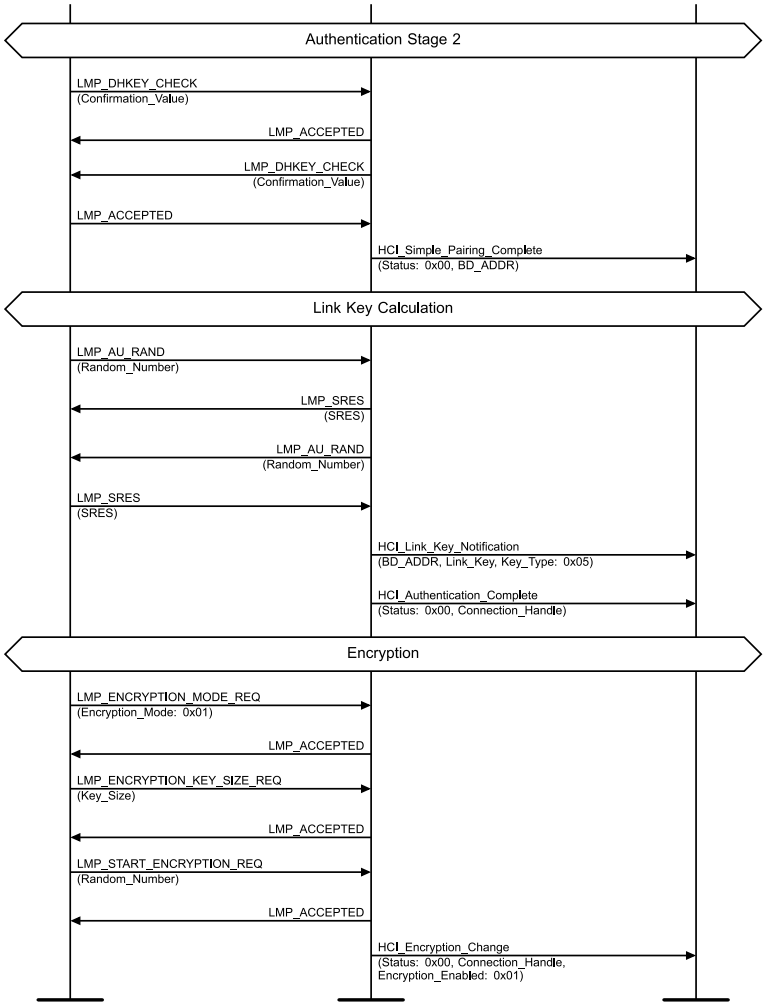


Figure 4.11-82: LMP/SP/BV-34-C [Passkey Entry with Keypress notification - IUT Responder – Success] MSC – Page 2 of 2

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.
15. The Lower Tester sends an LMP_KEYPRESS_NOTIFICATION PDU to the IUT with Notification_Type set to 0x00 (Passkey entry started).
16. The IUT sends an HCI_Keypress_Notification event to the Upper Tester with the BD_ADDR and Notification_Type set to 0x00 (Passkey entry started).
17. The Lower Tester sends an LMP_KEYPRESS_NOTIFICATION PDU to the IUT with Notification_Type set to 0x04 (Passkey entry completed).
18. The IUT sends an HCI_Keypress_Notification event to the Upper Tester with the BD_ADDR and Notification_Type set to 0x04 (Passkey entry completed).

Repeat Steps 19–24 20 times.

19. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
20. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
21. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
22. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
23. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
24. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
25. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
26. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
27. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
28. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
29. The IUT sends a successful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.
30. The Lower Tester sends an LMP_AU_RAND PDU to the IUT with a Random_Number.
31. The IUT responds to the Lower Tester with an LMP_SRES PDU with the SRES and an LMP_AU_RAND PDU with a Random_Number.
32. The Lower Tester sends an LMP_SRES PDU to the IUT with the SRES.

33. The IUT sends an HCI_Link_Key_Notification event to the IUT with the BD_ADDR, Link_Key, and Key_Type set to 0x05 (Authenticated Combination Key generated from P-192) and a successful HCI_Authentication_Complete event with the Connection_Handle.
34. The Lower Tester sends an LMP_ENCRYPTION_MODE_REQ PDU to the IUT with Encryption_Mode set to 0x01 (Encryption).
35. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_MODE_REQ PDU Opcode.
36. The Lower Tester sends an LMP_ENCRYPTION_KEY_SIZE_REQ PDU to the IUT with the Key_Size.
37. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCRYPTION_KEY_SIZE_REQ PDU Opcode.
38. The Lower Tester sends an LMP_START_ENCRYPTION_REQ PDU to the IUT with a Random_Number.
39. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_START_ENCRYPTION_REQ PDU Opcode.
40. The IUT sends a successful HCI_Encryption_Change event to the Upper Tester with the Connection_Handle and Encryption_Enabled set to 0x01.

- Expected Outcome

  Pass verdict

  In Step 27, the IUT sends the LMP_DHKEY_CHECK PDU to the Lower Tester with a valid Confirmation_Value.

  In Step 29, the IUT sends the HCI_Simple_Pairing_Complete event to the Upper Tester.

  In Step 33, the IUT sends the resulting Link Key and Key Type to the Host in an HCI_Link_Key_Notification event. The Link Key matches at the Upper and Lower Testers, and the Key_Type is 0x05 (Authenticated Combination Key generated from P-192).

- Notes

  If the Commitment_Value calculated by the Lower Tester does not match the Commitment_Value sent by the IUT, the Lower Tester sends the LMP_NOT_ACCEPTED PDU to the IUT with the Authentication Failure Error_Code.

  The Lower Tester sends an LMP_NOT_ACCEPTED PDU to the IUT in response to the LMP_DHKEY_CHECK PDU from the IUT if the Confirmation_Value that it calculates does not match the Confirmation_Value that the IUT has sent.

LMP/SP/BV-35-C [Passkey Entry with Keypress notification - IUT Initiator - Failure on Responding Side]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol and that the IUT responds correctly when the responding side fails the passkey entry check step.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Initiator.
  - An ACL connection has been established.

- Test Procedure

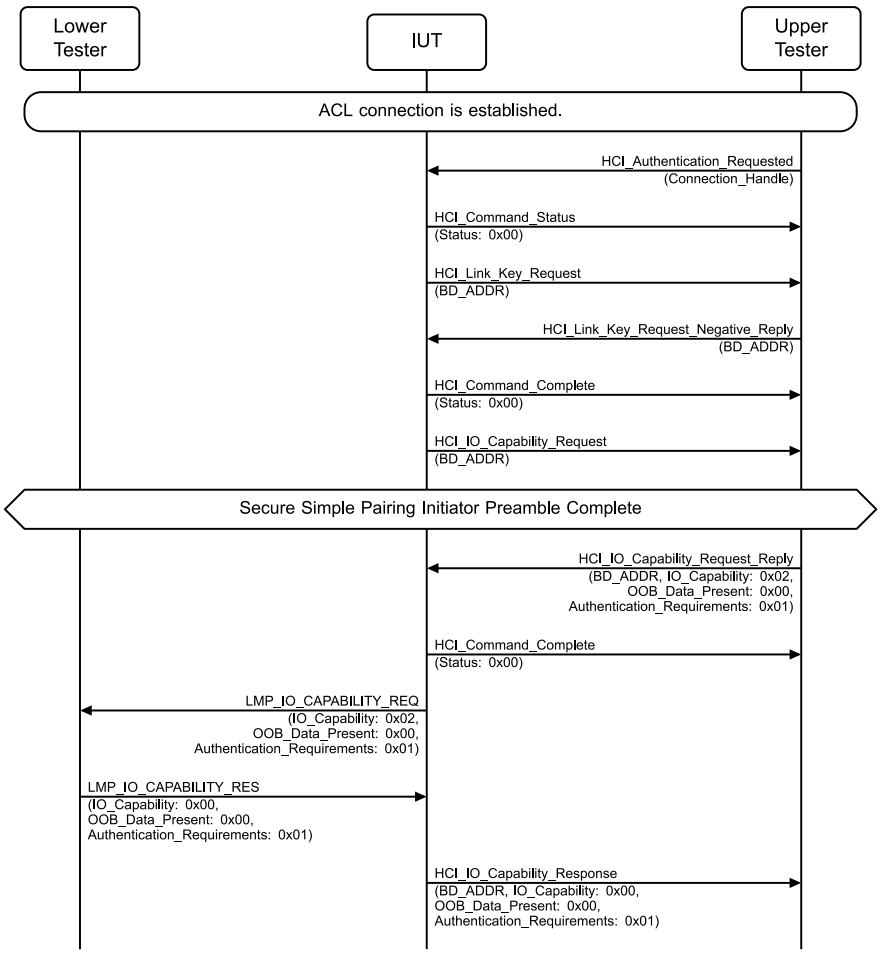


Figure 4.11-83: LMP/SP/BV-35-C [Passkey Entry with Keypress notification - IUT Initiator - Failure on Responding Side] MSC – Page 1 of 2

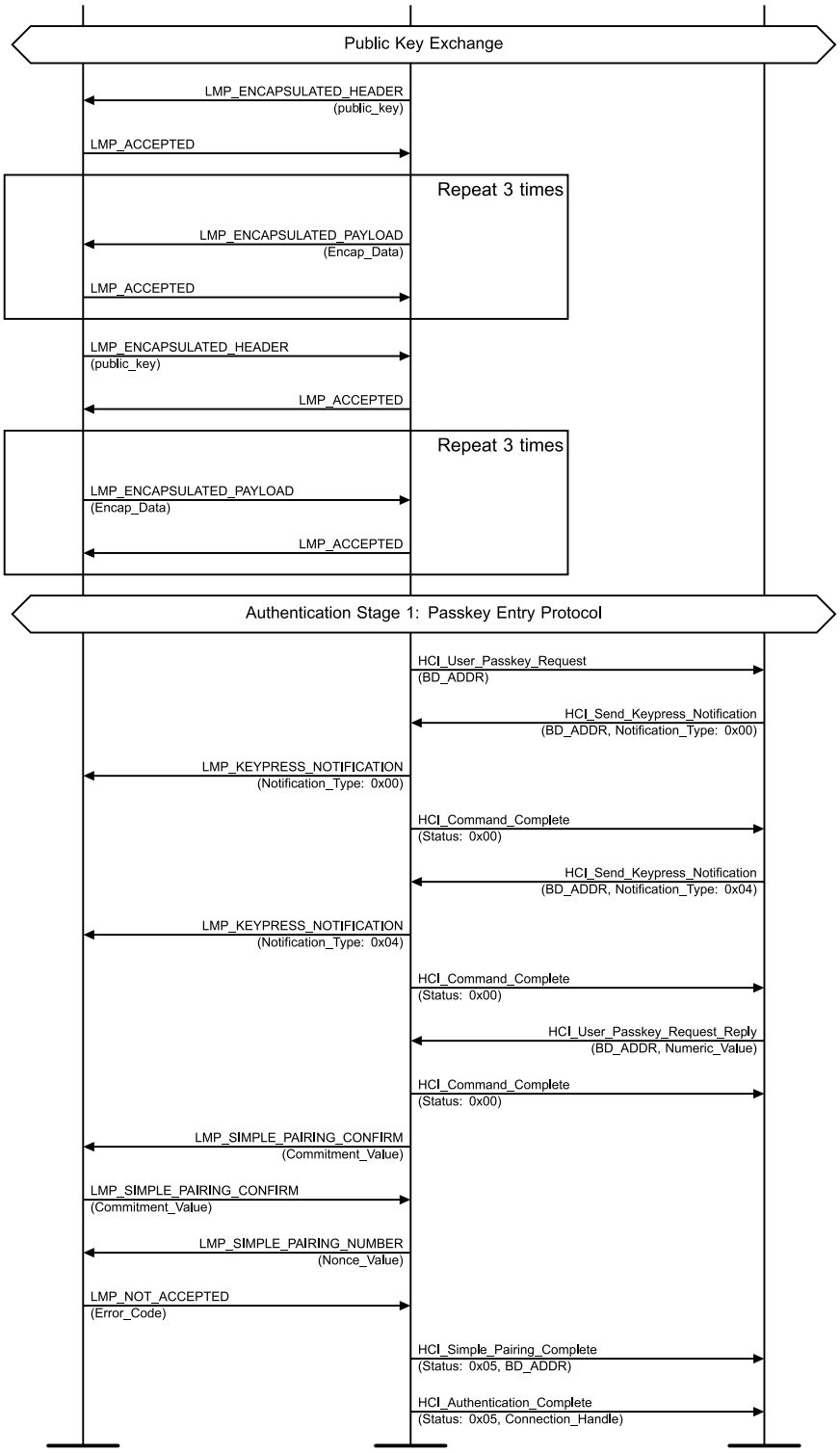


Figure 4.11-84: LMP/SP/BV-35-C [Passkey Entry with Keypress notification - IUT Initiator - Failure on Responding Side] MSC – Page 2 of 2

1. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
2. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
3. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
4. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
5. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
6. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
7. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
8. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
10. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times.

11. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times.

15. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds to the IUT with an HCI_Send_Keypress_Notification command with the BD_ADDR and Notification_Type set to 0x00 (Passkey entry started).
19. The IUT sends an LMP_KEYPRESS_NOTIFICATION PDU to the Lower Tester with Notification_Type set to 0x00 (Passkey entry started).
20. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
21. The Upper Tester sends an HCI_Send_Keypress_Notification command to the IUT with the BD_ADDR and Notification_Type set to 0x04 (Passkey entry completed).

22. The IUT sends an LMP_KEYPRESS_NOTIFICATION PDU to the Lower Tester with Notification_Type set to 0x04 (Passkey entry completed).
23. The IUT sends a successful HCI_Command_Complete event to the Upper Tester.
24. The Upper Tester responds to the IUT with an HCI_User_Passkey_Request_Reply command with the BD_ADDR and Numeric_Value and receives a successful HCI_Command_Complete event in response.
25. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
26. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
27. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
28. The Lower Tester responds to the IUT with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.
29. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
30. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
31. The IUT sends an HCI_Simple_Pairing_Complete event with Status set to 0x05 (Authentication Failure) and the BD_ADDR and an HCI_Authentication_Complete event with Status set to 0x05 (Authentication Failure) and the Connection_Handle.

- Expected Outcome

  Pass verdict

  The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) after the Lower Tester responds with an LMP_NOT_ACCEPTED PDU to the LMP_SIMPLE_PAIRING_NUMBER PDU send by the IUT.

LMP/SP/BV-36-C [Passkey Entry with Keypress notification - IUT Responder - Failure on Responding Side]

- Test Purpose

  Verify that the IUT supports the Passkey Entry protocol and that the IUT responds correctly when the responding side fails the passkey entry check step.

- Reference

  [1] 4.2.7

- Initial Condition

  - See the “Default settings” section.
  - The IUT is the Responder.
  - An ACL connection has been established.

- Test Procedure

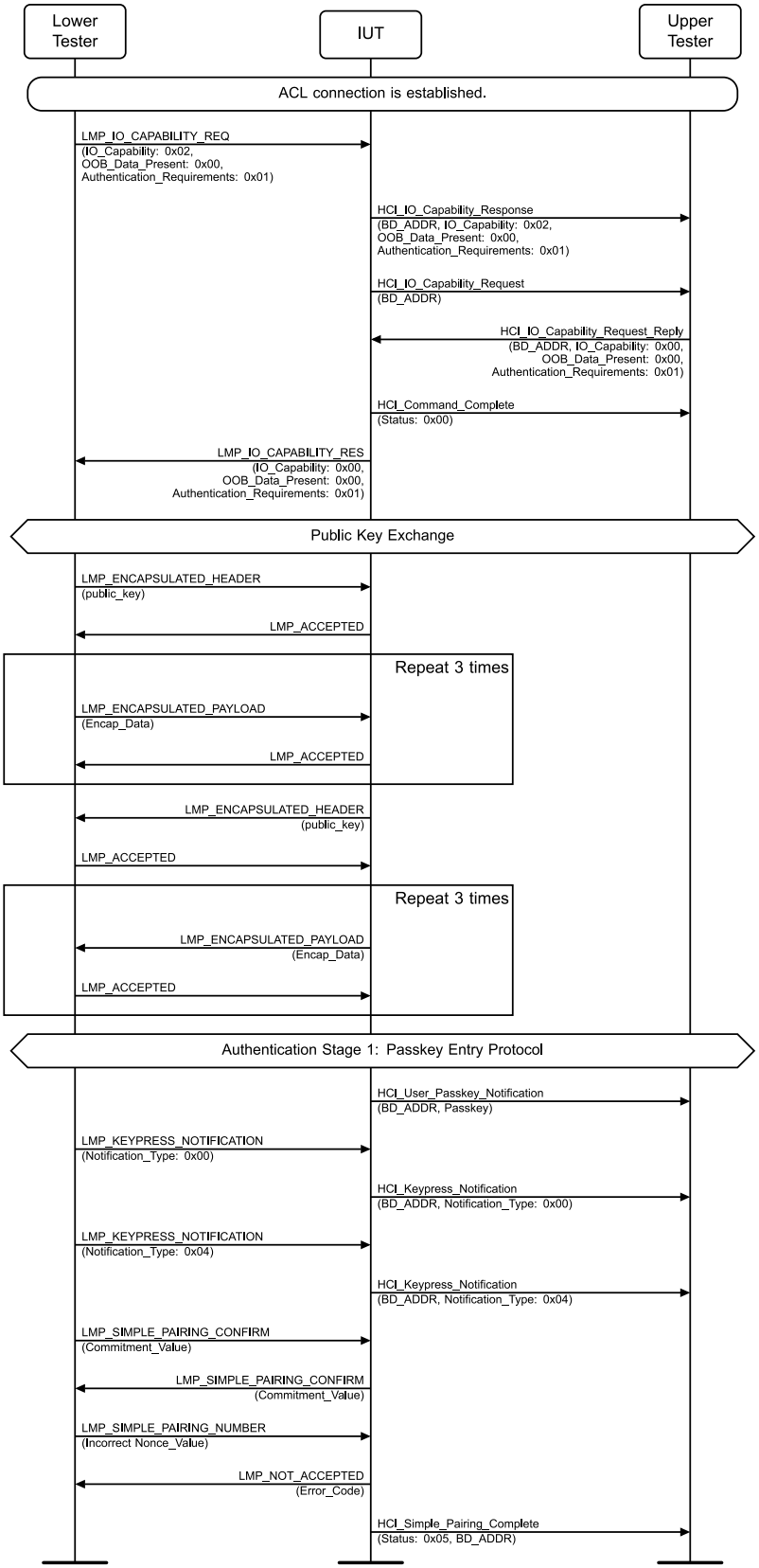


Figure 4.11-85: LMP/SP/BV-36-C [Passkey Entry with Keypress notification - IUT Responder - Failure on Responding Side] MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
7. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 8–9 three times.

8. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the Encap_Data.
9. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
10. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
11. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 12–13 three times.

12. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
14. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.
15. The Lower Tester sends an LMP_KEYPRESS_NOTIFICATION PDU to the IUT with Notification_Type set to 0x00 (Passkey entry started).
16. The IUT sends an HCI_Keypress_Notification event to the Upper Tester with the BD_ADDR and Notification_Type set to 0x00 (Passkey entry started).
17. The Lower Tester sends an LMP_KEYPRESS_NOTIFICATION PDU to the IUT with Notification_Type set to 0x04 (Passkey entry completed).
18. The IUT sends an HCI_Keypress_Notification event to the Upper Tester with the BD_ADDR and Notification_Type set to 0x04 (Passkey entry completed).
19. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
20. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
21. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with an invalid Nonce_Value.

22. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode and an Error_Code.
23. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester in response to the LMP_SIMPLE_PAIRING_NUMBER PDU sent by the Lower Tester.

  The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure).

4.11.7.1 Numeric Comparison – IUT Initiator – Invalid public key failure

- Test Purpose

  Verify that the IUT detects an invalid public key using the Numeric Comparison protocol and fails the pairing procedure using the key specified in Table 4.11-18.

- Reference

  [1] 4.2.7

- Initial Condition

  - The IUT is the Initiator.
  - TSPX_new_key_failed_count gives the number of failed pairing attempts before a new pairing key is generated for Table 4.11-19.
  - The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.

- Test Case Configuration

Test Case	Key
LMP/SP/BI-01-C [Numeric Comparison – IUT Initiator – Invalid Public Key Failure, P-192]	P-192
LMP/SP/BI-07-C [Numeric Comparison – IUT Initiator – Invalid Public Key Failure, P-256]	P-256

Table 4.11-18: Numeric Comparison – IUT Initiator – Invalid public key failure test cases

- Test Procedure

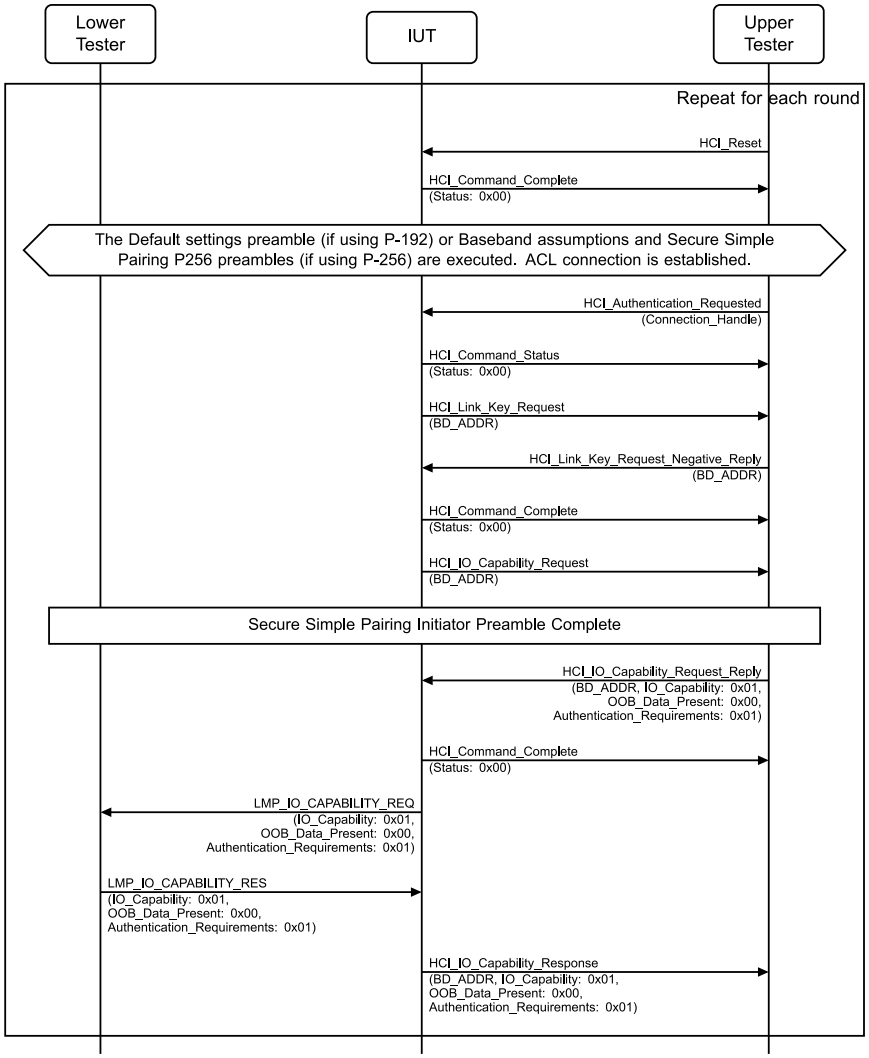


Figure 4.11-86: Numeric Comparison – IUT Initiator – Invalid public key failure MSC – Page 1 of 2

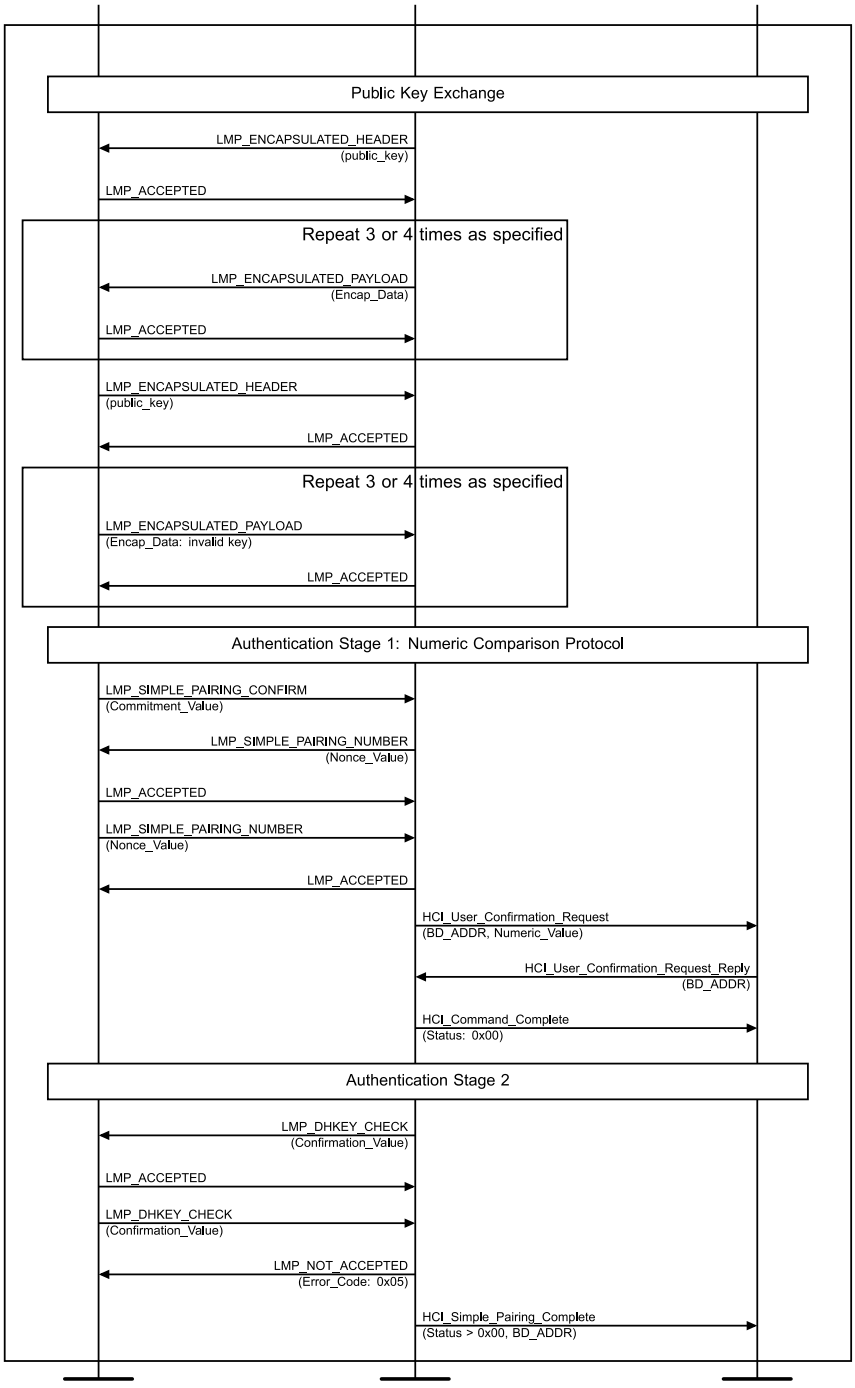


Figure 4.11-87: Numeric Comparison – IUT Initiator – Invalid public key failure MSC – Page 2 of 2

Repeat Steps 1–31 for each round and repetition in Table 4.11-19.

1. The Upper Tester sends an HCI_Reset command to the IUT and receives a successful HCI_Command_Complete event in response.
2. Execute the “Default settings” preamble if using P-192 or the “Baseband assumptions” and “Secure Simple Pairing P-256” preambles if using P-256.
3. An ACL connection is established between the IUT and the Lower Tester.
4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.

6. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
7. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
8. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
9. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
10. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
11. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
12. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 14–15 three times if using P-192 or four times if using P-256 as specified in Table 4.11-18.

14. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
15. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
16. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 18–19 three times if using P-192 or four times if using P-256 as specified in Table 4.11-18.

18. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the invalid key specified in Table 4.11-19.
19. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
21. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
22. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
23. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
24. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.

25. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
26. The Upper Tester sends an HCI_User_Confirmation_Request_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
27. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
28. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
29. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
30. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and Error_Code set to 0x05 (Authentication Failure).
31. The IUT sends an unsuccessful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.

Round	Key_Size	Invalid Key Type	Repeat # of times	Lower Tester DHKey
1	P-192	Generate valid public Key and set y-coordinate = 0	Max(20* TSPX_new_key_failed_count, 20)	0
2	P-192	Generate valid public Key and set y-coordinate = 0	1	Computed DHKey
3	P-192	Generate valid public Key and flip a bit in y-coordinate	1	Computed DHKey
4	P-192	Public Key coordinates (0, 0)	1	0
5	P-192	Generate valid public key and set x-coordinate same as IUT's	1	Computed DHKey

Note: In Authentication Stage 2, the Lower Tester uses either the computed DHKey or DHKey = 0 as specified in Table 4.11-19.

Table 4.11-19: Invalid public key generation for each round

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x05 (Authentication Failure).

  In Step 31, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status not set to Success (0x00).

  If TSPX_new_key_failed_count > 0, a different public key is used by the IUT after at most TSPX_new_key_failed_count pairings.

  Inconclusive verdict

  The IUT interrupts the pairing process by:

  - Sending an LMP_NOT_ACCEPTED PDU with a different Error_Code than 0x05 (Authentication Failure) in response to an LMP_DHKEY_CHECK PDU
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_ENCAPSULATED_PAYLOAD PDU containing an invalid public key
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_SIMPLE_PAIRING_NUMBER PDU
  - Sending an LMP_NUMERIC_COMPARISON_FAILED PDU during the Numeric Comparison protocol phase

- Notes

  The test verifies the recommendation of the specification that an IUT should return an LMP_NOT_ACCEPTED PDU to the Lower Tester's LMP_DHKEY_CHECK PDU if the Lower Tester's public key is invalid. Other potentially valid ways of rejecting the invalid key are listed in the expected outcome and will yield an Inconclusive verdict.

  To simulate an attacker, the Lower Tester may send an LMP_ACCEPTED PDU in response to all calculated values sent by the IUT, even if the LMP_SIMPLE_PAIRING_CONFIRM, the LMP_SIMPLE_PAIRING_NUMBER, or the LMP_DHKEY_CHECK values sent by the IUT do not match the expected calculations.

4.11.7.2 Numeric Comparison – IUT Responder – Invalid Public Key Failure

- Test Purpose

  Verify that the IUT detects an invalid public key using the Numeric Comparison protocol and fails the pairing procedure using the key specified in Table 4.11-20.

- Reference

  [1] 4.2.7

- Initial Condition

  - The IUT is the Responder.
  - TSPX_new_key_failed_count gives the number of failed pairing attempts before a new pairing key is generated for Table 4.11-19.
  - The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.

- Test Case Configuration

Test Case	Key	Encapsulated Payload Repeat
LMP/SP/BI-02-C [Numeric Comparison – IUT Responder – Invalid Public Key Failure, P-192]	P-192	3
LMP/SP/BI-08-C [Numeric Comparison – IUT Responder – Invalid Public Key Failure, P-256]	Secure Simple Pairing P-256 with P-256	4

Table 4.11-20: Numeric Comparison – IUT Responder – Invalid Public Key Failure test cases

- Test Procedure

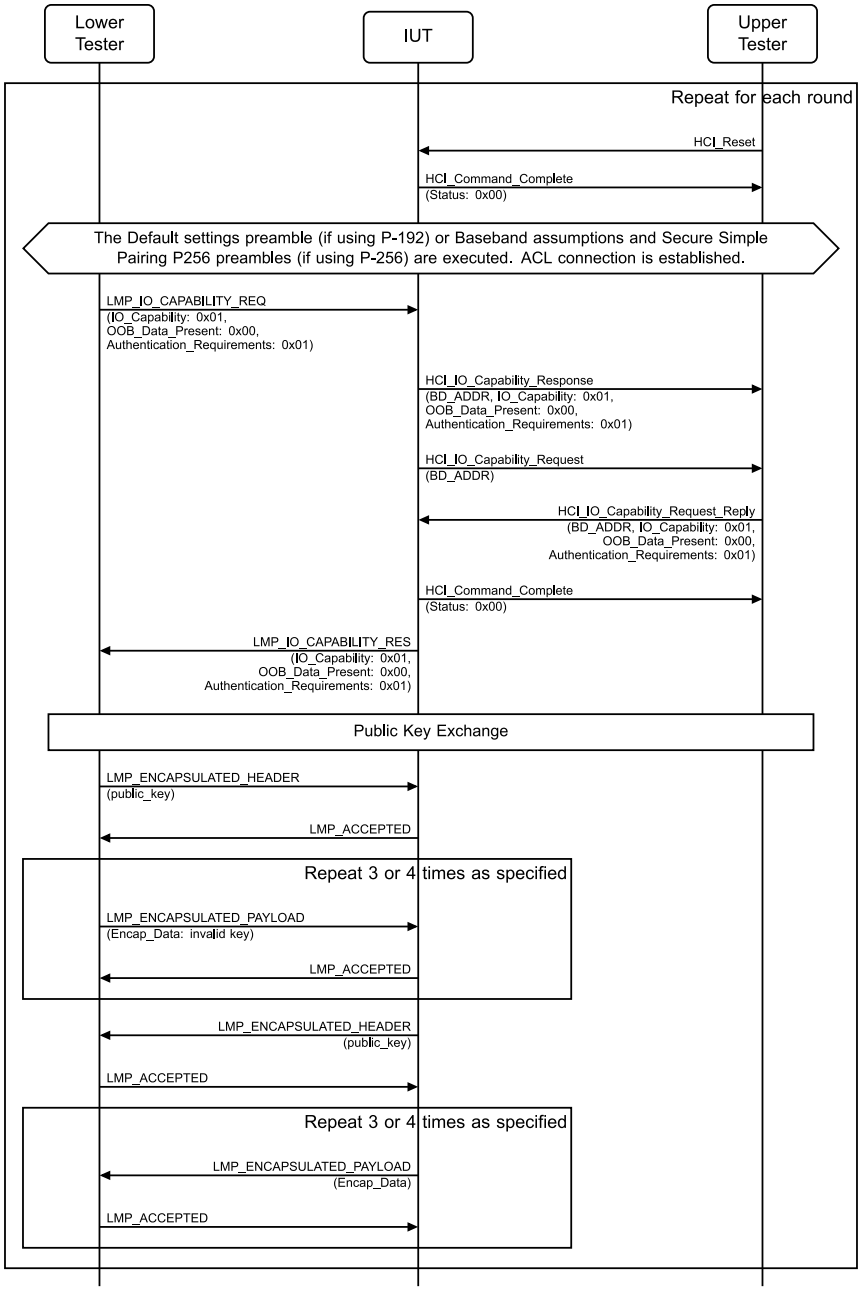


Figure 4.11-88: Numeric Comparison – IUT Responder – Invalid Public Key Failure MSC – Page 1 of 2

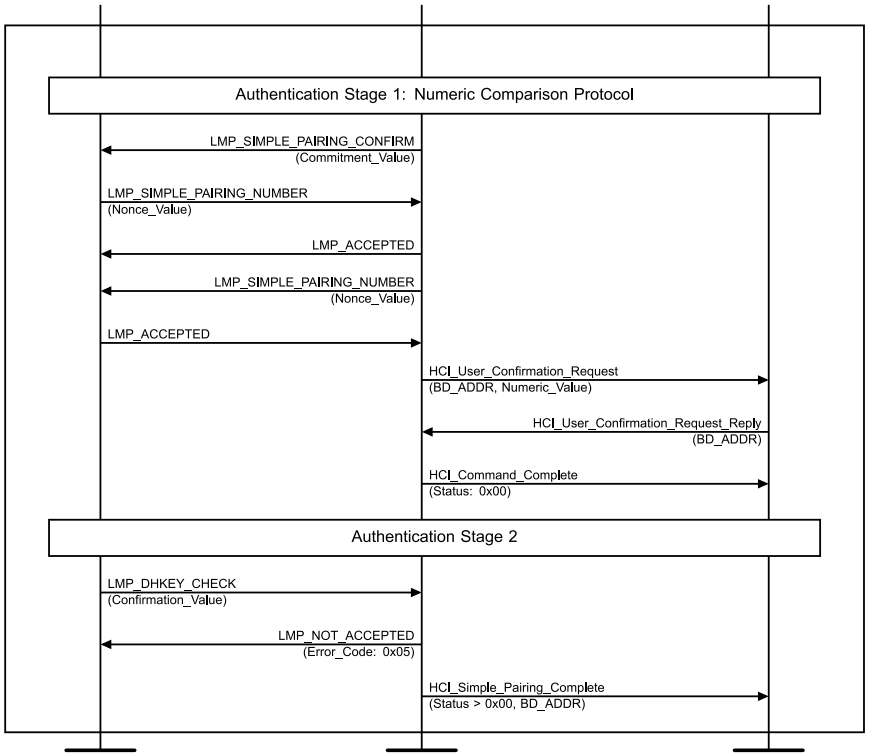


Figure 4.11-89: Numeric Comparison – IUT Responder – Invalid Public Key Failure MSC – Page 2 of 2

Repeat Steps 1–26 for each round in Table 4.11-19 except for round 5.

1. The Upper Tester sends an HCI_Reset command to the IUT and receives a successful HCI_Command_Complete event in response.
2. Execute the “Default settings” preamble if using P-192 or the “Baseband assumptions” and the “Secure Simple Pairing P-256” preambles if using P-256.
3. An ACL connection is established between the IUT and the Lower Tester.
4. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
5. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
7. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
8. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
10. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-20.

11. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the invalid key specified in Table 4.11-19.
12. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
14. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-20.

15. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
16. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
18. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_NUMBER PDU with the Nonce_Value.
19. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
20. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
21. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an HCI_User_Confirmation_Request event to the Upper Tester with the BD_ADDR and Numeric_Value.
23. The Upper Tester sends an HCI_User_Confirmation_Request_Negative_Reply command to the IUT with BD_ADDR and receives a successful HCI_Command_Complete event in response.
24. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
25. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and Error_Code set to 0x05 (Authentication Failure).
26. The IUT sends an unsuccessful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x05 (Authentication Failure).

  In Step 26, the IUT sends an HCI_Simple_Pairing_Complete event with Status not set to Success (0x00).

  If TSPX_new_key_failed_count > 0, a different public key is used by the IUT after at most TSPX_new_key_failed_count pairings.

Inconclusive verdict

The IUT interrupts the pairing process by:

- Sending an LMP_NOT_ACCEPTED PDU with a different Error_Code than 0x05 (Authentication Failure) in response to an LMP_DHKEY_CHECK PDU
- Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_ENCAPSULATED_PAYLOAD PDU containing an invalid public key
- Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_SIMPLE_PAIRING_NUMBER PDU
- Sending an LMP_NUMERIC_COMPARISON_FAILED PDU during the Numeric Comparison protocol phase

- Notes

The test verifies the recommendation of the specification that an IUT should return an LMP_NOT_ACCEPTED PDU to the Lower Tester's LMP_DHKEY_CHECK PDU if the Lower Tester's public key is invalid. Other potentially valid ways of rejecting the invalid key are listed in the expected outcome and will yield an Inconclusive verdict.

To simulate an attacker, the Lower Tester may send an LMP_ACCEPTED PDU in response to all calculated values sent by the IUT, even if the LMP_SIMPLE_PAIRING_CONFIRM, the LMP_SIMPLE_PAIRING_NUMBER, or the LMP_DHKEY_CHECK values sent by the IUT do not match the expected calculations.

4.11.7.3 Passkey Entry – IUT Initiator – Invalid Public Key Failure

- Test Purpose

Verify that the IUT detects an invalid public key using the Passkey Entry protocol and fails the pairing procedure using the key specified in Table 4.11-21.

- Reference

[1] 4.2.7

- Initial Condition
  - The IUT is the Initiator.
  - TSPX_new_key_failed_count gives the number of failed pairing attempts before a new pairing key is generated for Table 4.11-19.
  - The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.

- Test Case Configuration

Test Case	Key
LMP/SP/BI-03-C [Passkey Entry - IUT Initiator – Invalid Public Key Failure, P-192]	P-192
LMP/SP/BI-09-C [Passkey Entry - IUT Initiator – Invalid Public Key Failure, P-256]	P-256

Table 4.11-21: Passkey Entry – IUT Initiator – Invalid Public Key Failure test cases

- Test Procedure

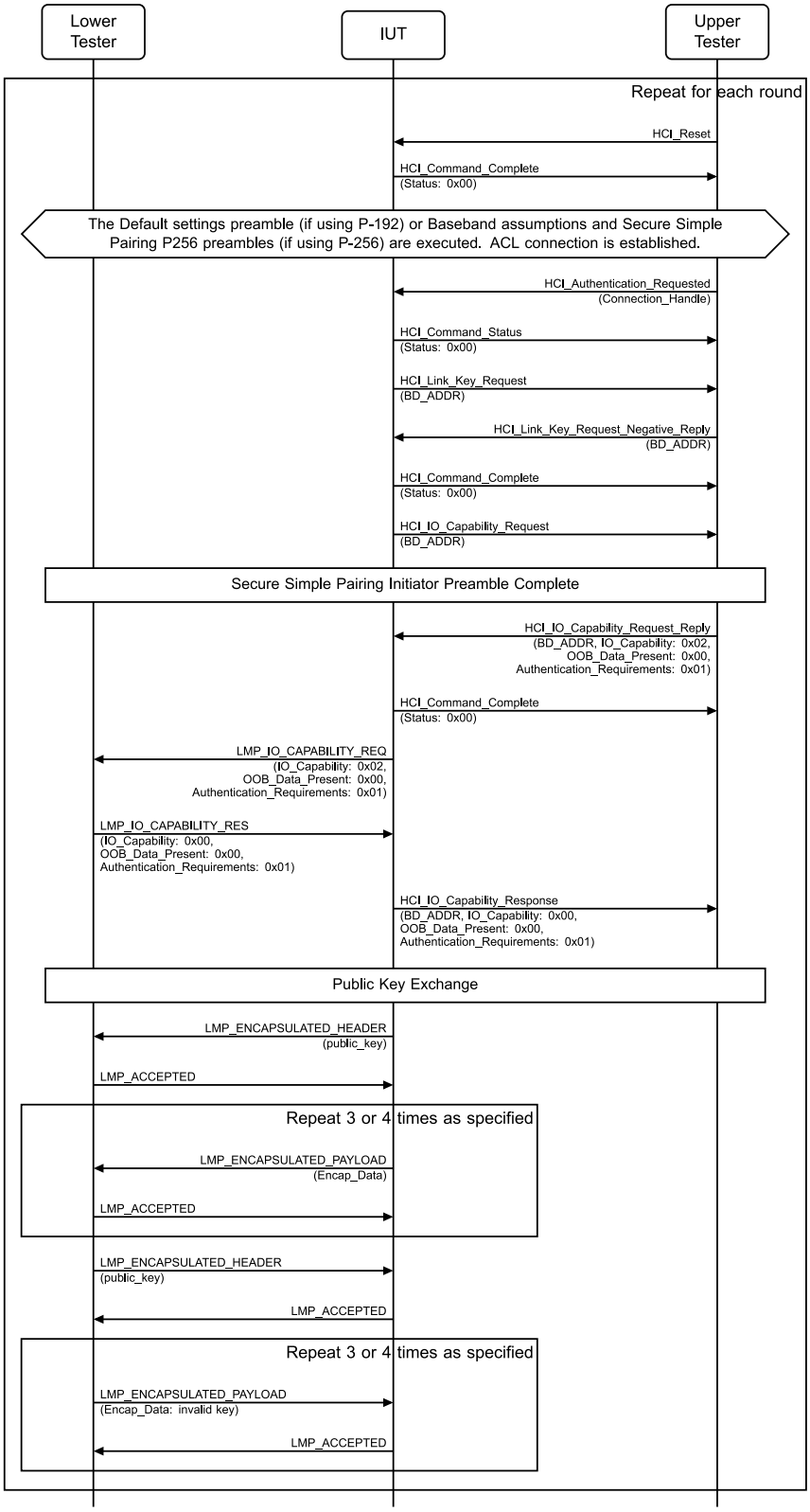


Figure 4.11-90: Passkey Entry – IUT Initiator – Invalid Public Key Failure MSC – Page 1 of 2

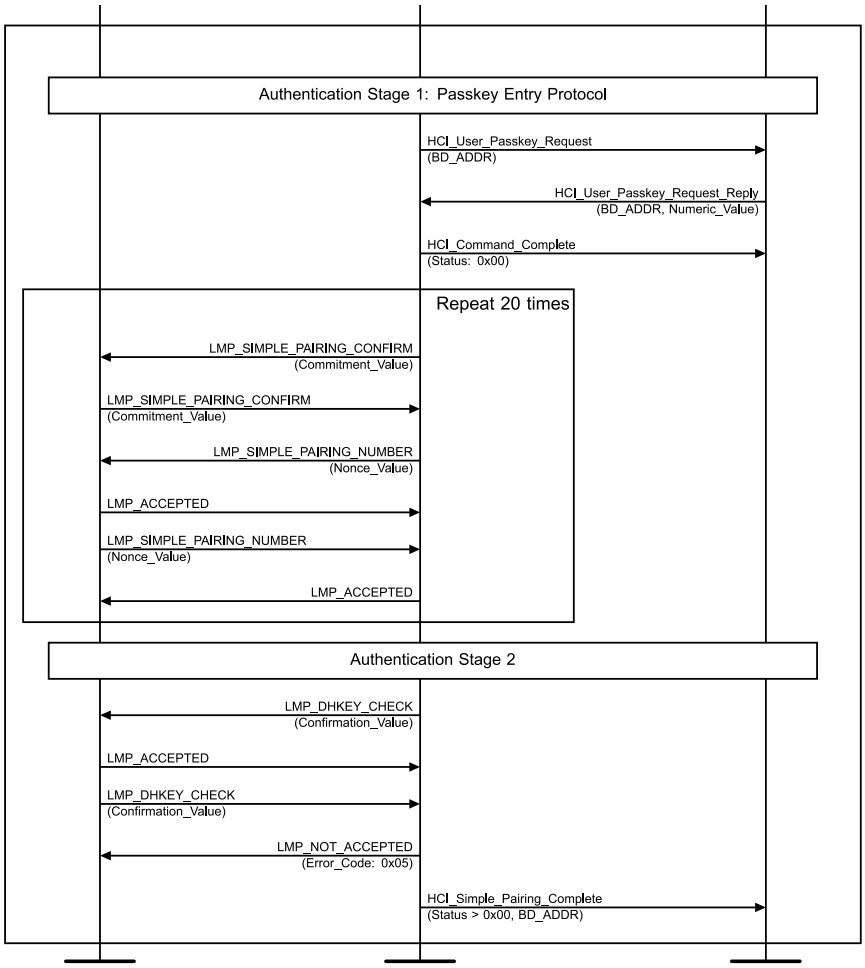


Figure 4.11-91: Passkey Entry – IUT Initiator – Invalid Public Key Failure MSC – Page 2 of 2

Repeat Steps 1–32 for each round and repetition in Table 4.11-19.

1. The Upper Tester sends an HCI_Reset command to the IUT and receives a successful HCI_Command_Complete event in response.
2. Execute the “Default settings” preamble if using P-192 or the “Baseband assumptions” and “Secure Simple Pairing P-256” preambles if using P-256.
3. An ACL connection is established between the IUT and the Lower Tester.
4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
6. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
7. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
8. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
9. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).

10. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
11. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
12. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 14–15 three times if using P-192 or four times if using P-256 as specified in Table 4.11-21.

14. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
15. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
16. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 18–19 three times if using P-192 or four times if using P-256 as specified in Table 4.11-21.

18. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the invalid key specified in Table 4.11-19.
19. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
20. The IUT sends an HCI_User_Passkey_Request event to the Upper Tester with the BD_ADDR.
21. The Upper Tester responds to the IUT with an HCI_User_Passkey_Request_Reply command with the BD_ADDR and Numeric_Value and receives a successful HCI_Command_Complete event in response.

Repeat Steps 22–27 20 times.

22. The IUT sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the Lower Tester with the Commitment_Value.
23. The Lower Tester responds to the IUT with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
24. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
25. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
26. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
27. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
28. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
29. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.

30. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
31. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and Error_Code set to 0x05 (Authentication Failure).
32. The IUT sends an unsuccessful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x05 (Authentication Failure).

  In Step 32, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status not set to Success (0x00).

  If TSPX_new_key_failed_count > 0, a different public key is used by the IUT after at most TSPX_new_key_failed_count pairings.

  Inconclusive verdict

  The IUT interrupts the pairing process by:

  - Sending an LMP_NOT_ACCEPTED PDU with a different Error_Code than 0x05 (Authentication Failure) in response to an LMP_DHKEY_CHECK PDU
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_ENCAPSULATED_PAYLOAD PDU containing an invalid public key
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_SIMPLE_PAIRING_NUMBER PDU
  - Sending an LMP_PASSKEY_ENTRY_FAILED PDU during the Passkey Entry protocol phase

- Notes

  The test verifies the recommendation of the specification that an IUT should return an LMP_NOT_ACCEPTED PDU to the Lower Tester's LMP_DHKEY_CHECK PDU if the Lower Tester's public key is invalid. Other potentially valid ways of rejecting the invalid key are listed in the expected outcome and will yield an Inconclusive verdict.

  To simulate an attacker, the Lower Tester may send an LMP_ACCEPTED PDU in response to all calculated values sent by the IUT, even if the LMP_SIMPLE_PAIRING_CONFIRM, the LMP_SIMPLE_PAIRING_NUMBER, or the LMP_DHKEY_CHECK values sent by the IUT do not match the expected calculations.

4.11.7.4 Passkey Entry – IUT Responder – Invalid Public Key Failure

- Test Purpose

  Verify that the IUT detects an invalid public key using the Passkey Entry protocol and fails the pairing procedure using the key specified in Table 4.11-22.

- Reference

  [1] 4.2.7

- Initial Condition

  - The IUT is the Responder.
  - TSPX_new_key_failed_count gives the number of failed pairing attempts before a new pairing key is generated for Table 4.11-19.
  - The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.

- Test Case Configuration

Test Case	Key	Encapsulated Payload Repeat
LMP/SP/BI-04-C [Passkey Entry – IUT Responder – Invalid Public Key Failure, P-192]	P-192	3
LMP/SP/BI-10-C [Passkey Entry – IUT Responder – Invalid Public Key Failure, P-256]	Secure Simple Pairing P-256 with P-256	4

Table 4.11-22: Passkey Entry – IUT Responder – Invalid Public Key Failure test cases

- Test Procedure

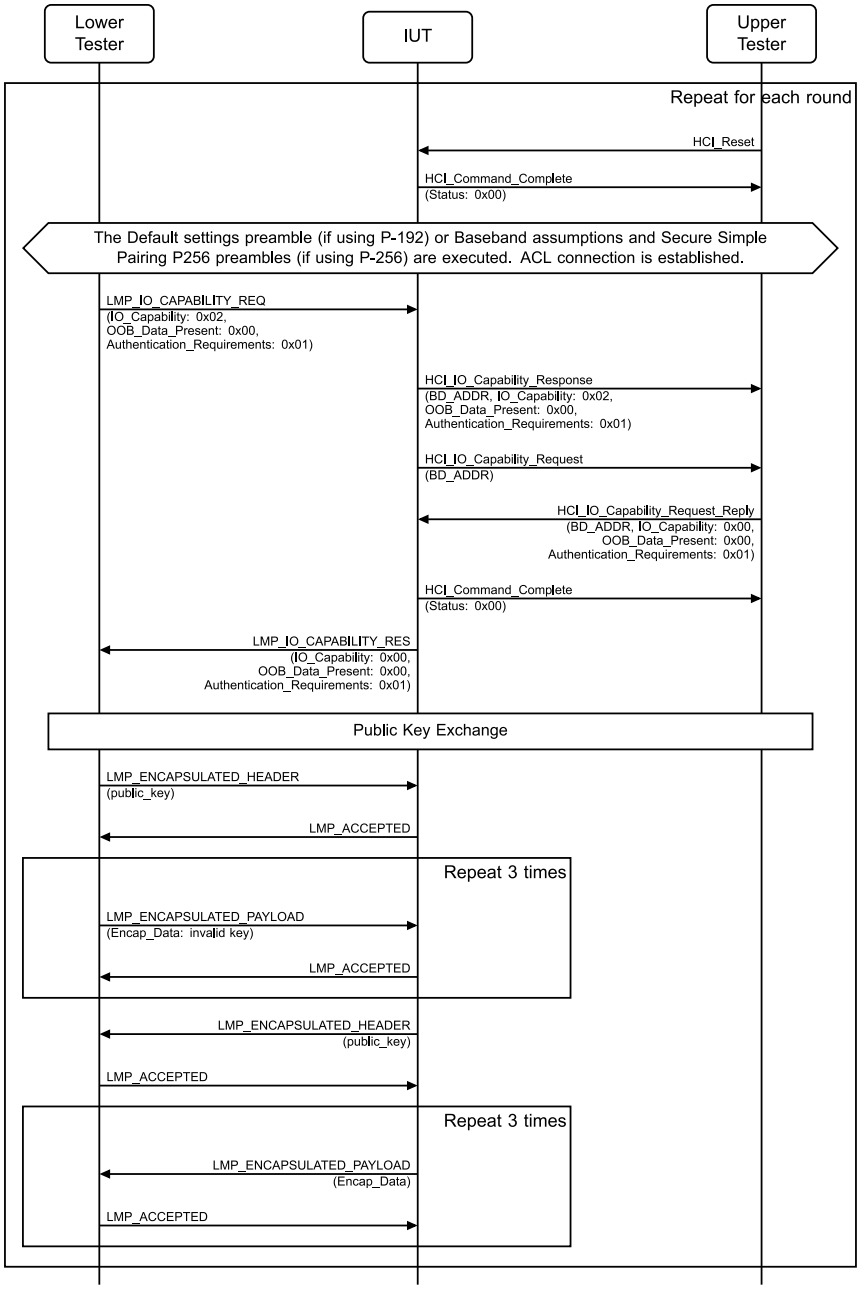


Figure 4.11-92: Passkey Entry – IUT Responder – Invalid Public Key Failure MSC – Page 1 of 2

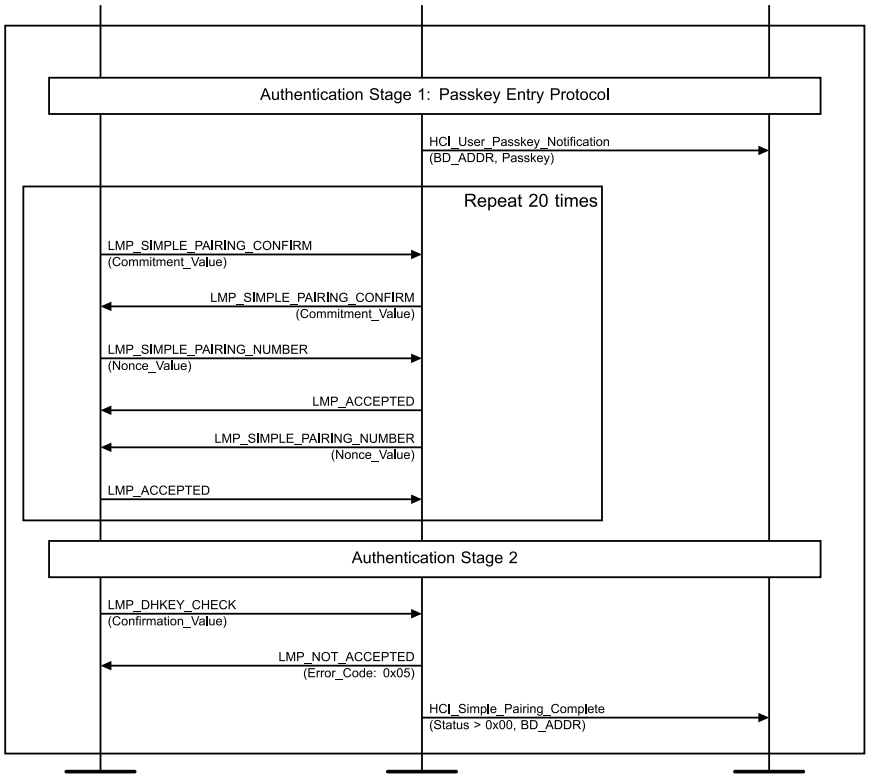


Figure 4.11-93: Passkey Entry – IUT Responder – Invalid Public Key Failure MSC – Page 2 of 2

Repeat Steps 1–26 for each round in Table 4.11-19 except for round 5.

1. The Upper Tester sends an HCI_Reset command to the IUT and receives a successful HCI_Command_Complete event in response.
2. Execute the “Default settings” preamble if using P-192 or the “Baseband assumptions” and “Secure Simple Pairing P-256” preambles if using P-256.
3. An ACL connection is established between the IUT and the Lower Tester.
4. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x02 (KeyboardOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
5. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x02 (KeyboardOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
7. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x00 (DisplayOnly), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
8. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x00 (DisplayOnly), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
10. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-22.

11. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the invalid key specified in Table 4.11-19.
12. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
14. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-22.

15. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
16. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_User_Passkey_Notification event to the Upper Tester with the BD_ADDR and the Passkey.

Repeat Steps 18–23 20 times.

18. The Lower Tester sends an LMP_SIMPLE_PAIRING_CONFIRM PDU to the IUT with the Commitment_Value.
19. The IUT responds to the Lower Tester with an LMP_SIMPLE_PAIRING_CONFIRM PDU with the Commitment_Value.
20. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
21. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
22. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
23. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
24. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
25. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and Error_Code set to 0x05 (Authentication Failure).
26. The IUT sends an unsuccessful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x05 (Authentication Failure).

  In Step 26, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status not set to Success (0x00).

  If TSPX_new_key_failed_count > 0, a different public key is used by the IUT after at most TSPX_new_key_failed_count pairings.

Inconclusive verdict

The IUT interrupts the pairing process by:

- Sending an LMP_NOT_ACCEPTED PDU with a different Error_Code than 0x05 (Authentication Failure) in response to an LMP_DHKEY_CHECK PDU
- Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_ENCAPSULATED_PAYLOAD PDU containing an invalid public key
- Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_SIMPLE_PAIRING_NUMBER PDU
- Sending an LMP_PASSKEY_ENTRY_FAILED PDU during the Passkey Entry protocol phase

- Notes

The test verifies the recommendation of the specification that an IUT should return an LMP_NOT_ACCEPTED PDU to the Lower Tester's LMP_DHKEY_CHECK PDU if the Lower Tester's public key is invalid. Other potentially valid ways of rejecting the invalid key are listed in the expected outcome and will yield an Inconclusive verdict.

To simulate an attacker, the Lower Tester may send an LMP_ACCEPTED PDU in response to all calculated values sent by the IUT, even if the LMP_SIMPLE_PAIRING_CONFIRM, the LMP_SIMPLE_PAIRING_NUMBER, or the LMP_DHKEY_CHECK values sent by the IUT do not match the expected calculations.

4.11.7.5 OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure

- Test Purpose

Verify that the IUT detects an invalid public key using the OOB protocol and fails the pairing procedure using the key specified in Table 4.11-23.

- Reference

[1] 4.2.7

- Initial Condition
  - The IUT is the Initiator.
  - TSPX_new_key_failed_count gives the number of failed pairing attempts before a new pairing key is generated for Table 4.11-19.
  - The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.

- Test Case Configuration

Test Case	Key
LMP/SP/BI-05-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-192]	P-192
LMP/SP/BI-11-C [OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-256]	P-256

Table 4.11-23: OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure test cases

- Test Procedure

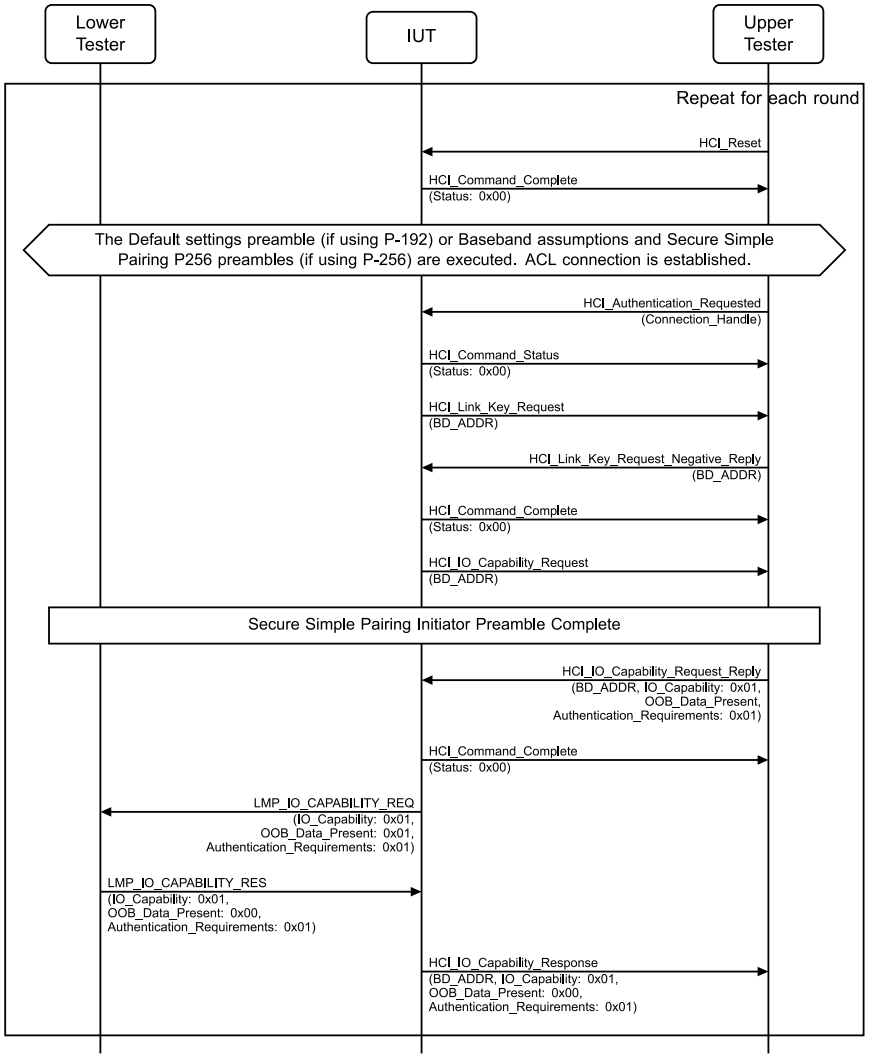


Figure 4.11-94: OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure MSC – Page 1 of 2

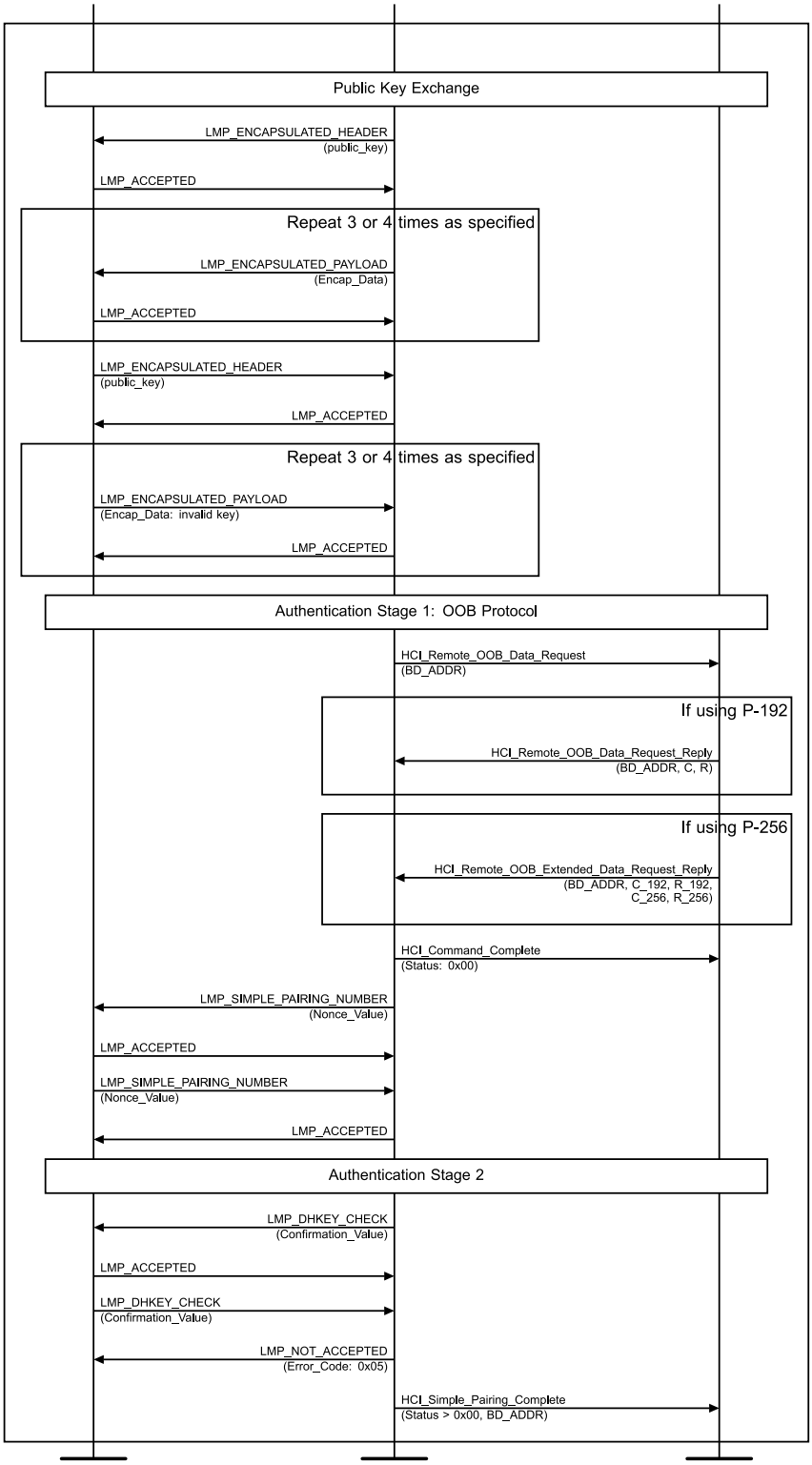


Figure 4.11-95: OOB Protocol – IUT Initiator – IUT with OOB_Auth_Data – Invalid Public Key Failure MSC – Page 2 of 2

Repeat Steps 1–30 for each round and repetition in Table 4.11-19.

1. The Upper Tester sends an HCI_Reset command to the IUT and receives a successful HCI_Command_Complete event in response.
2. Execute the “Default settings” preamble if using P-192 or the “Baseband assumptions” and “Secure Simple Pairing P-256” preambles if using P-256.
3. An ACL connection is established between the IUT and the Lower Tester.
4. The Upper Tester sends an HCI_Authentication_Requested command to the IUT with the Connection_Handle and receives a successful HCI_Command_Status event in return.
5. The IUT sends an HCI_Link_Key_Request event to the Upper Tester with the BD_ADDR.
6. The Upper Tester responds to the IUT with an HCI_Link_Key_Request_Negative_Reply command with the BD_ADDR and receives a successful HCI_Command_Complete event in response.
7. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
8. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x02 (P-256 OOB authentication data from remote device present) if using P-256 as specified in Table 4.11-23, otherwise set to 0x01 (P-192 OOB authentication data from remote device present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.
9. The IUT sends an LMP_IO_CAPABILITY_REQ PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
10. The Lower Tester responds to the IUT with an LMP_IO_CAPABILITY_RES PDU with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
11. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
12. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
13. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 14–15 three times if using P-192 or four times if using P-256 as specified in Table 4.11-23.

14. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
15. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
16. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 18–19 three times if using P-192 or four times if using P-256 as specified in Table 4.11-23.

18. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the invalid key specified in Table 4.11-19.
19. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
20. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.
21. The Upper Tester responds with an HCI_Remote_OOB_Data_Request_Reply command with the BD_ADDR, C, and R if using P-192 or an HCI_Remote_OOB_Extended_Data_Request_Reply command with the BD_ADDR, C_192, R_192, C_256, and R_256 if using P-256 as specified in Table 4.11-23 and receives an HCI_Command_Complete event in response.
22. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
23. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
24. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
25. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
26. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester with the Confirmation_Value.
27. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode.
28. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
29. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and Error_Code set to 0x05 (Authentication Failure).
30. The IUT sends an unsuccessful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x05 (Authentication Failure).

  In Step 30, the IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status not set to Success (0x00).

  If TSPX_new_key_failed_count > 0, a different public key is used by the IUT after at most TSPX_new_key_failed_count pairings.

  Inconclusive verdict

  The IUT interrupts the pairing process by:

  - Sending an LMP_NOT_ACCEPTED PDU with a different Error_Code than 0x05 (Authentication Failure) in response to an LMP_DHKEY_CHECK PDU
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_ENCAPSULATED_PAYLOAD PDU containing an invalid public key
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_SIMPLE_PAIRING_NUMBER PDU
  - Sending an LMP_OOB_FAILED PDU during the OOB protocol phase

- Notes

  The test verifies the recommendation of the specification that an IUT should return an LMP_NOT_ACCEPTED PDU to the Lower Tester's LMP_DHKEY_CHECK PDU if the Lower Tester's public key is invalid. Other potentially valid ways of rejecting the invalid key are listed in the expected outcome and will yield an Inconclusive verdict.

  To simulate an attacker, the Lower Tester may send an LMP_ACCEPTED PDU in response to all calculated values sent by the IUT, even if the LMP_SIMPLE_PAIRING_CONFIRM, the LMP_SIMPLE_PAIRING_NUMBER, or the LMP_DHKEY_CHECK values sent by the IUT do not match the expected calculations.

4.11.7.6 OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure

- Test Purpose

  Verify that the IUT detects an invalid public key using the OOB protocol and fails the pairing procedure using the key specified in Table 4.11-24.

- Reference

  [1] 4.2.7

- Initial Condition

  - The IUT is the Responder.
  - TSPX_new_key_failed_count gives the number of failed pairing attempts before a new pairing key is generated for Table 4.11-19.
  - The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.

- Test Case Configuration

Test Case	Key	Encapsulated Payload Repeat
LMP/SP/BI-06-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-192]	P-192	3
LMP/SP/BI-12-C [OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure, P-256]	Secure Simple Pairing P-256 with P-256	4

Table 4.11-24: OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure test cases

- Test Procedure

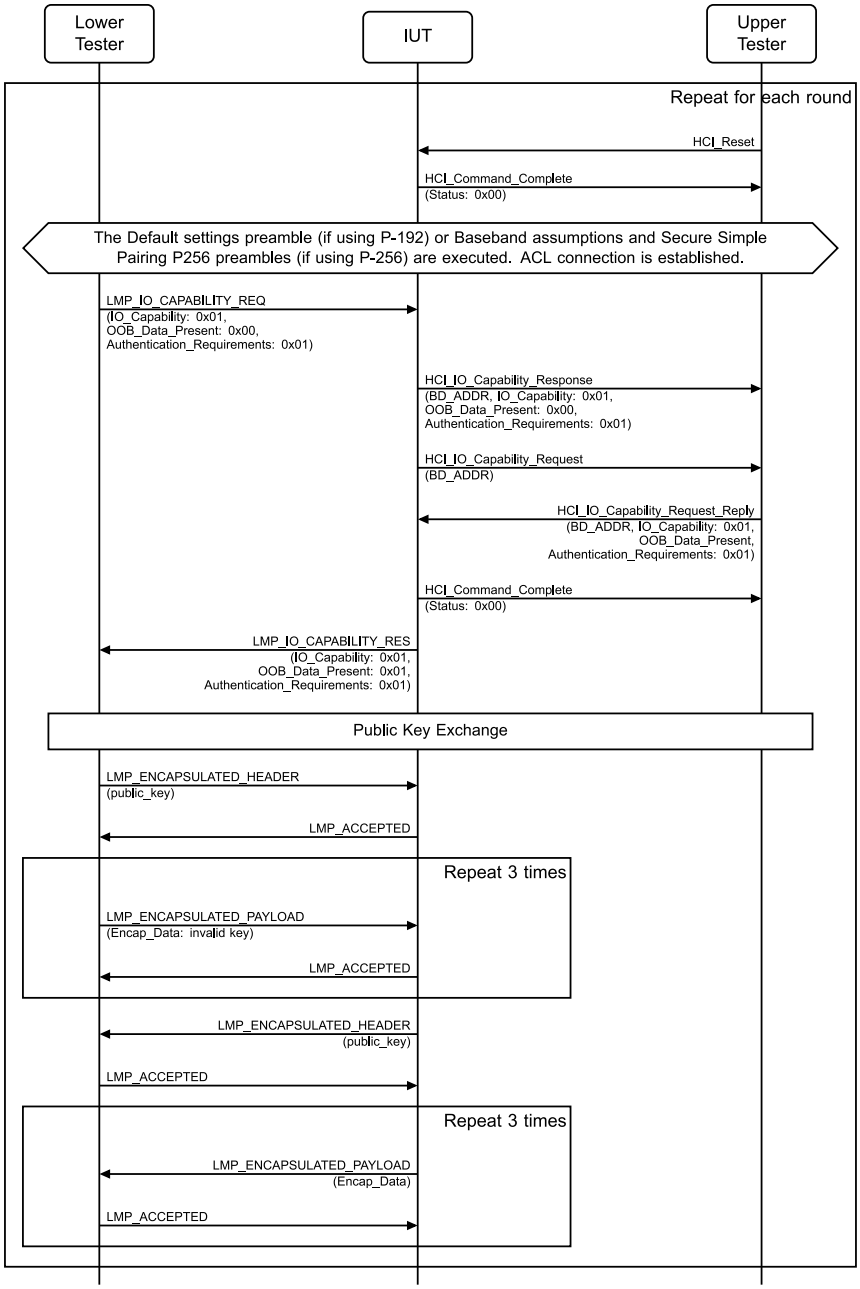


Figure 4.11-96: OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure MSC – Page 1 of 2

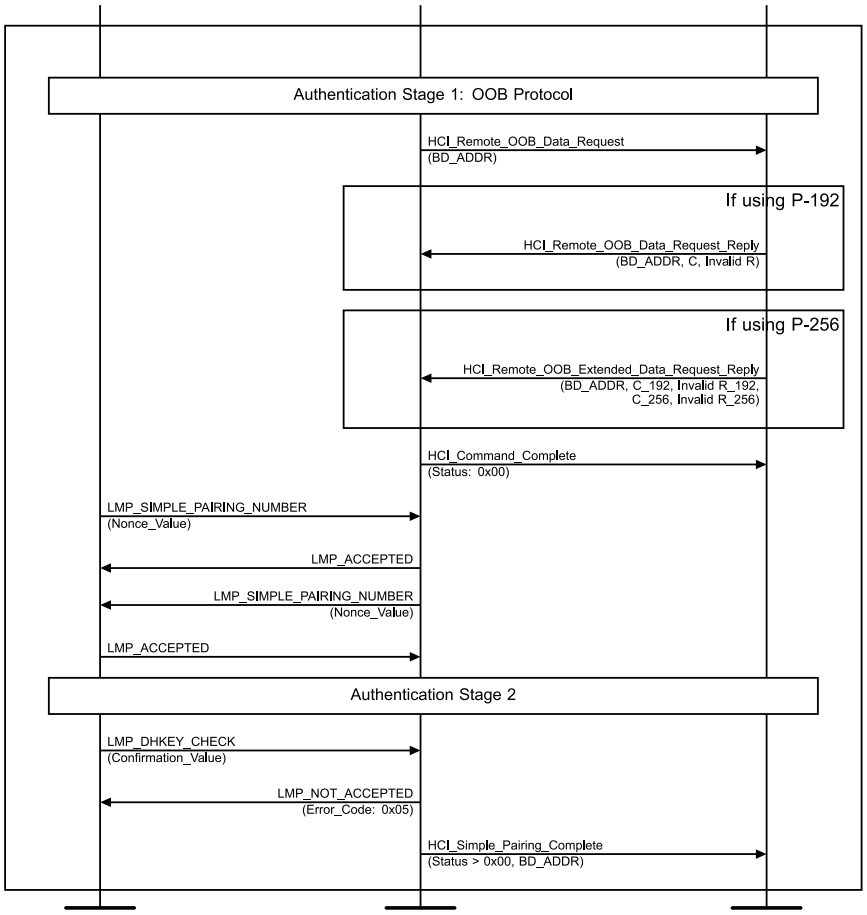


Figure 4.11-97: OOB Protocol – IUT Responder – IUT with OOB_Auth_Data – Invalid Public Key Failure MSC – Page 2 of 2

Repeat Steps 1–25 for each round in Table 4.11-19 except for round 5.

1. The Upper Tester sends an HCI_Reset command to the IUT and receives a successful HCI_Command_Complete event in response.
2. Execute the “Default settings” preamble if using P-192 or the “Baseband assumptions” and “Secure Simple Pairing P-256” preambles if using P-256.
3. An ACL connection is established between the IUT and the Lower Tester.
4. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
5. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
6. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
7. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Reply command with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x02 (P-256 OOB authentication data from remote device present) if using P-256 as specified in Table 4.11-24, otherwise set to 0x01 (P-192 OOB authentication data from remote device present), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding) and receives a successful HCI_Command_Complete event in response.

8. The IUT sends an LMP_IO_CAPABILITY_RES PDU to the Lower Tester with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x01 (OOB Authentication Data received), and Authentication_Requirements set to 0x01 (MITM Protection Required – No Bonding).
9. The Lower Tester sends an LMP_ENCAPSULATED_HEADER PDU to the IUT with the public key.
10. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 11–12 three times if using P-192 or four times if using P-256 as specified in Table 4.11-24.

11. The Lower Tester sends an LMP_ENCAPSULATED_PAYLOAD PDU to the IUT with the invalid key specified in Table 4.11-19.
12. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
13. The IUT sends an LMP_ENCAPSULATED_HEADER PDU to the Lower Tester with the public key.
14. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_HEADER PDU Opcode.

Repeat Steps 15–16 three times if using P-192 or four times if using P-256 as specified in Table 4.11-24.

15. The IUT sends an LMP_ENCAPSULATED_PAYLOAD PDU to the Lower Tester with the Encap_Data.
16. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_ENCAPSULATED_PAYLOAD PDU Opcode.
17. The IUT sends an HCI_Remote_OOB_Data_Request event to the Upper Tester with the BD_ADDR.
18. The Upper Tester responds with an HCI_Remote_OOB_Data_Request_Reply command with the BD_ADDR, C, and R if using P-192 or an HCI_Remote_OOB_Extended_Data_Request_Reply command with the BD_ADDR, C_192, R_192, C_256, and R_256 if using P-256 as specified in Table 4.11-24 and receives an HCI_Command_Complete event in response.
19. The Lower Tester sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the IUT with the Nonce_Value.
20. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
21. The IUT sends an LMP_SIMPLE_PAIRING_NUMBER PDU to the Lower Tester with the Nonce_Value.
22. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SIMPLE_PAIRING_NUMBER PDU Opcode.
23. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT with the Confirmation_Value.
24. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED PDU with the LMP_DHKEY_CHECK PDU Opcode and Error_Code set to 0x05 (Authentication Failure).
25. The IUT sends an unsuccessful HCI_Simple_Pairing_Complete event to the Upper Tester with the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code set to 0x05 (Authentication Failure).

  In Step 25, the IUT sends an HCI_Simple_Pairing_Complete event with Status not set to Success (0x00).

  If TSPX_new_key_failed_count > 0, a different public key is used by the IUT after at most TSPX_new_key_failed_count pairings.

  Inconclusive verdict

  The IUT interrupts the pairing process by:

  - Sending an LMP_NOT_ACCEPTED PDU with a different Error_Code than 0x05 (Authentication Failure) in response to an LMP_DHKEY_CHECK PDU
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_ENCAPSULATED_PAYLOAD PDU containing an invalid public key
  - Sending an LMP_NOT_ACCEPTED PDU in response to an LMP_SIMPLE_PAIRING_NUMBER PDU

- Notes

  The test verifies the recommendation of the specification that an IUT should return an LMP_NOT_ACCEPTED PDU to the Lower Tester's LMP_DHKEY_CHECK PDU if the Lower Tester's public key is invalid. Other potentially valid ways of rejecting the invalid key are listed in the expected outcome and will yield an Inconclusive verdict.

  To simulate an attacker, the Lower Tester may send an LMP_ACCEPTED PDU in response to all calculated values sent by the IUT, even if the LMP_SIMPLE_PAIRING_CONFIRM, the LMP_SIMPLE_PAIRING_NUMBER, or the LMP_DHKEY_CHECK values sent by the IUT do not match the expected calculations.

LMP/SP/BV-66-C [Simple Pairing Capable Controller – Host is in Non-Bondable Mode – IUT Responder]

- Test Purpose

  Verify that the LM on the IUT rejects the Simple Pairing procedure when the Host sends an HCI_IO_Capability_Request_Negative_Reply command.

- Reference

  [1] 4.2.7

- Initial Condition

  - The IUT is the Peripheral, the Claimant, and a Simple Pairing–Capable Controller.
  - The Upper Tester is Simple Pairing–Capable Host.
  - The Lower Tester is the Central, verifier of the pairing procedure, and an SSP-Enabled Initiator.
  - An ACL connection has been established.
  - The IUT is connected to the Lower Tester through LMP_HOST_CONNECTION_REQ and LMP_ACCEPTED.

- The Lower Tester's LMP features include:
  - Feature bit 51 (Secure Simple Pairing) set to 1
  - Feature bit 63 (Extended Features) set to 1
  - Feature bit 64 (Secure Simple Pairing - Host Support) set to 1
- The Upper Tester does not allow bonding.

- Test Procedure

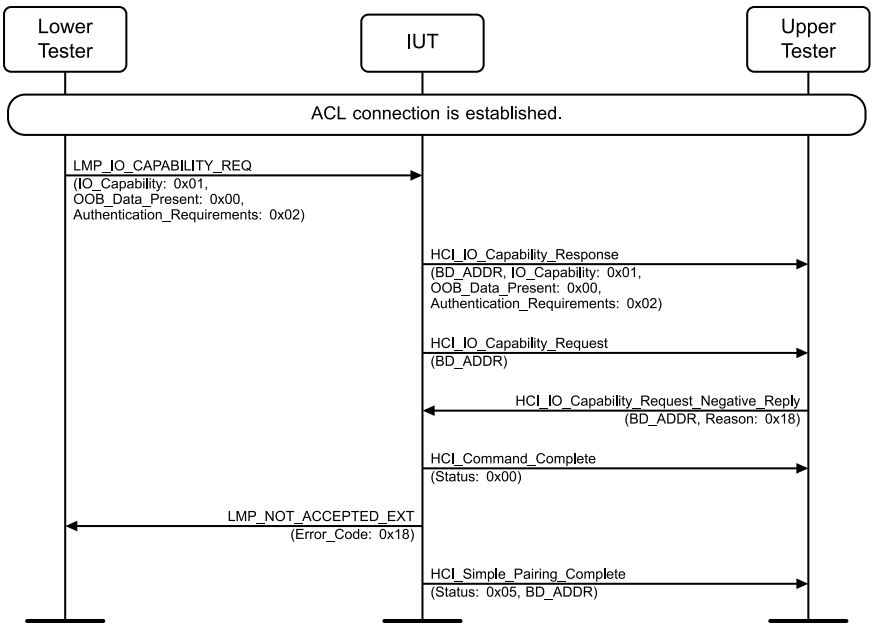


Figure 4.11-98: LMP/SP/BV-66-C [Simple Pairing Capable Controller – Host is in Non-Bondable Mode – IUT Responder] MSC

1. The Lower Tester sends an LMP_IO_CAPABILITY_REQ PDU to the IUT with IO_Capabilities set to 0x01 (Display YesNo), OOB_Auth_Data set to 0x00 (No OOB Authentication Data received), and Authentication_Requirements set to 0x02 (MITM Protection Not Required – Dedicated Bonding).
2. The IUT sends an HCI_IO_Capability_Response event to the Upper Tester with the BD_ADDR, IO_Capability set to 0x01 (Display YesNo), OOB_Data_Present set to 0x00 (OOB authentication data not present), and Authentication_Requirements set to 0x02 (MITM Protection Not Required – Dedicated Bonding).
3. The IUT sends an HCI_IO_Capability_Request event to the Upper Tester with the BD_ADDR.
4. The Upper Tester responds to the IUT with an HCI_IO_Capability_Request_Negative_Reply command with the BD_ADDR and Reason set to 0x18 (Pairing not Allowed) and receives a successful HCI_Command_Complete event in response.
5. The IUT sends an LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with the LMP_IO_CAPABILITY_REQ PDU Extended_Opcode, Escape_Opcode, and Error_Code set to 0x18 (Pairing not Allowed).
6. The IUT sends an HCI_Simple_Pairing_Complete event to the Upper Tester with Status set to 0x05 (Authentication Failure) and the BD_ADDR.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED_EXT PDU to the Lower Tester with Error_Code set to 0x18 (Pairing not Allowed) after it has received the LMP_IO_CAPABILITY_REQ PDU from the Lower Tester.

LMP/SP/BI-13-C [Authentication Stage 2, Invalid DHKey Check, Responder]

- Test Purpose

  Verify that the IUT handles an LMP_DHKEY_CHECK using an invalid DHKey as Responder.

- Reference

  [1] 4.2.7.4

- Initial Condition

  - The IUT and the Lower Tester have completed Authentication Stage 1 with an invalid Public Key from the Lower Tester.
  - The IUT is the Responder and the Lower Tester is the Initiator.

- Test Procedure

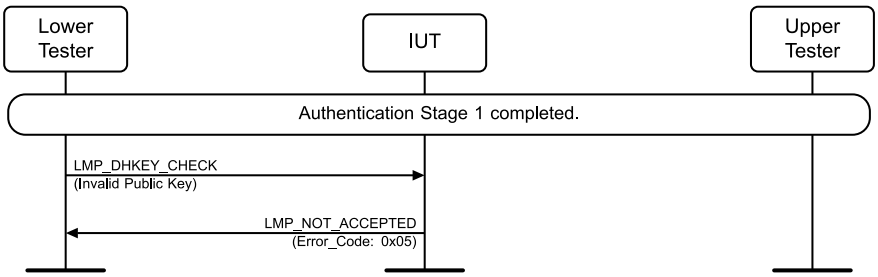


Figure 4.11-99: LMP/SP/BI-13-C [Authentication Stage 2, Invalid DHKey Check, Responder] MSC

1. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT using the invalid Public Key.
2. The IUT sends an LMP_NOT_ACCEPTED PDU with Error_Code set to Authentication Failure (0x05).

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code Authentication Failure (0x05).

LMP/SP/BI-14-C [Authentication Stage 2, Invalid DHKey Check, Initiator]

- Test Purpose

  Verify that the IUT handles an LMP_DHKEY_CHECK PDU using an invalid DHKey as Initiator.

- Reference

  [1] 4.2.7.4

- Initial Condition
  - The IUT and the Lower Tester have completed Authentication Stage 1 with an invalid Public Key from the Lower Tester.
  - The IUT is the Initiator and the Lower Tester is the Responder.
- Test Procedure

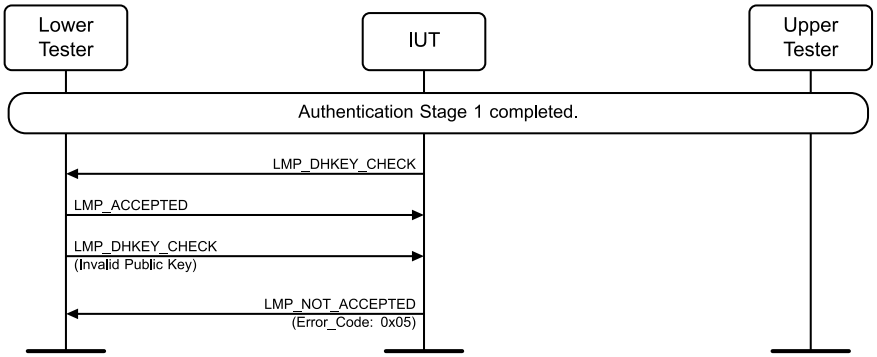


Figure 4.11-100: LMP/SP/BI-14-C [Authentication Stage 2, Invalid DHKey Check, Initiator] MSC

1. The IUT sends an LMP_DHKEY_CHECK PDU to the Lower Tester.
2. The Lower Tester sends an LMP_ACCEPTED PDU to the IUT.
3. The Lower Tester sends an LMP_DHKEY_CHECK PDU to the IUT using the invalid Public Key.
4. The IUT sends an LMP_NOT_ACCEPTED PDU with Error_Code set to Authentication Failure (0x05).

- Expected Outcome

  Pass verdict

  The IUT responds to the Lower Tester's LMP_DHKEY_CHECK PDU with an LMP_NOT_ACCEPTED PDU with Error_Code Authentication Failure (0x05).

LMP/SP/BV-69-C [Simple Pairing Capable Controller – Reject Secure Simple Pairing PDUs When SSP is Disabled]

- Test Purpose

  Verify that the LM on the IUT rejects Simple Pairing procedures when Secure Simple Pairing is disabled.

- Reference

  [1] 4.2.7

- Initial Condition
  - The IUT is the Peripheral, Claimant, and a Simple Pairing Capable Controller.
  - The Upper Tester is a Simple Pairing Capable Host.
  - The Lower Tester is the Central, verifier of the pairing procedure, and an SSP Enabled Initiator.
  - An ACL connection has been established.
  - The IUT is connected to the Lower Tester through LMP_HOST_CONNECTION_REQ and LMP_ACCEPTED.

- The Lower Tester's LMP features include:
  - Feature bit 51 (Secure Simple Pairing) set to 1
  - Feature bit 63 (Extended Features) set to 1
  - Feature bit 64 (Secure Simple Pairing - Host Support) set to 1
- The Upper Tester does not allow bonding.

- Test Procedure

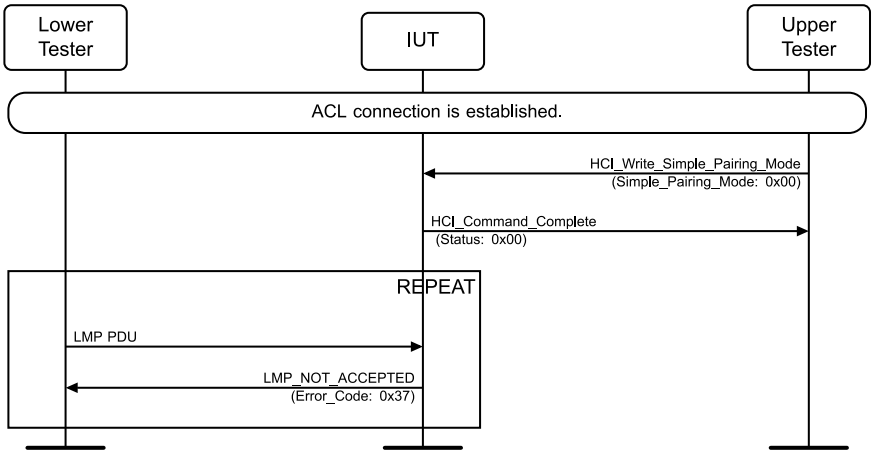


Figure 4.11-101: LMP/SP/BV-69-C [Simple Pairing Capable Controller – Reject Secure Simple Pairing PDUs When SSP is Disabled] MSC

1. The Upper Tester sends an HCI_Write_Simple_Pairing_Mode command to the IUT with Simple_Pairing_Mode set to 0x00 and receives a successful HCI_Command_Complete in response.

Repeat Steps 2 and 3 for each round in Table 4.11-25.

2. The Lower Tester sends the LMP PDU specified in Table 4.11-25 to the IUT.
3. The IUT sends an LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x37 (Secure Simple Pairing not Supported by Host).

Round	LMP PDU
1	LMP_IO_CAPABILITY_REQ
2	LMP_SIMPLE_PAIRING_CONFIRM
3	LMP_SIMPLE_PAIRING_NUMBER
4	LMP_DHKEY_CHECK
5	LMP_KEYPRESS_NOTIFICATION

Table 4.11-25: LMP/SP/BV-69-C [Simple Pairing Capable Controller – Reject Secure Simple Pairing PDUs When SSP is Disabled] rounds

- Expected Outcome

Pass verdict

In Step 3, the IUT sends a 0x37 error code.

4.12 Piconet Clock Adjust

4.12.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.12.2 Piconet Clock Adjust test cases

LMP/XCL/BV-01-C [Central Initiates Coarse Clock Adjustment]

- Test Purpose

  Verify that the IUT as the Central will correctly initiate a Coarse Clock Adjustment.

  This test can only be performed conclusively on a device that supports the MWS Coexistence Logical Signaling Specification or provides an alternative mechanism for triggering a clock adjustment.

- Reference

  [1] Vol. 2, Part C, 4.1.14.1

  [1] Vol. 7, Part A, 2.1

- Initial Condition
  - The Lower Tester is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - The Upper Tester is configured to issue MWS FRAME_SYNC.
  - The IUT is configured as the Central in the CONNECTION state (active mode, ACL link).
- Test Procedure

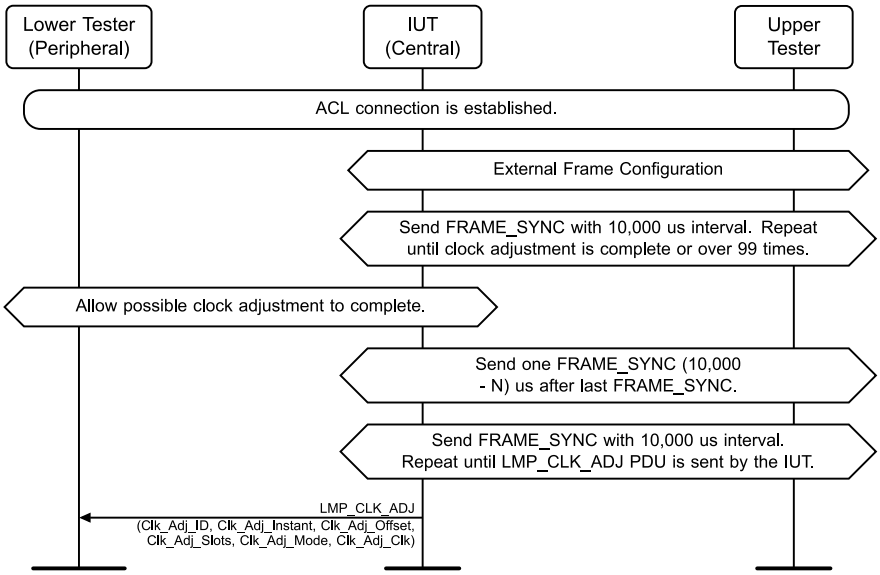


Figure 4.12-1: LMP/XCL/BV-01-C [Central Initiates Coarse Clock Adjustment] MSC

1. The Upper Tester starts issuing FRAME_SYNC with an exact 10 ms interval until clock adjustment completes or over 99 FRAME_SYNCs have been sent.
2. If Step 1 leads to a Coarse Clock Adjustment, wait for clock adjustment to complete.
3. The Upper Tester sends one FRAME_SYNC (10,000 – N) µs after the previous one where N is 100 ≤ N ≤ 300.
4. The Upper Tester continues sending FRAME_SYNC with 10 ms intervals until the IUT sends the LMP_CLK_ADJ PDU to the Lower Tester.
5. The IUT sends the LMP_CLK_ADJ PDU to the Lower Tester with the Clk_Adj_ID, Clk_Adj_Instant, Clk_Adj_Offset, Clk_Adj_Slots, Clk_Adj_Mode, and Clk_Adj_Clk.

- Expected Outcome

  Pass verdict

  The value of Clk_Adj_Offset in the LMP_CLK_ADJ PDU sent by the IUT is N µs ± 5%.

  The LMP_CLK_ADJ is transmitted as broadcast at least six times over APB-C link.

  The IUT enabled AFH as part of the connection establishment and kept it enabled throughout the test. Throughout the test, channels 0, 24, and 78 were marked as unused in the AFH_Channel_Map. If the IUT performed a Coarse Clock Adjustment, Clk_Adj_Instant = CLKp + X, where CLKp is CLK of the first LMP_CLK_ADJ packet, and X is ≥ 12 slots and < 12 hours.

  Inconclusive verdict

  The IUT does not support the HCI_Set_External_Frame_Configuration command and does not provide an alternative mechanism for triggering a clock adjustment.

LMP/XCL/BV-02-C [Peripheral Coarse Clock Adjustment Request]

- Test Purpose

  Verify that the IUT as the Peripheral will correctly initiate a Coarse Clock Adjustment request. The Lower Tester will accept the request.

  This test can only be performed conclusively on a device that supports the MWS Coexistence Logical Signaling Specification.

- Reference

  [1] Vol. 2, Part C, 4.1.14.2

  [1] Vol. 7, Part A, 2.1

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The Bluetooth clock of the Lower Tester is chosen to include clock wrap-around (2^{27}–1 to 0) during the test procedure.
  - The Upper Tester is configured to issue MWS FRAME_SYNC.
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).

- Test Procedure

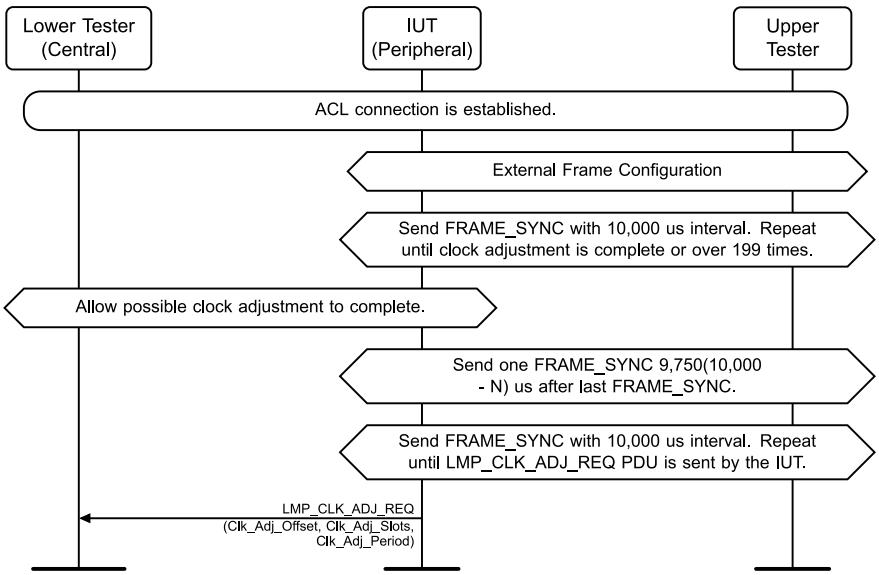


Figure 4.12-2: LMP/XCL/BV-02-C [Peripheral Coarse Clock Adjustment Request] MSC

1. The Upper Tester starts issuing FRAME_SYNC with an exact 10 ms interval until clock adjustment completes or over 199 FRAME_SYNCs have been sent.
2. If Step 1 leads to a Coarse Clock Adjustment request, wait for clock adjustment to complete.
3. The Upper Tester sends one FRAME_SYNC (10,000 – N) µs after the previous one where N is 100 ≤ N ≤ 300.
4. The Upper Tester continues sending FRAME_SYNC with 10 ms intervals until the IUT sends the LMP_CLK_ADJ_REQ PDU to the Lower Tester.
5. The IUT sends the LMP_CLK_ADJ_REQ PDU to the Lower Tester with Clk_Adj_Offset, Clk_Adj_Slots, and Clk_Adj_Period.

- Expected Outcome

  Pass verdict

  The value of Clk_Adj_Offset in the LMP_CLK_ADJ_REQ PDU sent by the IUT is N µs ± 5%.

  Inconclusive verdict

  The IUT does not support the HCI_Set_External_Frame_Configuration command and does not provide an alternative mechanism for triggering a clock adjustment.

LMP/XCL/BV-03-C [Test that Central does not reuse Clk_Adj_ID within LSTO]

- Test Procedure

  Verify that the IUT as the Central will not reuse the same Clk_Adj_ID within the longest LSTO of all connected Peripherals.

- Reference

  [1] Vol. 2, Part C, 4.1.14.1

- Initial Condition
  - The Lower Tester is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Central in the CONNECTION state (active mode, ACL link).

- Test Procedure

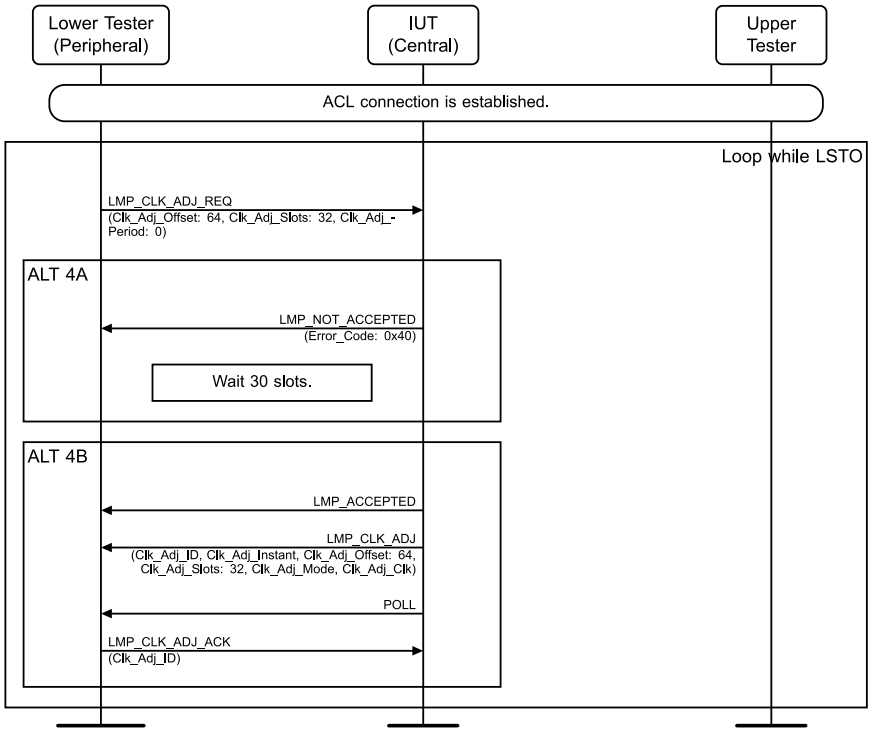


Figure 4.12-3: LMP/XCL/BV-03-C [Test that Central does not reuse Clk_Adj_ID within LSTO] MSC

1. Set N = 32.
2. Configure the Lower Tester to issue LMP_CLK_ADJ_REQ PDU and handle responses for a period of its LSTO.
3. The Lower Tester sends an LMP_CLK_ADJ_REQ PDU to the IUT with Clk_Adj_Offset set to 64, Clk_Adj_Slots set to 32, and Clk_Adj_Period set to 0.
4. Perform either alternative 4A or 4B depending on the IUT's response.

   Alternative 4A (The IUT sends the LMP_NOT_ACCEPTED PDU):

   4A.1. The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with the LMP_CLK_ADJ_REQ PDU Opcode and Error_Code set to 0x40 (Coarse Clock Adjustment Rejected but Will Try to Adjust Using Clock Dragging).

   4A.2. The Lower Tester waits 30 slots, increments N by 1, and then starts over from Step 3.

   Alternative 4B (The IUT sends the LMP_ACCEPTED PDU):

   4B.1. The IUT sends an LMP_ACCEPTED PDU to the Lower Tester with the LMP_CLK_ADJ_REQ PDU Opcode.

   4B.2. The IUT broadcasts the LMP_CLK_ADJ PDU to the Lower Tester on an APB-C link with the Clk_Adj_ID, Clk_Adj_Instant, Clk_Adj_Offset set to 64, Clk_Adj_Slots set to 32, Clk_Adj_Mode, and Clk_Adj_Clk. The Lower Tester stores the value of the Clk_Adj_ID parameter.

   4B.3. The IUT sends a POLL packet to the Lower Tester.

   4B.4. The Lower Tester sends an LMP_CLK_ADJ_ACK PDU with Clk_Adj_ID set to the same Clk_Adj_ID in Step 5.
5. Repeat Steps 3–4 while less time than LSTO has passed since the start of the test.

- Expected Outcome

  Pass verdict

  In Step 5, the IUT does not reuse the same Clk_Adj_ID value during the test.

  The IUT enabled AFH as part of the connection establishment and kept it enabled throughout the test. Throughout the test, channels 0, 24, and 78 were marked as unused in the AFH_Channel_Map.

  If the IUT performed a Coarse Clock Adjustment, Clk_Adj_Instant = CLKp + X, where CLKp is CLK of the first LMP_CLK_ADJ packet, and X is ≥ 12 slots and < 12 hours. Also, LMP_CLK_ADJ is transmitted as broadcast over APB-C link.

  Fail verdict

  The IUT rejects the LMP_CLK_ADJ_REQ PDU with Error_Code not set to 0x40 (Coarse Clock Adjustment Rejected but Will Try to Adjust Using Clock Dragging) before the 256th request.

  The IUT reuses a Clk_Adj_ID value in the LMP_CLK_ADJ PDU.

- Notes

  In the event that the timing of LSTO and the duration of a complete Coarse Clock Adjustment makes it possible to perform 256 iterations or more, it is unavoidable that at least one value will be repeated. In this case, the Central may delay the Clk_Adj_Instant of the LMP_CLK_ADJ with the duplicate value to beyond the time of the longest LSTO of the piconet after the previous instant where the same value was used. It may also solve the problem by dragging the last request or by rejecting the last request altogether.

LMP/XCL/BV-04-C [Test that a Clock Adjust is ignored on ACL-C]

- Test Procedure

  Verify that the IUT as the Peripheral will ignore a Clock Adjust sent on an ACL-C link.

- Reference

  [1] Vol. 2, Part C, 4.1.14.1, 5.1

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).

- Test Procedure

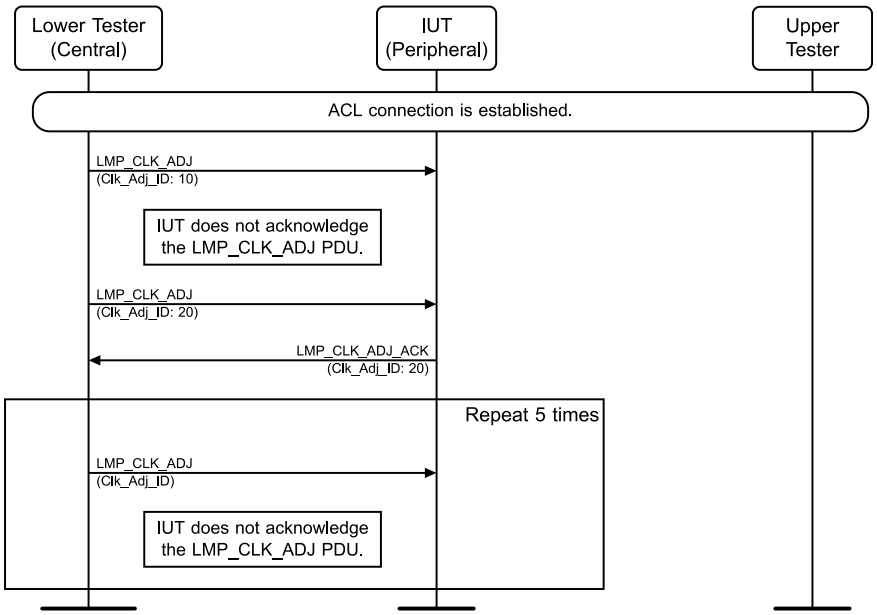


Figure 4.12-4: LMP/XCL/BV-04-C [Test that a Clock Adjust is ignored on ACL-C] MSC

1. The Lower Tester sends an LMP_CLK_ADJ PDU on an ACL-C link with Clk_Adj_ID set to 10.
2. The IUT does not respond with an LMP_CLK_ADJ_ACK PDU.
3. The Lower Tester sends an LMP_CLK_ADJ PDU on an APB-C link with Clk_Adj_ID set to 20.
4. The IUT sends an LMP_CLK_ADJ_ACK PDU to the Lower Tester with Clk_Adj_ID set to 20.
5. The Lower Tester sends an LMP_CLK_ADJ PDU on an ACL-C link with Clk_Adj_ID set to a different value than Steps 1 and 3.
6. The IUT does not respond with an LMP_CLK_ADJ_ACK PDU.
7. Repeat Steps 5 and 6 four times.

- Expected Outcome

  Pass verdict

  In Steps 2 and 6, the IUT does not respond to the LMP_CLK_ADJ PDU.

4.13 Slot Availability Mask

4.13.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

4.13.2 Slot Availability Mask test cases

LMP/SAM/BV-01-C [Respond to three SAM instances]

- Test Purpose

  Verify that the IUT accepts a SAM type 0 submap configuration and SAM slot map define sequences for three SAM instances initiated by the Lower Tester and correctly responds to a SAM switch sequence. Also, verify that the timing control flags bits 0 and 2 in the LMP_SAM_SWITCH PDU sent by the Lower Tester are ignored by the IUT.

- Reference

  [10] 4.1.15

- Initial Condition
  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM is disabled on the Lower Tester and the IUT.
- Test Procedure

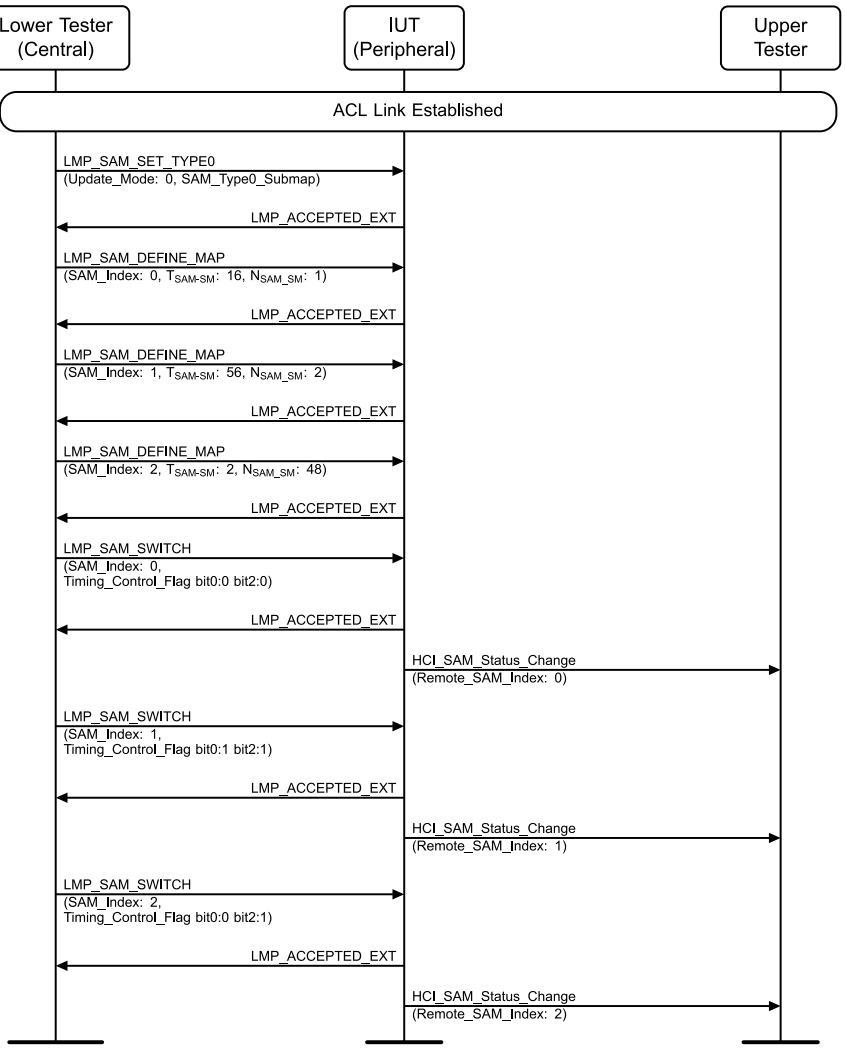


Figure 4.1: LMP/SAM/BV-01-C [Respond to three SAM instances] MSC

1. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU to the IUT with the following parameters:
   - Update_Mode = 0
   - SAM_Type0_Submap = 0x4A, 0x55, 0xBD, 0xAA, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.

3. The Lower Tester sends the first LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters:
   - SAM_Index = 0
   - T_{SAM-SM} = 16
   - N_{SAM_SM} = 1
   - SAM_Submaps = 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
5. The Lower Tester sends the second LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters:
   - SAM_Index = 1
   - T_{SAM-SM} = 56
   - N_{SAM_SM} = 2
   - SAM_Submaps = 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
7. The Lower Tester sends the third LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters:
   - SAM_Index = 2
   - T_{SAM-SM} = 2
   - N_{SAM_SM} = 48
   - SAM_Submaps = 0x09, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55, 0x55
8. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
9. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with SAM_Index = 0, Timing_Control_Flag bit 1 as determined by CLK_{27} of the Central, and bits 0 and 2 set to 0.
10. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
11. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
12. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with SAM_Index = 1, Timing_Control_Flag bit 1 as determined by CLK_{27} of the Central, and bits 0 and 2 set to 1 and receives an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode in response.
13. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
14. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with SAM_Index = 2, Timing_Control_Flag bit 1 as determined by CLK_{27} of the Central, bit 0 set to 0, and bit 2 set to 1.
15. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode in response.
16. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT accepts the SAM type 0 submap configuration and SAM slot map define sequences for three SAM instances initiated by the Lower Tester correctly.

  The IUT responds to SAM switch sequences initiated by the Lower Tester and generates HCI_SAM_Status_Change events correctly.

  In Step 10, the first HCI_SAM_Status_Change event is generated with the following parameters:

  - Connection_handle
  - Local_SAM_Index = 0xFF
  - Local_SAM_TX_Availability = 0xFF
  - Local_SAM_RX_Availability = 0xFF
  - Remote_SAM_Index = 0x00
  - Remote_SAM_TX_Availability = 0x7F
  - Remote_SAM_RX_Availability = 0x8F

  In Step 12, the second HCI_SAM_Status_Change event is generated with the following parameters:

  - Connection_handle
  - Local_SAM_Index = 0xFF
  - Local_SAM_TX_Availability = 0xFF
  - Local_SAM_RX_Availability = 0xFF
  - Remote_SAM_Index = 0x01
  - Remote_SAM_TX_Availability = 0x7F
  - Remote_SAM_RX_Availability = 0x7F

  In Step 14, the third HCI_SAM_Status_Change event is generated with the following parameters:

  - Connection_handle
  - Local_SAM_Index = 0xFF
  - Local_SAM_TX_Availability = 0xFF
  - Local_SAM_RX_Availability = 0xFF
  - Remote_SAM_Index = 0x02
  - Remote_SAM_TX_Availability = 0xEF
  - Remote_SAM_RX_Availability = 0xF9

LMP/SAM/BV-02-C [Initiate three SAM instances]

- Test Purpose

  Verify that the IUT will correctly initiate a SAM type 0 submap configuration and SAM slot map define sequences for three SAM instances and can correctly initiate SAM switch sequence.

- Reference

  [10] 4.1.15

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM disabled on the Lower Tester and the IUT.
  - SAM negotiations are triggered by the IUT with predefined parameters.

- Test Procedure

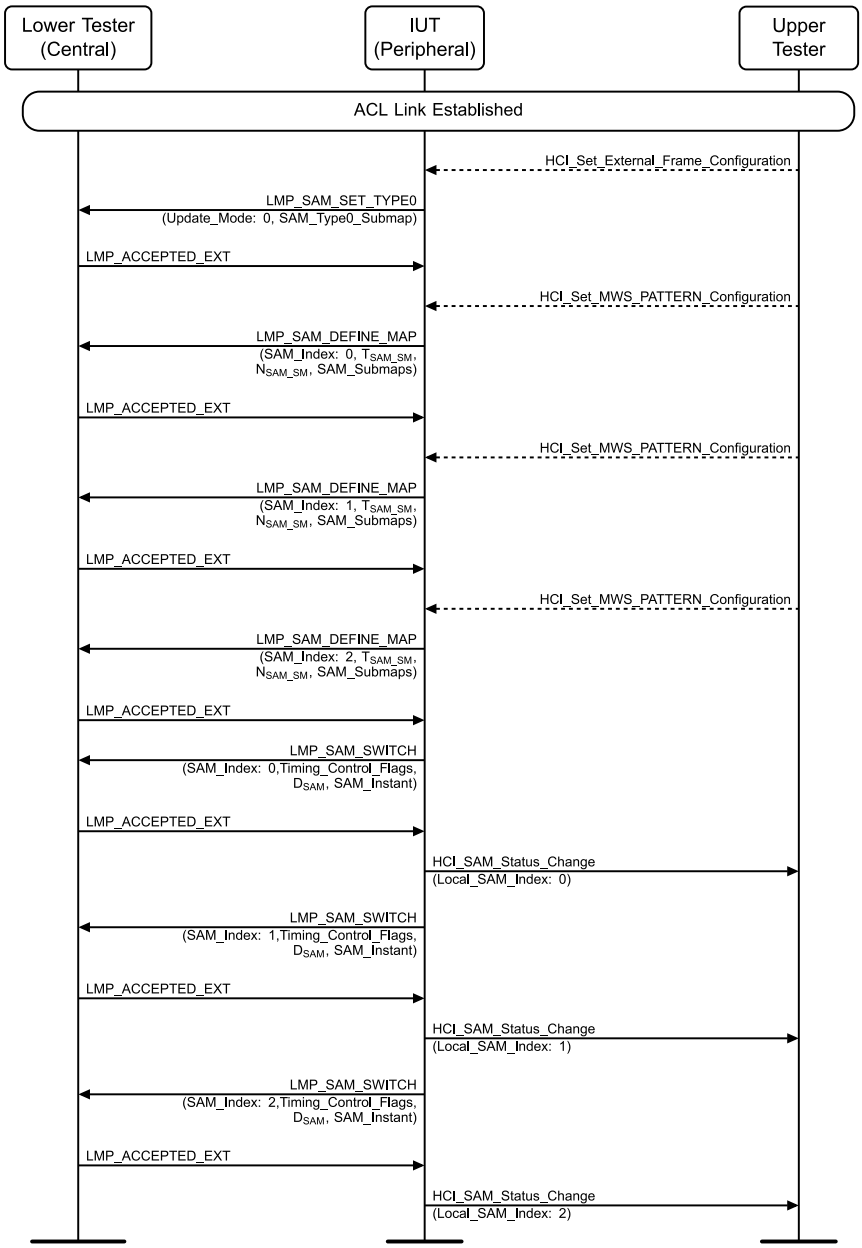


Figure 4.2: LMP/SAM/BV-02-C [Initiate three SAM instances] MSC

1. If necessary, the Upper Tester sends an HCI_Set_External_Frame_Configuration command to the IUT with MWS_Frame_Duration, MWS_Frame_Sync_Assert_Offset, MWS_Frame_Sync_Assert_Jitter, MWS_Num_Periods, Period_Duration, and Period_Type.
2. The IUT sends an LMP_SAM_SET_TYPE0 PDU to the Lower Tester with the Update_Mode and SAM_Type0_Submap to configure a type 0 submap.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
4. If necessary, the Upper Tester sends an HCI_Set_MWS_PATTERN_Configuration command to the IUT with MWS_Pattern_Index, MWS_Pattern_Num_Intervals, MWS_Pattern_Interval_Duration, and MWS_Pattern_Interval_Type.
5. The IUT sends an LMP_SAM_DEFINE_MAP PDU to the Lower Tester with SAM_Index set to 0, T_{SAM_SM}, N_{SAM_SM}, and SAM_Submaps.
6. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
7. If necessary, the Upper Tester sends an HCI_Set_MWS_PATTERN_Configuration command to the IUT with MWS_Pattern_Index, MWS_Pattern_Num_Intervals, MWS_Pattern_Interval_Duration, and MWS_Pattern_Interval_Type.
8. The IUT sends an LMP_SAM_DEFINE_MAP PDU to the Lower Tester with SAM_Index set to 1, T_{SAM_SM}, N_{SAM_SM}, and SAM_Submaps.
9. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
10. If necessary, the Upper Tester sends an HCI_Set_MWS_PATTERN_Configuration command to the IUT with MWS_Pattern_Index, MWS_Pattern_Num_Intervals, MWS_Pattern_Interval_Duration, and MWS_Pattern_Interval_Type.
11. The IUT sends an LMP_SAM_DEFINE_MAP PDU to the Lower Tester with SAM_Index set to 2, T_{SAM_SM}, N_{SAM_SM}, and SAM_Submaps.
12. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
13. The IUT sends an LMP_SAM_SWITCH PDU to the Lower Tester with SAM_Index set to 0, Timing_Control_Flags, D_{SAM}, and SAM_Instant.
14. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
15. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index set to 0, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
16. The IUT sends an LMP_SAM_SWITCH PDU to the Lower Tester with SAM_Index set to 1, Timing_Control_Flags, D_{SAM}, and SAM_Instant.
17. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
18. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index set to 1, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
19. The IUT sends an LMP_SAM_SWITCH PDU to the Lower Tester with SAM_Index set to 2, Timing_Control_Flags, D_{SAM}, and SAM_Instant.
20. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.

21. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index set to 2, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT correctly initiates SAM type 0 submap configuration and SAM slot map define sequences for three SAM instances and correctly initiates the SAM switch sequence.

  The IUT generates the HCI_SAM_Status_Change events correctly.

LMP/SAM/BI-03-C [Respond to invalid SAM type 0 submap configuration sequence]

- Test Purpose

  Verify that the IUT does not accept a SAM type 0 submap configuration sequence from the Lower Tester when the Update_Mode is set to 0 and the SAM slot map in use contains the type 0 submap.

- Reference

  [10] 4.1.15.1

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM is disabled on the Lower Tester and the IUT.

- Test Procedure

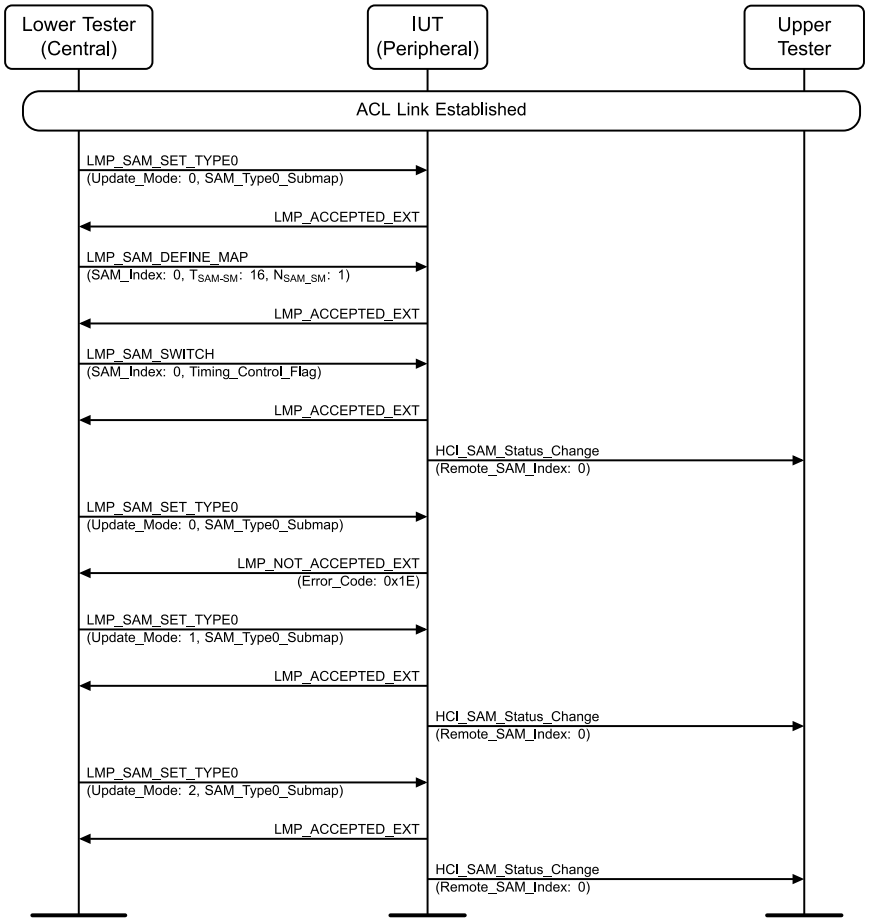


Figure 4.3: LMP/SAM/BI-03-C [Respond to invalid SAM type 0 submap configuration sequence] MSC

1. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU with the following parameters to the IUT to configure a type 0 submap:
   - Update_Mode = 0
   - SAM_Type0_Submap = 0x4A, 0x55, 0xBD, 0xAA, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
3. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters to define one SAM slot map containing the type 0 submap:
   - SAM_Index = 0
   - T_{SAM-SM} = 16
   - N_{SAM_SM} = 1
   - SAM_Submaps = 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
5. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT to enable the SAM slot map. Timing_Control_Flag is determined by CLK_{27} of the Central.
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.

7. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
8. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU to the IUT with the following parameters to reconfigure the type 0 submap:
    - Update_Mode = 0
    - SAM_Type0_Submap = 0x4A, 0x55, 0xBD, 0xAA, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
9. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode, Escape_Opcode, and Error_Code set to 0x1E (Invalid LMP Parameters).
10. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU to the IUT with the following parameters to reconfigure the type 0 submap:
    - Update_Mode = 1
    - SAM_Type0_Submap = 0xBD, 0xAA, 0x4A, 0x55, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
11. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
12. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
13. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU to the IUT with the following parameters to reconfigure the type 0 submap:
    - Update_Mode = 2
    - SAM_Type0_Submap = 0x4A, 0x55, 0xBD, 0xAA, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
15. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT accepts the SAM type 0 submap configuration and the SAM slot map define sequence and responds to the SAM switch sequence initiated by the Lower Tester correctly.

  The IUT sends the LMP_NOT_ACCEPTED PDU with Error_Code 0x1E (Invalid LMP Parameters) upon reception of the LMP_SAM_SET_TYPE0 PDU with the Update_Mode parameter set to 0.

  The IUT generates the HCI_SAM_Status_Change events correctly.

LMP/SAM/BI-04-C [Respond to invalid SAM type 0 submap]

- Test Purpose

  Verify that the IUT does not accept a SAM slot map define request from the Lower Tester when SAM type 0 submap is carried by the LMP_SAM_DEFINE_MAP PDU without prior configuration.

- Reference

  [10] 4.1.15.1

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM is disabled on the Lower Tester and the IUT.

- Test Procedure

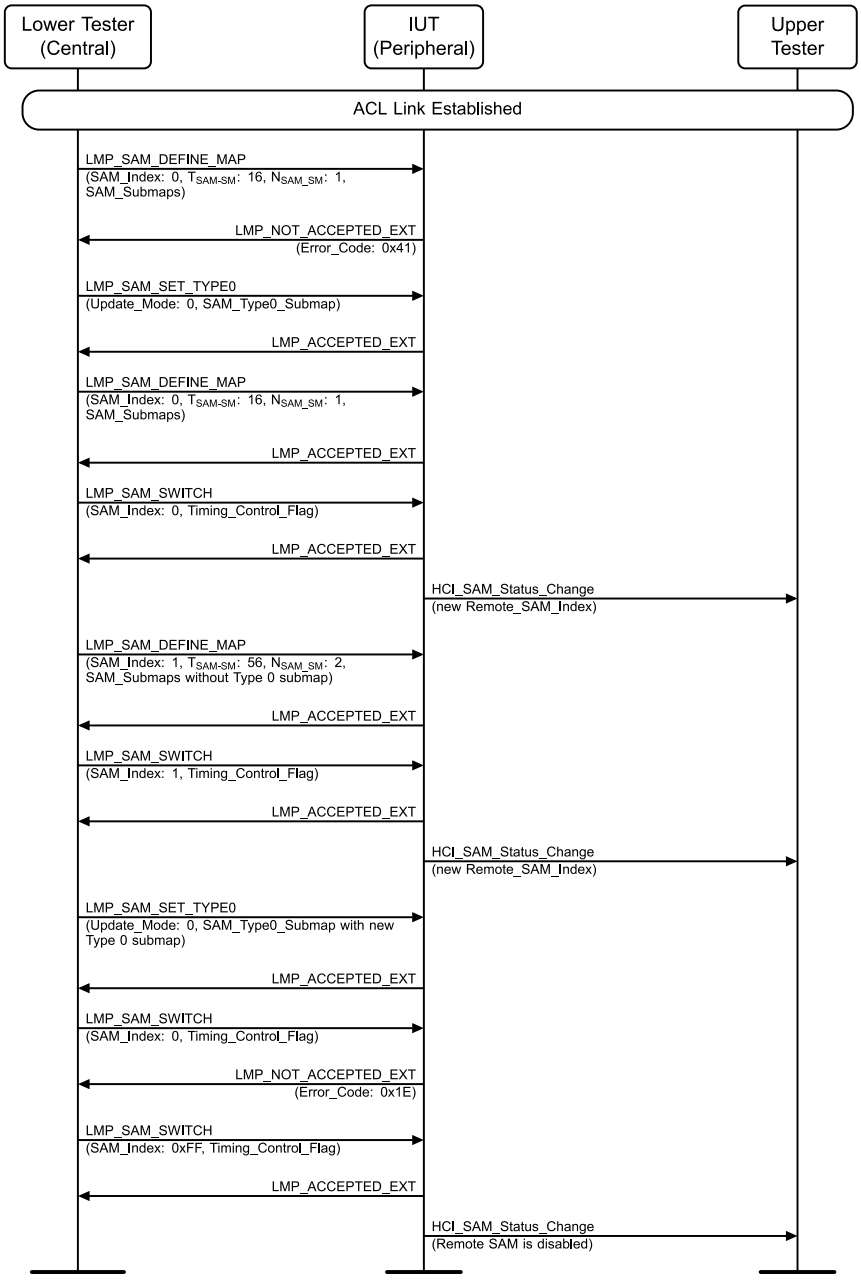


Figure 4.4: LMP/SAM/BI-04-C [Respond to invalid SAM type 0 submap] MSC

1. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters to define one SAM slot map containing a type 0 submap:
   - SAM_Index = 0
   - T_{SAM-SM} = 16
   - N_{SAM_SM} = 1
   - SAM_Submaps = 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode, Escape_Opcode, and Error_Code set to 0x41 (Type 0 Submap Not Defined).

3. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU with the following parameters to the IUT to configure a type 0 submap:
   - Update_Mode = 0
   - SAM_Type0_Submap = 0x4A, 0x55, 0xBD, 0xAA, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
5. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters to define one SAM slot map containing a type 0 submap:
   - SAM_Index = 0
   - $T_{SAM\text{-}SM}$ = 16
   - N_{SAM_SM} = 1
   - SAM_Submaps = 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
7. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT to enable the SAM slot map with SAM_Index = 0. Timing_Control_Flag is determined by CLK_{27} of the Central.
8. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
9. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
10. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters to define one SAM slot map without a type 0 submap:
    - SAM_Index = 1
    - $T_{SAM\text{-}SM}$ = 56
    - N_{SAM_SM} = 2
    - SAM_Submaps = 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
11. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
12. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT to enable the SAM slot map with SAM_Index = 1. Timing_Control_Flag is determined by CLK_{27} of the Central.
13. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
14. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
15. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU with the following parameters to the IUT to configure a new type 0 submap:
    - Update_Mode = 0
    - SAM_Type0_Submap = 0xBD, 0xAA, 0x4A, 0x55, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
16. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
17. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT to enable the SAM slot map with SAM_Index = 0. Timing_Control_Flag is determined by CLK_{27} of the Central.

18. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode, Escape_Opcode, and Error_Code set to 0x1E (Invalid LMP Parameters).
19. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with a valid SAM_Index = 0xFF to disable SAM.
20. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
21. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x41 (Type 0 Submap Not Defined*)* upon reception of the LMP_SAM_DEFINE_MAP PDU from the Lower Tester containing an undefined SAM type 0 submap.

  The IUT accepts slot map define sequence and responds to the SAM switch sequence initiated by the Lower Tester correctly.

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code 0x1E (Invalid LMP Parameters*)* upon reception of the LMP_SAM_SWITCH PDU from the Lower Tester with SAM_Index = 1 after the type 0 submap is reconfigured with Update_Mode = 0.

  The IUT continues on the current SAM slot map after a failed LMP_SAM_SWITCH, and no HCI_SAM_Status_Change event is generated.

  The IUT generates the HCI_SAM_Status_Change events correctly.

LMP/SAM/BI-05-C [Respond to invalid SAM index]

- Test Purpose

  Verify that the IUT does not accept a SAM switch request from the Lower Tester when the SAM index carried by the LMP_SAM_DEFINE_MAP PDU is not defined.

- Reference

  [10] 4.1.15

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM is disabled on the Lower Tester and the IUT.

- Test Procedure

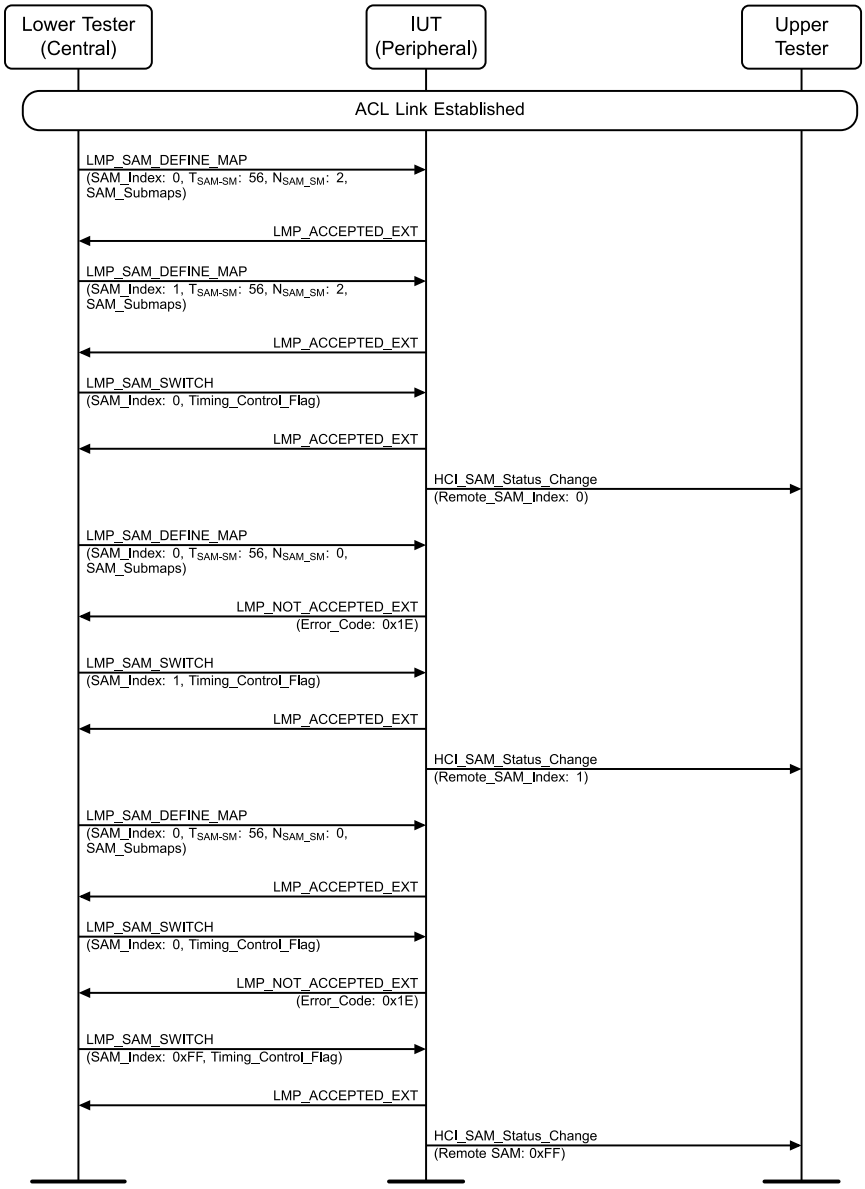


Figure 4.5: LMP/SAM/BI-05-C [Respond to invalid SAM index] MSC

1. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters:
   - SAM_Index = 0
   - T_{SAM-SM} = 56
   - N_{SAM_SM} = 2
   - SAM_Submaps = 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.

3. The Lower Tester sends the second LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters:
   - SAM_Index = 1
   - T_{SAM-SM} = 56
   - N_{SAM_SM} = 2
   - SAM_Submaps = 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
5. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with a SAM_Index = 0 to enable the SAM slot map. Timing_Control_Flag is determined by CLK_{27} of the Central.
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
7. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
8. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters to delete the SAM slot map with SAM_Index = 0:
   - SAM_Index = 0
   - T_{SAM-SM} = 56
   - N_{SAM_SM} = 0
   - SAM_Submaps = SAM_Submaps can have any value
9. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode, Escape_Opcode, and Error_Code set to 0x1E (Invalid LMP Parameters).
10. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with SAM_Index = 1 to enable the SAM slot map.
11. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
12. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 1, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
13. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters again to delete the SAM slot map with SAM_Index = 0:
   - SAM_Index = 0
   - T_{SAM-SM} = 56
   - N_{SAM_SM} = 0
   - SAM_Submaps = SAM_Submaps can have any value
14. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
15. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with an invalid SAM_Index = 0 to enable the SAM slot map.
16. The IUT responds to the Lower Tester with an LMP_NOT_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode, Escape_Opcode, and Error_Code set to 0x1E (Invalid LMP Parameters).
17. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with SAM_Index = 0xFF to disable SAM.
18. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.

19. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0xFF, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT accepts the slot map define sequence and responds to the SAM switch sequence initiated by the Lower Tester correctly.

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1E (Invalid LMP Parameters) upon reception of the LMP_SAM_DEFINE_MAP PDU from the Lower Tester with N_{SAM_SM} = 0 containing a currently selected SAM_Index.

  The IUT sends the LMP_NOT_ACCEPTED PDU to the Lower Tester with Error_Code set to 0x1E (Invalid LMP Parameters) upon reception of the LMP_SAM_SWITCH PDU containing an invalid SAM_Index.

  The IUT continues on the current SAM slot map after a failed LMP_SAM_SWITCH, and no HCI_SAM_Status_Change event is generated.

  The IUT generates the HCI_SAM_Status_Change events correctly.

LMP/SAM/BV-06-C [SAM is disabled after a successful role switch]

- Test Purpose

  Verify that SAM is disabled by the IUT after a successful role switch.

- Reference

  [10] 4.1.15.5

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM is disabled on the Lower Tester and the IUT.

- Test Procedure

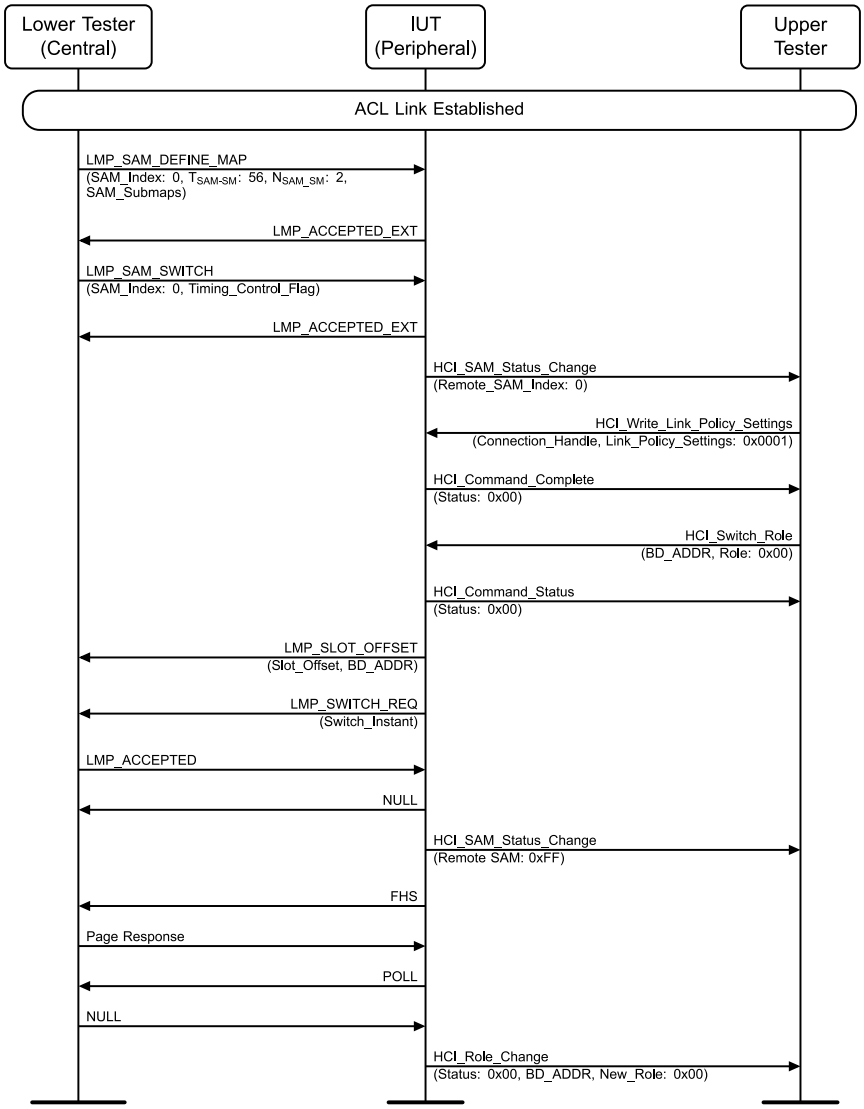


Figure 4.6: LMP/SAM/BV-06-C [SAM is disabled after a successful role switch] MSC

1. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters to define one SAM slot map:
   - SAM_Index = 0
   - T_{SAM-SM} = 56
   - N_{SAM_SM} = 2
   - SAM_Submaps = 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
3. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT to enable the SAM slot map. Timing_Control_Flag is determined by CLK_{27} of the Central.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.

5. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
6. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Hande and Link_Policy_Settings set to 0x0001 and receives a successful HCI_Command_Complete event in response.
7. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
8. The IUT sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant.
9. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
10. The IUT sends a NULL packet to the Lower Tester.
11. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0xFF, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
12. The IUT sends an FHS packet to the Lower Tester.
13. The Lower Tester responds to the IUT with a Page Response packet.
14. The IUT sends a POLL packet to the Lower Tester.
15. The Lower Tester responds to the IUT with a NULL packet.
16. The IUT sends a successful HCI_Role_Change event to the Upper Tester with the BD_ADDR and New_Role set to 0x00 (Central).

- Expected Outcome

  Pass verdict

  The IUT accepts the slot map define sequence and responds to the SAM switch sequence initiated by the Lower Tester correctly.

  The IUT generates the HCI_SAM_Status_Change events correctly, and SAM is disabled by the IUT when the role switch succeeds.

LMP/SAM/BV-07-C [SAM is resumed after a failed role switch]

- Test Purpose

  Verify that SAM is resumed by the IUT after a failed role switch.

- Reference

  [10] 4.1.15.5

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM disabled on the Lower Tester and the IUT.

- Test Procedure

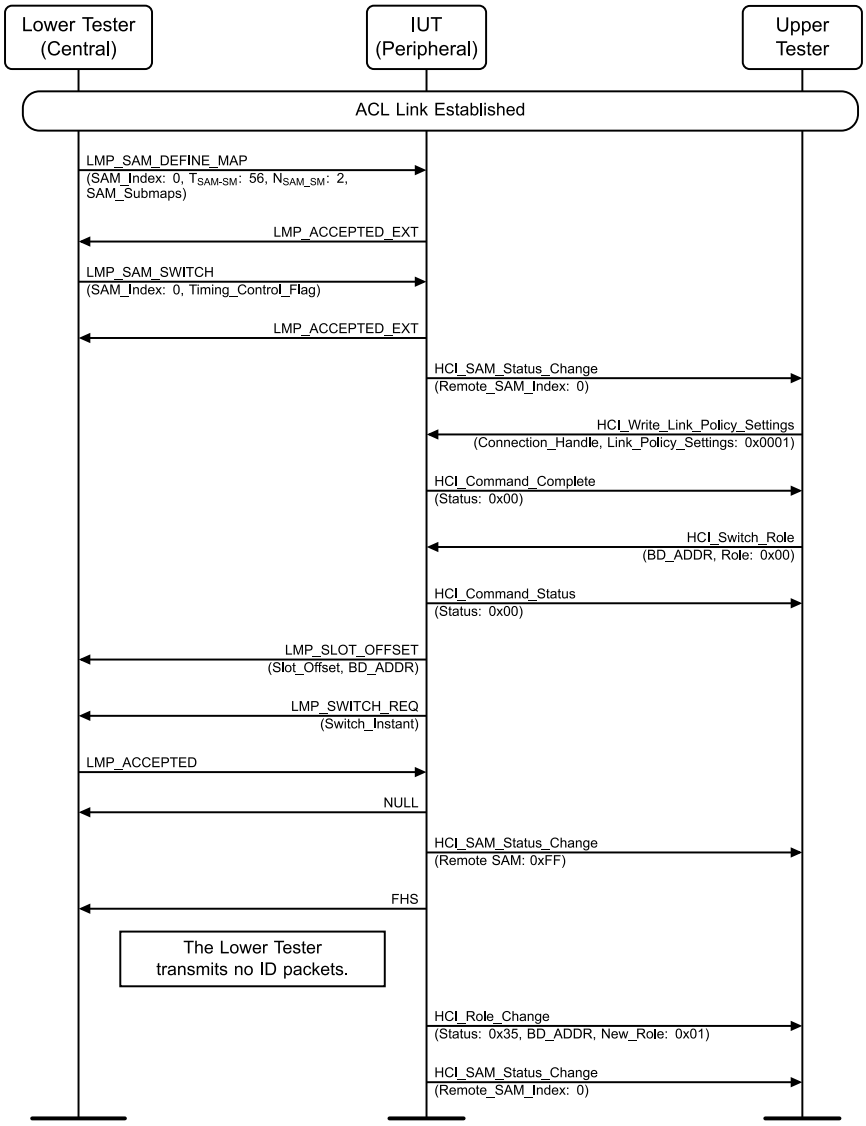


Figure 4.7: LMP/SAM/BV-07-C [SAM is resumed after a failed role switch] MSC

1. The Lower Tester sends an LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters to define one SAM slot map:
   - SAM_Index = 0
   - T_{SAM-SM} = 56
   - N_{SAM_SM} = 2
   - SAM_Submaps = 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
3. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT to enable the SAM slot map. Timing_Control_Flag is determined by CLK_{27} of the Central.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode in response.

5. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
6. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Hande and Link_Policy_Settings set to 0x0001 and receives a successful HCI_Command_Complete event in response.
7. The Upper Tester sends an HCI_Switch_Role command to the IUT with the BD_ADDR and Role set to 0x00 (Central) and receives a successful HCI_Command_Status event in response.
8. The IUT sends an LMP_SLOT_OFFSET PDU to the Lower Tester with the Slot_Offset and BD_ADDR and an LMP_SWITCH_REQ PDU with the Switch_Instant.
9. The Lower Tester responds to the IUT with an LMP_ACCEPTED PDU with the LMP_SWITCH_REQ PDU Opcode.
10. The IUT sends a NULL packet to the Lower Tester.
11. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0xFF, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
12. The IUT sends an FHS packet to the Lower Tester.
13. The Lower Tester transmits no ID packets to the IUT.
14. The IUT sends an HCI_Role_Change event to the Upper Tester with the Status set to 0x35 (Role Switch Failed), BD_ADDR, and New_Role set to 0x01 (Peripheral).
15. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT accepts the slot map define sequence and responds to the SAM switch sequence initiated by the Lower Tester correctly.

  SAM is resumed when the role switch fails.

LMP/SAM/BV-08-C [SAM and sniff mode]

- Test Purpose

  Verify that the Sniff mode takes precedence over SAM and that SAM is reinstated on exit from the Sniff mode.

- Reference

  [10] 4.1.15.6

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM disabled on the Lower Tester and the IUT.

- Test Procedure

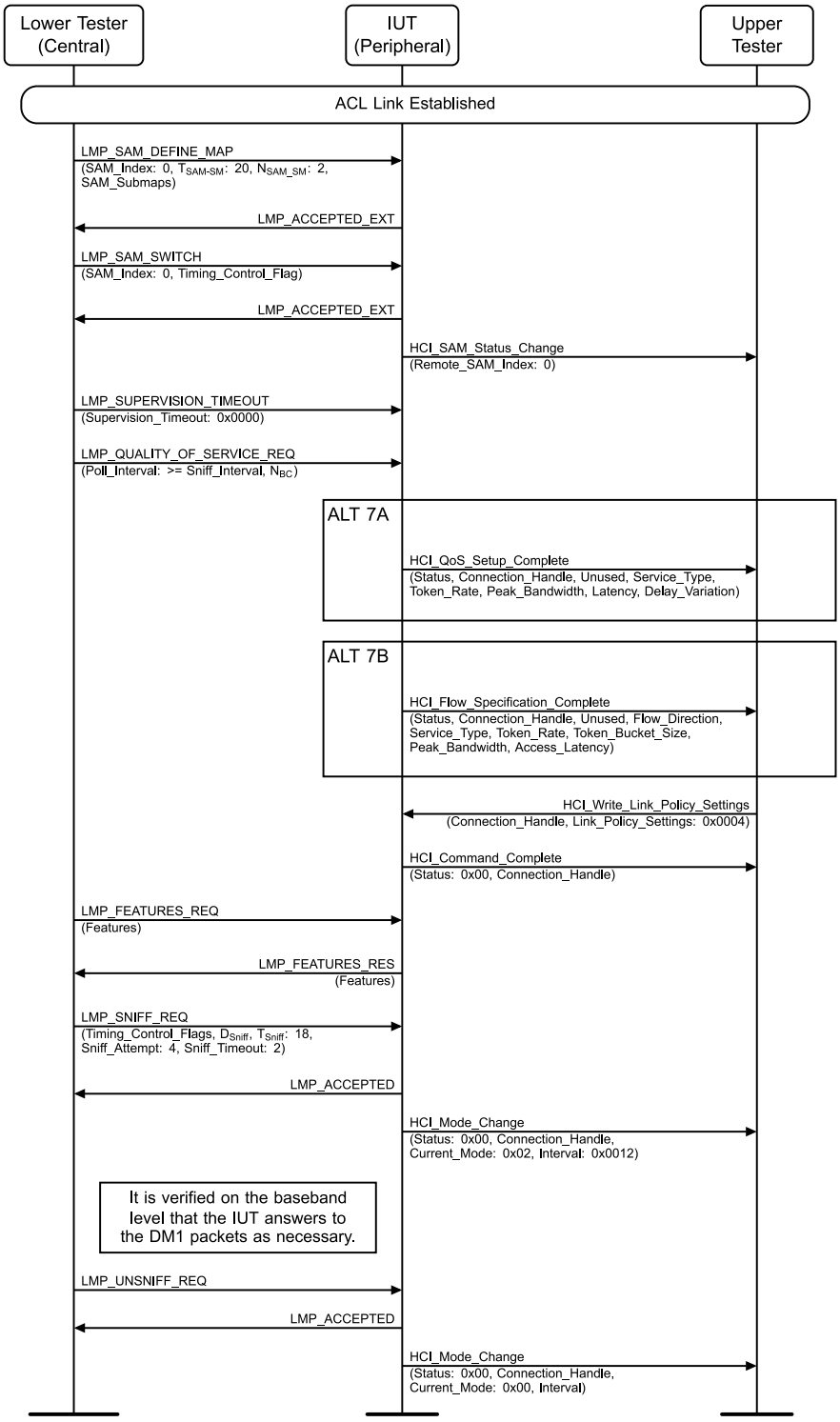


Figure 4.8: LMP/SAM/BV-08-C [SAM and sniff mode] MSC

1. The Lower Tester sends the first LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters:
   - SAM_Index = 0
   - T_{SAM-SM} = 20
   - N_{SAM_SM} = 2
   - SAM_Submaps = 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
3. The Lower Tester sends the first LMP_SAM_SWITCH PDU to the IUT with SAM_Index = 0. Timing_Control_Flag is determined by CLK_{27} of the Central.
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode in response.
5. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index set to 0, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.
6. The Lower Tester sends an LMP_SUPERVISION_TIMEOUT PDU with the Supervision_Timeout set to 0x0000 and an LMP_QUALITY_OF_SERVICE_REQ PDU with the Poll_Interval greater than or equal to the Sniff_Interval and N_{BC}.
7. Perform either alternative 7A or 7B depending on the IUT's response.

   Alternative 7A (The IUT sends the HCI_QoS_Setup_Complete event to the Upper Tester):

   7A.1. The IUT sends an HCI_QoS_Setup_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Service_Type, Token_Rate, Peak_Bandwidth, Latency, and Delay_Variation.

   Alternative 7B (The IUT sends the HCI_Flow_Specification_Complete event to the Upper Tester):

   7B.1. The IUT sends an HCI_Flow_Specification_Complete event to the Upper Tester with the Status, Connection_Handle, Unused, Flow_Direction, Service_Type, Token_Rate, Token_Bucket_Size, Peak_Bandwidth, and Access_Latency.
8. The Upper Tester sends an HCI_Write_Link_Policy_Settings command to the IUT with the Connection_Handle and Link_Policy_Settings set to 0x0004 (Sniff Mode) and receives a successful HCI_Command_Complete event in response.
9. The Lower Tester sends an LMP_FEATURES_REQ PDU to the IUT with the Features parameter.
10. The IUT responds to the Lower Tester with an LMP_FEATURES_RES PDU with the Features parameter.
11. The Lower Tester sends an LMP_SNIFF_REQ PDU to the IUT with Timing_Control_Flags, D_{Sniff}, T_{Sniff} set to 18, Sniff_Attempt set to 4, and Sniff_Timeout set to 2.
12. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_SNIFF_REQ PDU Opcode.
13. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x02 (Sniff Mode), and Interval set to 0x0012.
14. The Lower Tester sends POLL packets to the IUT according to Figure 4.9. DM1 packets transmitted by the Lower Tester contain data.
15. The IUT acknowledges each DM1 packet for at least 20*T_{Sniff} slots.
16. The Lower Tester sends an LMP_UNSNIFF_REQ PDU to the IUT.
17. The IUT responds to the Lower Tester with an LMP_ACCEPTED PDU with the LMP_UNSNIFF_REQ PDU Opcode.
18. The IUT sends a successful HCI_Mode_Change event to the Upper Tester with the Connection_Handle, Current_Mode set to 0x00 (Active Mode), and Interval.

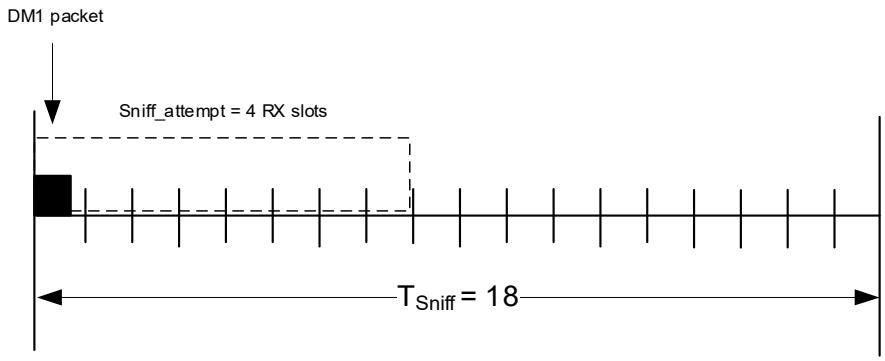


Figure 4.9: LMP/SAM/BV-08-C, Polling

- Expected Outcome

  Pass verdict

  The IUT responds to the SAM switch sequences initiated by the Lower Tester and generates the HCI_SAM_Status_Change events correctly.

  The IUT enters Sniff mode and acknowledges every DM1 packet for at least 20*T_{Sniff} slots, which is verified on the baseband level.

  No HCI_SAM_Status_Change event is generated after the IUT enters or exits Sniff mode.

LMP/SAM/BV-09-C [Respond to request for SAM Anchor Point using Initialization Procedure 2]

- Test Purpose

  Verify that the IUT correctly responds to a SAM anchor point with a SAM switch sequence to D_{SAM} initialization procedure 2.

- Reference

  [10] 4.1.15

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM disabled on the Lower Tester and the IUT.

- Test Procedure

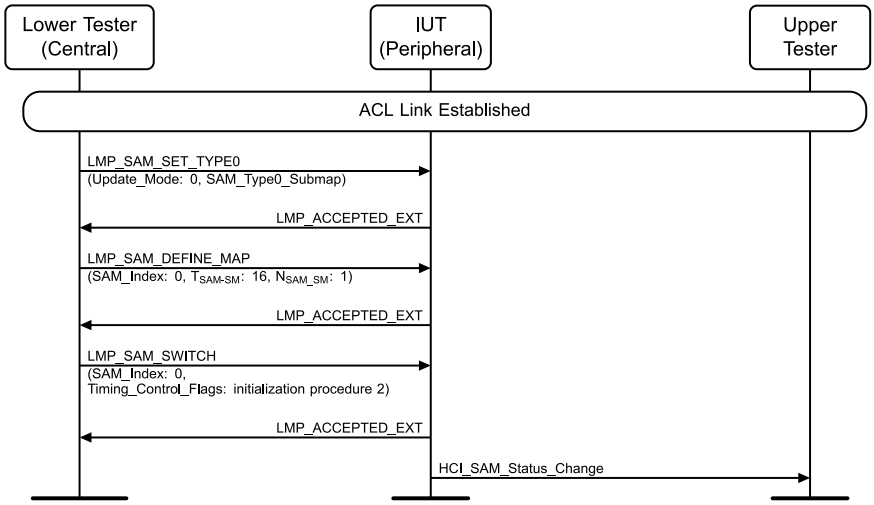


Figure 4.10: LMP/SAM/BV-09-C [Respond to request for SAM Anchor Point using Initialization Procedure 2] MSC

1. The Lower Tester sends an LMP_SAM_SET_TYPE0 PDU to the IUT with the following parameters:
   - Update_Mode = 0
   - SAM_Type0_Submap = 0x4A, 0x55, 0xBD, 0xAA, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
2. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
3. The Lower Tester sends the first LMP_SAM_DEFINE_MAP PDU to the IUT with the following parameters:
   - SAM_Index = 0
   - T_{SAM-SM} = 16
   - N_{SAM_SM} = 1
   - SAM_Submaps = 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
4. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
5. The Lower Tester sends an LMP_SAM_SWITCH PDU to the IUT with SAM_Index = 0 at the precise time such that the MLB of the Central's current clock (CLK_{27}) is 1; Timing_Control_Flags = initialization procedure 2.
6. The IUT responds to the Lower Tester with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
7. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT accepts the SAM switch to initialization procedure 2.

  The IUT generates an HCI_SAM_Status_Change event correctly.

LMP/SAM/BV-10-C [Initiate SAM Anchor Point using Initialization Procedure 2]

- Test Purpose

  Verify that the IUT will correctly initiate a SAM anchor point with a SAM switch sequence to D_{SAM} using initialization procedure 2.

- Reference

  [10] 4.1.15

- Initial Condition

  - The Lower Tester is configured as the Central in the CONNECTION state (active mode, ACL link).
  - The IUT is configured as the Peripheral in the CONNECTION state (active mode, ACL link).
  - SAM disabled on the Lower Tester and the IUT. SAM negotiations are triggered by the IUT with predefined parameters.

- Test Procedure

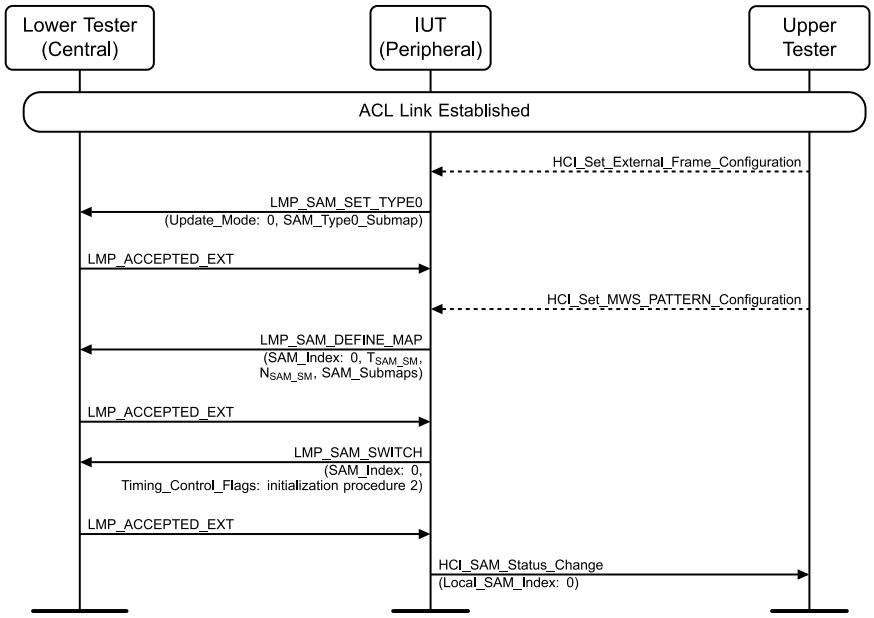


Figure 4.11: LMP/SAM/BV-10-C [Initiate SAM Anchor Point using Initialization Procedure 2] MSC

1. If necessary, the Upper Tester sends an HCI_Set_External_Frame_Configuration command to the IUT with MWS_Frame_Duration, MWS_Frame_Sync_Assert_Offset, MWS_Frame_Sync_Assert_Jitter, MWS_Num_Periods, Period_Duration, and Period_Type.
2. The IUT sends an LMP_SAM_SET_TYPE0 PDU to the Lower Tester with Update_Mode and SAM_Type0_Submap to configure a type 0 submap.
3. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SET_TYPE0 PDU Opcode and Escape_Opcode.
4. If necessary, the Upper Tester sends an HCI_Set_MWS_PATTERN_Configuration command to the IUT with MWS_Pattern_Index, MWS_Pattern_Num_Intervals, MWS_Pattern_Interval_Duration, and MWS_Pattern_Interval_Type.
5. The IUT sends an LMP_SAM_DEFINE_MAP PDU to the Lower Tester with SAM_Index set to 0, T_{SAM_SM}, N_{SAM_SM}, and SAM_Submaps to define three SAM slot maps.

6. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_DEFINE_MAP PDU Opcode and Escape_Opcode.
7. The IUT sends an LMP_SAM_SWITCH PDU to the Lower Tester with SAM_Index = 0 at the precise time such that the MLB of the Central's current clock (CLK_{27}) is 1; Timing_Control_Flags = initialization procedure 2.
8. The Lower Tester responds to the IUT with an LMP_ACCEPTED_EXT PDU with the LMP_SAM_SWITCH PDU Opcode and Escape_Opcode.
9. The IUT sends an HCI_SAM_Status_Change event to the Upper Tester with the Connection_Handle, Local_SAM_Index set to 0, Local_SAM_TX_Availability, Local_SAM_RX_Availability, Remote_SAM_Index, Remote_SAM_TX_Availability, and Remote_SAM_RX_Availability.

- Expected Outcome

  Pass verdict

  The IUT initiates the SAM switch with initialization procedure 2.

  The IUT generates an HCI_SAM_Status_Change event correctly.

5 Test case mapping

5.1 References

This document incorporates provisions from other publications by dated or undated reference. These references are cited at the appropriate places in the text, and the publications are listed hereinafter. Additional definitions and abbreviations can be found in [1] and [5].

[1] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification

[2] ISO/IEC 9646-1 and ISO/IEC 9646-2 OSI Conformance Testing Methodology and Framework

[3] ICS Proforma for Link Manager Protocol (LMP)

[4] Bluetooth Core Specification, Volume 3, Part D, Test Support

[5] Test Strategy and Terminology Overview

[6] Bluetooth Core Specification, Volume 2, Part A, Radio Specification

[7] Bluetooth Core Specification, Volume 2, Part E (Versions 1.2 to 5.1) or Volume 4, Part E (Version 5.2 and higher), Host Controller Interface Functional Specification

[8] Bluetooth Core Specification, Volume 2, Part F, Message Sequence Charts

[9] Bluetooth Core Specification, Volume 2, Part B, Baseband Specification

[10] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 4.2 or later

[11] Baseband (BB) Test Suite, BB.TS

[12] Bluetooth Core Specification, Volume 3, Part C, Generic Access Profile, Version 4.2 or later

[13] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 5.3 or later

[14] Bluetooth Core Specification, Volume 2, Part C, Link Manager Protocol Specification, Version 5.3 or later

[15] Appropriate Language Mapping Tables document

[16] Bluetooth Core Specification, Volume 2, Part H, Security Specification, Version 6.2 or later

5.2 Test Case Mapping Table

The Test Case Mapping Table (TCMT) maps test cases to specific requirements in the ICS. The IUT is tested in all roles for which support is declared in the ICS document.

The columns for the TCMT are defined as follows:

Item: Contains a logical expression based on specific entries from the associated ICS document. Contains a logical expression (using the operators AND, OR, NOT as needed) based on specific entries from the applicable ICS document(s). The entries are in the form of y/x references, where y corresponds to the table number and x corresponds to the feature number as defined in the ICS document for Link Manager Protocol [3].

If a test case is mandatory within the respective layer, then the y/x reference is omitted.

Feature: A brief, informal description of the feature being tested.

Test Case(s): The applicable test case identifiers are required for Bluetooth Qualification if the corresponding y/x references defined in the Item column are supported. Further details about the function of the TCMT are elaborated in [5].

For the purpose and structure of the ICS/IXIT, refer to [5].

Item	Feature	Test Case(s)
Deadlock Avoidance		
LMP 1/1	Accept Message	LMP/LIH/BV-80-C LMP/LIH/BV-81-C
Authentication		
LMP 3/3	Respond to authentication request	LMP/AUT/BV-01-C LMP/AUT/BV-42-C
LMP 4/3 AND CORE 1a/51	Error Return When a Unit Key is Requested, v5.1 or higher	LMP/AUT/BI-01-C
LMP 3/2 AND CORE 1b/54	Initiate Authentication Request, v5.4 or lower	LMP/AUT/BV-36-C
LMP 3/2 AND CORE 1a/60	Initiate Authentication Request, v6.0 or higher	LMP/AUT/BV-40-C LMP/SP/BI-14-C
LMP 3/3 AND CORE 1a/60	Respond to Authentication Request, v6.0 or higher	LMP/SP/BI-13-C
Pairing		
NOT LMP 3/1 AND (LMP 4/1 OR LMP 4/2)	Initiate Pairing	LMP/AUT/BV-04-C
LMP 3/1 AND (LMP 4/1 OR LMP 4/2)	Initiate Pairing – HCI Command Required to Pair	LMP/AUT/BV-52-C
NOT LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND CORE 1b/60	Initiate Pairing, v6.0 or earlier	LMP/AUT/BV-25-C

Item	Feature	Test Case(s)
LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND CORE 1b/60	Initiate Pairing, v6.0 or earlier – HCI Command Required to Pair	LMP/AUT/BV-54-C
NOT LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND CORE 1a/61	Initiate Pairing, v6.1 or later	LMP/AUT/BV-45-C
LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND CORE 1a/61	Initiate Pairing, v6.1 or later	LMP/AUT/BV-55-C
(LMP 4/1 OR LMP 4/2) AND HCI 16/16	Delete Stored Link Key	LMP/AUT/BV-41-C
(LMP 4/1 OR LMP 4/2) AND CORE 1a/60	Initiate Pairing, v6.0 or higher	LMP/AUT/BI-08-C
LMP 4/3 AND LMP 4/5 AND LMP 5/2	Respond to Pairing Request (variable PIN code)	LMP/AUT/BV-03-C LMP/AUT/BV-24-C
LMP 4/3 AND LMP 4/4 AND LMP 5/2	Respond to Pairing Request (Fixed PIN code)	LMP/AUT/BV-05-C LMP/AUT/BI-04-C
LMP 4/3 AND LMP 4/4 AND LMP 5/2 AND CORE 1b/60	Respond to Pairing Request (Fixed PIN code), v6.0 or earlier	LMP/AUT/BV-26-C
NOT LMP 3/1 AND LMP 4/3 AND LMP 4/4 AND LMP 5/2 AND CORE 1a/61	Respond to Pairing Request (Fixed PIN code), v6.1 or later	LMP/AUT/BV-46-C
LMP 3/1 AND LMP 4/3 AND LMP 4/4 AND LMP 5/2 AND CORE 1a/61	Respond to Pairing Request (Fixed PIN code), v6.1 or later – HCI Command Required to Pair	LMP/AUT/BV-57-C
NOT LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND LMP 4/5 AND LMP 4/6 AND LMP 5/2	Accept Switch (initiator becomes responder)	LMP/AUT/BV-06-C
LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND LMP 4/5 AND LMP 4/6 AND LMP 5/2	Accept Switch (initiator becomes responder) – HCI Command Required to Pair	LMP/AUT/BV-53-C

Item	Feature	Test Case(s)
NOT LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND LMP 4/5 AND LMP 4/6 AND LMP 5/2 AND CORE 1b/60	Accept Switch (initiator becomes responder), v6.0 or earlier	LMP/AUT/BV-27-C
LMP 3/1 AND (LMP 4/1 OR LMP 4/2) AND LMP 4/5 AND LMP 4/6 AND LMP 5/2 AND CORE 1b/60	Accept Switch (initiator becomes responder), v6.0 or earlier – HCI Command Required to Pair	LMP/AUT/BV-56-C
(LMP 4/1 OR LMP 4/2) AND LMP 4/5 AND LMP 4/6 AND LMP 5/2 AND CORE 1a/61	Accept Switch (initiator becomes responder), v6.1 or later	LMP/AUT/BV-47-C
LMP 4/3 AND LMP 2/19b AND LMP 24/2	Reject a pairing procedure (Host controlled)	LMP/AUT/BV-34-C
Link Keys		
LMP 5/3 AND LMP 5/2	Initiate Change of Link Key	LMP/AUT/BV-13-C LMP/AUT/BV-29-C
LMP 5/4 AND LMP 5/2	Accept Change of Link Key	LMP/AUT/BV-12-C LMP/AUT/BV-28-C
Encryption		
LMP 6/1 AND LMP 6/6	Initiate Encryption32	LMP/ENC/BV-06-C LMP/ENC/BV-09-C LMP/ENC/BI-02-C
LMP 6/1 AND LMP 6/6 AND CORE 1b/61	Initiate Encryption, Core v6.1 or earlier	LMP/ENC/BV-05-C
LMP 6/1 AND LMP 6/6 AND CORE 1a/62	Initiate Encryption, Core v6.2 or later	LMP/ENC/BV-82-C
NOT HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 5/8 AND CORE 1b/61	Initiate Encryption, Core v6.1 or earlier	LMP/ENC/BV-61-C LMP/ENC/BV-63-C
NOT HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 5/8 AND CORE 1a/62	Initiate Encryption, Core v6.2 or later	LMP/ENC/BV-79-C LMP/ENC/BV-89-C

Item	Feature	Test Case(s)
HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 5/8 AND CORE 1b/61	Initiate Encryption, Set Min Encryption Key_Size, Core v6.1 or earlier	LMP/ENC/BV-53-C LMP/ENC/BV-55-C
HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 5/8 AND CORE 1a/62	Initiate Encryption, Set Min Encryption Key_Size, Core v6.2 or later	LMP/ENC/BV-75-C LMP/ENC/BV-87-C
LMP 6/2	Accept Encryption Requests	LMP/ENC/BI-01-C LMP/ENC/BI-03-C LMP/ENC/BI-04-C LMP/ENC/BI-07-C
LMP 6/2 AND CORE 1b/61	Accept Encryption Requests, Core v6.1 or earlier	LMP/ENC/BV-01-C
LMP 6/2 AND CORE 1a/62	Accept Encryption Requests, Core v6.2 or later	LMP/ENC/BV-66-C LMP/ENC/BV-81-C
NOT HCI 16/68 AND LMP 6/2 AND CORE 1b/61	Accept Encryption Requests, Core v6.1 or earlier	LMP/ENC/BV-59-C
NOT HCI 16/68 AND LMP 6/2 AND CORE 1a/62	Accept Encryption Requests, Core v6.2 or later	LMP/ENC/BV-77-C
HCI 16/68 AND LMP 6/2 AND CORE 1b/61	Accept Encryption Requests, Set Min Encryption Key_Size, Core v6.1 or earlier	LMP/ENC/BV-51-C
HCI 16/68 AND LMP 6/2 AND CORE 1a/62	Accept Encryption Requests, Set Min Encryption Key_Size, Core v6.2 or later	LMP/ENC/BV-73-C
NOT HCI 16/68 AND LMP 6/2 AND LMP 6/11 AND LMP 5/8 AND CORE 1b/61	Accept AES-CCM Encryption Request, Core v6.1 or earlier	LMP/ENC/BV-60-C
NOT HCI 16/68 AND LMP 6/2 AND LMP 6/11 AND LMP 5/8 AND CORE 1a/62	Accept AES-CCM Encryption Request, Core v6.2 or later	LMP/ENC/BV-78-C
HCI 16/68 AND LMP 6/2 AND LMP 6/11 AND LMP 5/8 AND CORE 1b/61	Accept AES-CCM Encryption Request, Set Min Encryption Key_Size, Core v6.1 or earlier	LMP/ENC/BV-52-C

Item	Feature	Test Case(s)
HCI 16/68 AND LMP 6/2 AND LMP 6/11 AND LMP 5/8 AND CORE 1a/62	Accept AES-CCM Encryption Request, Set Min Encryption Key_Size, Core v6.2 or later	LMP/ENC/BV-74-C
NOT HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 6/11 AND LMP 5/8 AND CORE 1b/61	Initiate AES-CCM Encryption as Central, Core v6.1 or earlier	LMP/ENC/BV-64-C
NOT HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 6/11 AND LMP 5/8 AND CORE 1a/62	Initiate AES-CCM Encryption as Central, Core v6.2 or later	LMP/ENC/BV-90-C
HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 6/11 AND LMP 5/8 AND CORE 1b/61	Initiate AES-CCM Encryption as Central, Set Min Encryption Key_Size, Core v6.1 or earlier	LMP/ENC/BV-54-C
HCI 16/68 AND LMP 6/1 AND LMP 6/6 AND LMP 6/11 AND LMP 5/8 AND CORE 1a/62	Initiate AES-CCM Encryption as Central, Set Min Encryption Key_Size, Core v6.2 or later	LMP/ENC/BV-88-C
NOT HCI 16/68 AND LMP 6/1 AND LMP 6/11 AND LMP 5/8 AND CORE 1b/61	Initiate AES-CCM Encryption as Peripheral, Core v6.1 or earlier	LMP/ENC/BV-62-C
NOT HCI 16/68 AND LMP 6/1 AND LMP 6/11 AND LMP 5/8 AND CORE 1a/62	Initiate AES-CCM Encryption as Peripheral, Core v6.2 or later	LMP/ENC/BV-80-C
HCI 16/68 AND LMP 6/1 AND LMP 6/11 AND LMP 5/8 AND CORE 1b/61	Initiate AES-CCM Encryption as Peripheral, Set Min Encryption Key_Size, Core v6.1 or earlier	LMP/ENC/BV-56-C
HCI 16/68 AND LMP 6/1 AND LMP 6/11 AND LMP 5/8 AND CORE 1a/62	Initiate AES-CCM Encryption as Peripheral, Set Min Encryption Key_Size, Core v6.2 or later	LMP/ENC/BV-76-C

Item	Feature	Test Case(s)
LMP 6/8	Stop Encryption	LMP/ENC/BV-07-C
LMP 6/9	Accept Stop of Encryption	LMP/ENC/BV-04-C LMP/ENC/BV-08-C
LMP 2/14 AND LMP 6/6 AND CORE 1b/61	Initiate Broadcast Encryption, Core v6.1 or earlier	LMP/ENC/BV-10-C LMP/ENC/BV-13-C
LMP 2/14 AND LMP 6/6 AND CORE 1a/62	Initiate Broadcast Encryption, Core v6.2 or later	LMP/ENC/BV-83-C LMP/ENC/BV-84-C
LMP 2/14	Accept Broadcast Encryption	LMP/ENC/BV-57-C LMP/ENC/BV-58-C
LMP 2/14 AND CORE 1b/61	Accept Broadcast Encryption, Core v6.1 or earlier	LMP/ENC/BV-02-C LMP/ENC/BV-11-C
LMP 2/14 AND CORE 1a/62	Accept Broadcast Encryption, Core v6.2 or later	LMP/ENC/BV-67-C LMP/ENC/BV-68-C
LMP 1/2 AND NOT LMP 2/14	Accept Broadcast Encryption	LMP/ENC/BV-12-C
LMP 6/10	Encryption Pause/Resume	LMP/ENC/BV-14-C LMP/ENC/BV-15-C LMP/ENC/BV-16-C LMP/ENC/BV-17-C LMP/ENC/BV-18-C LMP/ENC/BV-19-C LMP/ENC/BV-20-C LMP/ENC/BV-21-C LMP/ENC/BV-23-C LMP/ENC/BV-24-C LMP/ENC/BI-05-C LMP/ENC/BI-08-C
LMP 6/1 AND CORE 1b/61	Initiate Encryption as Peripheral, Core v6.1 or earlier	LMP/ENC/BV-22-C
LMP 6/1 AND CORE 1a/62	Initiate Encryption as Peripheral, Core v6.2 or later	LMP/ENC/BV-69-C
Clock_Offset Information		
LMP 7/1	Request Clock_Offset Information	LMP/INF/BV-02-C
LMP 7/2	Respond to Clock_Offset Requests	LMP/INF/BV-01-C
Timing Accuracy Information		
LMP 9/2	Respond to Timing Accuracy Information Requests	LMP/INF/BV-05-C
LM Version Information		
LMP 10/1	Request LM version information	LMP/INF/BV-09-C
LMP 10/2	Version Information Requests	LMP/INF/BV-08-C LMP/LIH/BI-07-C LMP/LIH/BI-08-C
Feature Support		
LMP 11/1	Request Supported Features	LMP/INF/BV-11-C

Item	Feature	Test Case(s)
LMP 11/2	Respond to Supported Features Requests	LMP/INF/BV-10-C LMP/LIH/BV-150-C
LMP 11/3	Request Extended_Features	LMP/INF/BV-16-C
LMP 11/4	Respond to Extended_Features Requests	LMP/INF/BV-17-C
LMP 11/4 AND NOT (LMP 2/22 OR LMP 2/23)	Request Extended_Features – LE Features not Supported	LMP/INF/BV-22-C
LMP 2/22 AND NOT LMP 2/23	Request Extended_Features – LE Supported (Controller) only	LMP/INF/BV-23-C
LMP 2/22 AND LMP 2/23	Request Extended_Features – All LE Features Supported	LMP/INF/BV-24-C
Name Information		
LMP 12/1	Request Name Information	LMP/INF/BV-13-C LMP/INF/BV-18-C LMP/INF/BV-19-C
LMP 12/2	Respond to Name Requests	LMP/INF/BV-12-C
Role Switch		
LMP 8/1 AND LMP 13/1	Request Role Switch	LMP/LIH/BV-01-C LMP/LIH/BV-79-C LMP/LIH/BV-142-C LMP/LIH/BV-143-C LMP/LIH/BV-146-C LMP/LIH/BV-149-C
LMP 13/2	Accept Role Switch Requests	LMP/LIH/BV-02-C LMP/LIH/BV-78-C LMP/LIH/BV-144-C LMP/LIH/BV-148-C LMP/LIH/BV-151-C
LMP 1/2 AND NOT LMP 13/2	Unsupported Role Switch Requests	LMP/LIH/BV-03-C
LMP 13/2 AND LMP 2/4 AND LMP 2/6 AND LMP 11/4	Accept Role Switch Requests, Slot Offset and Extended Features supported	LMP/LIH/BI-04-C
Detach		
LMP 14/1	Detach Connection	LMP/LIH/BV-04-C LMP/LIH/BV-05-C LMP/LIH/BV-82-C
Invalid Packet Handling		
LMP 1/2	Incorrect Packets	LMP/LIH/BI-06-C LMP/LIH/BI-10-C
LMP 28/1	Invalid LMP packet type, APB	LMP/LIH/BI-09-C

Item	Feature	Test Case(s)
Hold Mode		
LMP 15/1 OR LMP 15/2 OR LMP 15/3	Force or Request Hold Mode, Peripheral	LMP/LIH/BV-09-C
LMP 15/1 OR LMP 15/2	Force or Request Hold Mode, Central	LMP/LIH/BV-10-C
LMP 15/1 OR LMP 15/3 OR LMP 15/4	Accept Hold Mode Request, Central	LMP/LIH/BV-11-C
LMP 15/3 OR LMP 15/4	Verify Hold Mode	LMP/LIH/BV-06-C
LMP 1/2 AND NOT LMP 2/7	Respond to Unsupported Hold Mode Requests	LMP/LIH/BV-12-C
Sniff Mode		
LMP 16/2	Request Sniff Mode	LMP/LIH/BV-15-C LMP/LIH/BV-17-C
LMP 16/2	Request Sniff Mode and Name	LMP/LIH/BV-18-C
LMP 16/3	Respond to Sniff Mode Requests	LMP/LIH/BV-14-C
LMP 1/2 AND NOT LMP 16/3	Respond to Unsupported Sniff Mode Requests	LMP/LIH/BV-20-C
LMP 16/5	Request Un-sniff	LMP/LIH/BV-19-C
LMP 16/6	Accept Un-sniff Requests	LMP/LIH/BV-16-C
Power Control		
LMP 18/1	Request to Increase Power	LMP/LIH/BV-77-C
LMP 18/2	Request to Decrease Power	LMP/LIH/BV-76-C
LMP 18/3	Respond when Max. Power Reached	LMP/LIH/BV-36-C
LMP 18/4	Respond when Min. Power Reached	LMP/LIH/BV-35-C
LMP 1/1 AND NOT LMP 2/13a	Power Control Request not supported	LMP/LIH/BV-152-C
Link Supervision Timeout		
LMP 19/1	Set Link Supervision_Timeout Value	LMP/LIH/BV-74-C
LMP 19/2	Accept Link Supervision_Timeout Setting with Event Reporting	LMP/LIH/BV-126-C

Item	Feature	Test Case(s)
Secure Simple Pairing		
LMP 2/19b	Secure Simple Pairing	LMP/SP/BV-01-C LMP/SP/BV-02-C LMP/SP/BV-03-C LMP/SP/BV-04-C LMP/SP/BV-05-C LMP/SP/BV-06-C LMP/SP/BV-07-C LMP/SP/BV-08-C LMP/SP/BV-09-C LMP/SP/BV-10-C LMP/SP/BV-11-C LMP/SP/BV-12-C LMP/SP/BV-13-C LMP/SP/BV-16-C LMP/SP/BV-17-C LMP/SP/BV-14-C LMP/SP/BV-15-C LMP/SP/BV-24-C LMP/SP/BV-25-C LMP/SP/BV-26-C LMP/SP/BV-27-C LMP/SP/BV-28-C LMP/SP/BV-29-C LMP/SP/BV-30-C LMP/SP/BV-31-C LMP/SP/BV-32-C LMP/SP/BV-66-C LMP/SP/BV-69-C
LMP 2/19b AND CORE 1b/62	Secure Simple Pairing, v6.2 and earlier	LMP/SP/BV-18-C LMP/SP/BV-19-C LMP/SP/BV-20-C LMP/SP/BV-21-C LMP/SP/BV-22-C LMP/SP/BV-23-C
LMP 2/19b AND CORE 1a/63	Secure Simple Pairing, v6.3 and later	LMP/SP/BV-70-C LMP/SP/BV-71-C LMP/SP/BV-72-C LMP/SP/BV-73-C LMP/SP/BV-74-C LMP/SP/BV-75-C
LMP 2/19b AND CORE 1a/62	Secure Simple Pairing, v6.2 or later	LMP/SP/BV-67-C LMP/SP/BV-68-C
LMP 2/19b AND LMP 4/7	Secure Simple Pairing with keyboard I/O capabilities	LMP/SP/BV-33-C LMP/SP/BV-34-C LMP/SP/BV-35-C LMP/SP/BV-36-C

Item	Feature	Test Case(s)
LMP 2/19a AND LMP 2/19b	Pairing Key Validation	LMP/SP/BI-01-C LMP/SP/BI-02-C LMP/SP/BI-03-C LMP/SP/BI-04-C LMP/SP/BI-05-C LMP/SP/BI-06-C
LMP 2/19a AND LMP 2/19b AND LMP 2/26	Pairing Key Validation – P256	LMP/SP/BI-07-C LMP/SP/BI-08-C LMP/SP/BI-09-C LMP/SP/BI-10-C LMP/SP/BI-11-C LMP/SP/BI-12-C
Sniff Subrating		
LMP 16/7	Sniff Subrating	LMP/LIH/BV-117-C LMP/LIH/BV-118-C LMP/LIH/BV-119-C LMP/LIH/BV-120-C LMP/LIH/BV-121-C LMP/LIH/BV-122-C LMP/LIH/BV-123-C LMP/LIH/BV-124-C LMP/LIH/BV-125-C
Quality of Service		
LMP 20/2	Force Change of Quality of Service	LMP/LIH/BV-39-C LMP/LIH/BV-42-C
LMP 20/3	Request Change of Quality of Service	LMP/LIH/BV-40-C LMP/LIH/BV-41-C
SCO Links		
LMP 2/12	SCO Links	LMP/LIH/BV-43-C LMP/LIH/BV-46-C LMP/LIH/BV-51-C LMP/LIH/BV-52-C LMP/LIH/BV-53-C LMP/LIH/BV-54-C LMP/LIH/BV-58-C LMP/LIH/BV-59-C
LMP 2/12 AND LMP 2a/12	SCO Link, HV2 packets	LMP/LIH/BV-44-C LMP/LIH/BV-47-C LMP/LIH/BV-49-C LMP/LIH/BV-50-C LMP/LIH/BV-57-C
LMP 2/12 AND LMP 2a/13	SCO Link, HV3 packets	LMP/LIH/BV-45-C LMP/LIH/BV-48-C LMP/LIH/BV-55-C LMP/LIH/BV-56-C

Item	Feature	Test Case(s)
LMP 1/2 AND NOT LMP 2/12	Reject SCO Links	LMP/LIH/BV-60-C
eSCO Links		
LMP 2/15	ESCO Support	LMP/LIH/BV-100-C LMP/LIH/BV-108-C LMP/LIH/BV-111-C LMP/LIH/BV-115-C LMP/LIH/BV-103-C LMP/LIH/BV-109-C LMP/LIH/BV-110-C LMP/LIH/BV-114-C
LMP 1/2 AND NOT LMP 2/15	No ESCO Support	LMP/LIH/BV-116-C
LMP 2/15 AND LMP 2a/32	eSCO Link, EV4 packets	LMP/LIH/BV-101-C LMP/LIH/BV-104-C LMP/LIH/BV-106-C LMP/LIH/BV-107-C
LMP 2/15 AND LMP 2a/33	eSCO Link, EV5 packets	LMP/LIH/BV-102-C LMP/LIH/BV-105-C
LMP 2/15 AND (LMP 2a/32 OR LMP 2a/33)	Modify eSCO	LMP/LIH/BV-112-C LMP/LIH/BV-113-C
Multi-Slot Packets		
LMP 22/1 OR LMP 22/2	Allow/Request Maximum Number of Slots to be used	LMP/LIH/BV-61-C LMP/LIH/BV-64-C
LMP 22/3	Accept Request of Maximum Number of slots to be used	LMP/LIH/BV-63-C
Paging_Scheme		
LMP 1/2 AND NOT LMP 23/2	Reject Suggested Page mode	LMP/LIH/BV-71-C
LMP 1/2 AND NOT LMP 23/4	Reject Suggested Page Scan mode	LMP/LIH/BV-72-C
Test Mode		
LMP 25/2 AND LMP 25/4	Ability to reject Test Mode when Test mode is disabled	LMP/TEM/BV-01-C
AFH		
LMP 26/2	Adaptive Frequency Hopping	LMP/AFH/BV-01-C LMP/AFH/BV-02-C LMP/AFH/BV-03-C
LMP 2/6 AND LMP 26/1 AND LMP 26/4	AFH and Role Switch, Peripheral	LMP/AFH/BV-06-C
LMP 2/6 AND LMP 26/1 AND LMP 26/4a	AFH and Role Switch, Central	LMP/AFH/BV-05-C
LMP 2/6 AND LMP 26/1	AFH and Role Switch	LMP/AFH/BV-09-C

Item	Feature	Test Case(s)
LMP 2/7 AND LMP 26/4	AFH and Hold Mode	LMP/AFH/BV-08-C
LMP 26/4	Channel Classification	LMP/AFH/BV-04-C
EDR		
LMP 1/2 AND NOT LMP 2/17	Device does not support Enhanced Data_Rate	LMP/LIH/BV-83-C
LMP 14a/1	Enter Enhanced Data_Rate	LMP/LIH/BV-84-C
LMP 14a/2	Exit Enhanced Data_Rate	LMP/LIH/BV-85-C
LMP 14b/1	Enter and Exit eSCO Enhanced Data_Rate Connection	LMP/LIH/BV-86-C LMP/LIH/BV-87-C
EPC		
LMP 18/8	Respond to EPC Increase Requests	LMP/LIH/BV-127-C
LMP 18/9	Respond to EPC Decrease Requests	LMP/LIH/BV-128-C
LMP 18/10	Respond to EPC go to Max Requests	LMP/LIH/BV-129-C
LMP 18/5	Request EPC Increase	LMP/LIH/BV-130-C
LMP 18/6	Request EPC Decrease	LMP/LIH/BV-131-C
LMP 2/20	Report Unsupported Modulations Correctly	LMP/LIH/BV-133-C
BR/EDR Secure Connections		
LMP 2/26	Secure Authentication	LMP/AUT/BV-14-C LMP/AUT/BV-15-C LMP/AUT/BV-16-C LMP/AUT/BV-17-C LMP/AUT/BV-18-C LMP/AUT/BV-19-C LMP/AUT/BV-20-C LMP/AUT/BV-21-C LMP/AUT/BV-22-C LMP/AUT/BV-23-C LMP/AUT/BI-02-C LMP/AUT/BI-03-C LMP/AUT/BV-35-C LMP/AUT/BV-43-C LMP/AUT/BV-44-C
LMP 2/26 AND CORE 1b/60	Secure Authentication, v6.0 or earlier	LMP/AUT/BV-30-C LMP/AUT/BV-31-C LMP/AUT/BV-32-C LMP/AUT/BV-33-C
LMP 2/26 AND CORE 1a/61	Secure Authentication, v6.1 or later	LMP/AUT/BV-48-C LMP/AUT/BV-49-C LMP/AUT/BV-50-C LMP/AUT/BV-51-C
LMP 6/2 AND LMP 6/11 AND CORE 1b/61	Accept AES-CCM Encryption Request, Core v6.1 or earlier	LMP/ENC/BV-26-C LMP/ENC/BV-33-C

Item	Feature	Test Case(s)
LMP 6/2 AND LMP 6/11 AND CORE 1a/62	Accept AES-CCM Encryption Request, Core v6.2 or later	LMP/ENC/BV-71-C LMP/ENC/BV-72-C
LMP 6/9 AND LMP 6/11	Stop AES-CCM Encryption from Central	LMP/ENC/BV-27-C
LMP 6/1 AND LMP 6/11	Stop AES-CCM Encryption from Host	LMP/ENC/BV-28-C
LMP 6/11 AND LMP 27/1	Initiate LMP Ping	LMP/ENC/BV-29-C LMP/ENC/BV-31-C LMP/ENC/BV-32-C LMP/ENC/BV-46-C LMP/ENC/BV-48-C LMP/ENC/BV-49-C
LMP 6/11 AND LMP 2/12	SCO Connection creation fails when AES-CCM encryption is enabled	LMP/LIH/BI-01-C LMP/LIH/BI-02-C
LMP 6/1 AND LMP 6/6 AND LMP 6/11 AND CORE 1b/61	Initiate AES-CCM Encryption as Central, Core v6.1 or earlier	LMP/ENC/BV-34-C LMP/ENC/BV-50-C
LMP 6/1 AND LMP 6/6 AND LMP 6/11 AND CORE 1a/62	Initiate AES-CCM Encryption as Central, Core v6.2 or later	LMP/ENC/BV-85-C LMP/ENC/BV-86-C
LMP 6/1 AND LMP 6/11 AND CORE 1b/61	Initiate AES-CCM Encryption as Peripheral, Core v6.1 or earlier	LMP/ENC/BV-25-C
LMP 6/1 AND LMP 6/11 AND CORE 1a/62	Initiate AES-CCM Encryption as Peripheral, Core v6.2 or later	LMP/ENC/BV-70-C
LMP 6/8 AND LMP 6/11	Initiate AES-CCM Encryption Stop	LMP/ENC/BV-35-C
LMP 6/9 AND LMP 6/11	Stop AES-CCM Encryption, Peripheral request	LMP/ENC/BV-36-C
LMP 6/10 AND LMP 6/11	Encryption Pause/Resume	LMP/ENC/BV-37-C LMP/ENC/BV-38-C LMP/ENC/BV-39-C LMP/ENC/BV-40-C LMP/ENC/BI-06-C LMP/ENC/BI-09-C
LMP 2/14 AND LMP 6/6 AND LMP 6/11	Broadcast Encryption	LMP/ENC/BV-45-C
LMP 6/11 AND LMP 27/2	Respond to LMP Ping	LMP/ENC/BV-30-C LMP/ENC/BV-47-C

Item	Feature	Test Case(s)
LMP 2/19b AND LMP 2/26	Secure Simple Pairing – P256	LMP/SP/BV-41-C LMP/SP/BV-42-C LMP/SP/BV-43-C LMP/SP/BV-44-C LMP/SP/BV-45-C LMP/SP/BV-46-C LMP/SP/BV-48-C LMP/SP/BV-49-C LMP/SP/BV-50-C LMP/SP/BV-51-C LMP/SP/BV-52-C LMP/SP/BV-53-C LMP/SP/BV-60-C LMP/SP/BV-61-C LMP/SP/BV-62-C LMP/SP/BV-63-C LMP/SP/BV-64-C LMP/SP/BV-65-C LMP/SP/BV-37-C LMP/SP/BV-38-C LMP/SP/BV-39-C LMP/SP/BV-40-C LMP/SP/BV-47-C
LMP 2/19b AND LMP 2/26 AND CORE 1b/62	Secure Simple Pairing – P256, v6.2 and earlier	LMP/SP/BV-54-C LMP/SP/BV-55-C LMP/SP/BV-56-C LMP/SP/BV-57-C LMP/SP/BV-58-C LMP/SP/BV-59-C
LMP 2/19b AND LMP 2/26 AND CORE 1a/63	Secure Simple Pairing – P256, v6.3 and later	LMP/SP/BV-76-C LMP/SP/BV-77-C LMP/SP/BV-78-C LMP/SP/BV-79-C LMP/SP/BV-80-C LMP/SP/BV-81-C
LMP 2/6 AND LMP 2/26 AND LMP 6/10 AND LMP 6/11	Secure Connections and role switch	LMP/ENC/BV-41-C LMP/ENC/BV-42-C LMP/ENC/BV-43-C LMP/ENC/BV-44-C
LMP 2/19b AND LMP 2/26 AND LMP 3/2	Secure Authentication	LMP/AUT/BV-37-C
LMP 3/2	Initiate authentication after connection completed	LMP/AUT/BV-38-C LMP/AUT/BV-39-C
LMP 2/26 AND LMP 3/3	Secure Authentication and role switch	LMP/AUT/BI-06-C LMP/AUT/BI-07-C

Item	Feature	Test Case(s)
Piconet Clock Adjust		
LMP 2/28		LMP/XCL/BV-01-C LMP/XCL/BV-02-C LMP/XCL/BV-03-C LMP/XCL/BV-04-C
Slot Availability Mask		
LMP 29/1	Initiate SAM negotiations	LMP/SAM/BV-02-C LMP/SAM/BV-10-C
LMP 29/2	Respond to SAM negotiations	LMP/SAM/BV-01-C LMP/SAM/BI-03-C LMP/SAM/BI-04-C LMP/SAM/BI-05-C LMP/SAM/BV-09-C
LMP 2/6 AND LMP 2/29	SAM and role switch	LMP/SAM/BV-06-C LMP/SAM/BV-07-C
LMP 2/8 AND LMP 2/29	SAM and sniff mode	LMP/SAM/BV-08-C
Connection Establishment		
LMP 24/3	Connection Establishment	LMP/LIH/BV-145-C LMP/LIH/BV-147-C

Table 5.1: Test case mapping

6 Revision history and acknowledgments

Revision History

Publication Number	Revision Number	Date	Comments
10	2.0.E.0	2004-10-20	Initial document from version 1.2 Changes for V2.0 + EDR TSE 641 for TP/ENC/BV-03-CTP/ENC/BV-12-C, TP/LIH/BV-03-C, TP/LIH/BV-12-C,TP/LIH/BV-20-C, TP/LIH/BV-34-C, TP/LIH/BV-60-C.TP/LIH/BV116-C, TP/LIH/BV-71-C, and TP/LIH/BV-72-C TSE 655 for TP/LIH/BV-111-C TSE 656 for TP/AUT/BV-01-C TSE 665 for TP/LIH/BV-101-C and TP/LIH/BV-102-C TSE 598 for TP/ENC/BV-10-C TSE 599 for TP/LIH/BV-104-C and TP/LIH/BV-104-C TSE 600 for TP/AFH/BV-05-C TSE 653 for TP/LIH/BV-103-C, TP/LIH/BV-106-CTP/LIH/BV-107-C, TP/LIH/BV-110-C, TP/LIH/BV-112-C, and TP/LIH/BV-113-C. TSE 679 for TP/LIH/BV-100-C, TP/LIH/BV-104-C, and TP/LIH/BV-105-C.
	2.0.E.1 Draft	2004-11-04	Editorial change to TCMT 14a to 15 and 14b to 16.
11	2.0.E.1	2004-11-04	First version for 1.2/2.0/2.0 + EDR available for qualification.
	2.0.E.2r1	2005-03-24	TSE 695 for TCMT
12	2.0.E.2	2005-03-24	Prepare for publication.
	2.0.E.3r1	2005-03-29	Remove TSE 653 for TP/LIH/BV-103-C, TP/LIH/BV-106-CTP/LIH/BV-107-C, TP/LIH/BV-110-C, TP/LIH/BV-112-C, and TP/LIH/BV-113-C.
13	2.0.E.3	2005-03-29	Prepare for publication.
	2.0.E.4r1	2005-03-31	TSE 757 for TP/LIH/BV-83-C.
14	2.0.E.4	2005-03-31	Prepare for publication.
	2.0.E.5r0	2005-08-08	TSE 733 for TP/LIH/BV-61-C and TP/LIH/BV-64-C750 for TP/AFH/BV-09-C TSE 753 for TP/AFH/BV-04-C801 for TOC814 for TP/ENC/BV-12-C
15	2.0.E.5r1	2005-10-18	Add TSE 699 for TP/LIH/BV-84-C and TP/LIH/BV-84-C. Fix header and front page for specification versions
	2.0.E.6r0	2006-05-10	TSE 844: TP/LIH/BV-42-C: Add comment to MSCTSE 918: TP/LIH/BV-42-C: Changed comment beneath MSC from “POLL” to “allowed” TSE 935: TP/LIH/BV-86: Fix 2-EV3 packet sentence in test procedure. TSE 943: change TP/LIH/BV-84-C and TP/LIH/BV-85-C: ‘Notes’ TSE 1819: correct references to TP/ENC/BV figures
16	2.0.E.6	2006-06-05	Prepare for publication.

Publication Number	Revision Number	Date	Comments
	2.1.E.1r0 to 2.1.E.1r10	2006-09-27	Add Section 4.1.x “Conformance” TSE 1741: TP/LIH/BV-83-C: Add 0x20 as an error code. TSE 1798: TP/LIH/BV-61 and TP/LIH/BV-64: Add DM1 as an allowed packet parameter MSCs. TSE 1785: See TSE 935; TSE 1800: TP/LIH/BV-47-C, TP/LIH/BV-48-C: use of HV packets; TSE 1801: TP/LIH/BV-106-C: use of HV packets; TSE 1808: TP/LIH/BV-85-C; change MSC, change Notes TSE 1888:Remove “Applicable if” clauses from TP/AUT/BV-05-C, TP/AUT/BV-12-C, TP/AUT/BV-0513-C,TP/ENC/BV-01-C,TP/ENC/BV-04-C, TP/ENC/BV-05-C, TP/ENC/BV-06-C, TP/ENC/BV-07-C, TP/ENC/BV-08-C, TP/ENC/BV-09-C, TP/ENC/BV-10-C, TP/ENC/BV-11-C, TP/ENC/BV-12-C, TP/ENC/BV-13-C, TP/INF/BV-05-C, TP/INF/BV-4-C, TP/LIH/BV-01|02|03|06|09|10|11|12|14|15|16|17|18|19|20|22|23|24|25|26|27|29|32|34|35|36|37|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|60|61|63|64|71|72|75|76|77|78|79|87110|101|102|103|104|105|106|107|108|109|110|111|112|113|1114|115|116|-C,TP/AFH/BV-04|05|06|07|08|09-C. Add TP/LIH/BV-117-C through TP/LIH/BV-125-C for Sniff Subrating. Add TP/LIH/BV-126-C for Link Supervision Timeout Changed Event. Add TP/ENC/BV-14-C through TP/ENC/BV-19-C for Encryption Pause Resume.
	2.1.E.1r0 to 2.1.E.1r10	2006-09-27	Make changes to match updated Sniff subrating test case spec Add Section 5.2 and subsections for Simple Pairing Test cases and related text, including TP/SP/BV-01-C through TP/SP/BV-29-C Change MSC for SP/BV-07 Spec erratum #1993 changes to Figs 5.9, 5.10, 5.11, 5.13, 5.14 Remove “Applicable for” statement in sections 5.x.x -Figure 5.139 and 5.140: Add boxes with ALT labels -Figures 5.1 – 5.4: Add Optional box to MSCs Spec erratum 1993 changes to Figs 5.9, 5.10, 5.11, 5.13, 5.14 Add test cases: TP/SP/BV-30-C, TP/SP/BV-31-C, TP/SP/BV-32-C, TP/INF/BV-18-C, TP/INF/BI-19-C, TPINF/BV-20-C, TP/INF/BV-21-C TCMT Changes: Add row for TP/INF/BV-18-C, TP/INF/BI-19-C, TPINF/BV-20-C, TP/INF/BV-21-C, Add new SP cases TSE 2038: Change ‘F’ to ‘D’ in Event_Mask parameter Figure 5.1 – 5.4

Publication Number	Revision Number	Date	Comments
			TSE 2045: TP/SP/BV-06, TP/SP/BV-07, TP/SP/BV-12, TP/SP/BV-13, TP/SP/BV-18, TP/SP/BV-19, TP/SP/BV-20, TP/SP/BV-21, TP/SP/BV-22, TP/SP/BV-23, TP/SP/BV-29 TSE 2047: SP/BV/20-C AND SP/BV/21-C: Remove HCI_Remote_OOB_Data_Request event and –Reply command TSE 2053: add to MSC TP/SP/BV-21 TP/SP/BV-23, TP/SP/BV-27, Remove from MSC TP/SP/BV-18-C AND TP/SP/BV-24-C TSE 2058: Fixed second MSC for SP/BV-29-C TSE 2066: TP/LIH/BV-118-C Pass verdict 2D = 640 slots. TSE 2074: TP/SP/BV-24, TP/SP/BV-25, TP/SP/BV-26, TP/SP/BV-27: MSCs and pass verdict for -24 TSE 2080: TP/SP/BV-28-C similar to TP/SP/BV-29
17	2.1.E.1	2007-06-07	Prepare for publication.
	2.1.E.2r0	2007-Nov	TSE 2075: TP/SP/BV-19, TPSP/BV-23, TP/SP/BV-25, TP/SP/BV-27 TSE 2060: Add notes to TP/ENC/BV-14, TP/ENC/BV-15, TP/ENC/BV-18 and TP/ENC/BV-19, TP/ENC/BV-16 and TP/ENC/BV-17 TSE 2214: Move 5.7.22.11 to 5.7.27; add 5.7.27.3TSE 2225: TP/ENC/BV-14-C, TP/ENC/BV-15-C, TP/ENC/BV-16-C, TP/ENC/BV-17-C: update MSCs TSE 2235: TP/ENC/BV-18-C TSE 2242: TP/SP/BV-14-C TSE 2243: TP/SP/BV-05-C, TP/SP/BV-30-C TP/SP/BV-31-C, TP/SP/BV-32-C, updated MSCs TSE 2247:TP/INF/BV-21: Change parameter in MSC TSE 2248: TP/INF/BV-18 TP/INF/BV-19 TP/INF/BV-20 MSCs. TSE 2250: TP/SP/BV-31-C MSC caption. TSE 2277: TP/INF/BV-18 TP/INF/BV-19 TP/INF/BV-20TSE 2306: TP/SP/BV-11-C: Change pass verdict TSE 2312: TP/SP/BV-30-C,TP/SP/BV-31-C and TP/SP/BV-32-C change MSC opcodes TSE 2313: TP/SP/BV-28-C: MSC update TSE 2322: TP/INF/BV-18-C, TP/INF/BV-19-C, TP/INF/BV-20-C: Add parameters (remove ‘?’)TSE 2394: TP/LIH/BV-84-C, TP/LIH/BV-85-C TSE 2398: TP/ENC/BV-17-C: update MSC TSE 2401; TP/SP/BV-31-C: update MSC TSE 2403: TP/SP/BV-04-C: update Initial condition
18	2.1.E.2	2008-04-01	Prepare for publication.
	2.1.E.3r0	2008-10-08	TSE 2419: TP/SP/BV-20-C: Correct MSC TSE 2617: TP/INF/BV-14-C TSE 2626: TP/ENC/BV-05-C, TP/ENC/BV-06-C, TP/ENC/BV-07-C, TP/ENC/BV-10-C, TP/ENC/BV-18-C, TP/ENC/BV-19-C: Add note in test procedure
19	2.1.E.3	2008-12-12	Prepare for publication.

Publication Number	Revision Number	Date	Comments
	2.1.E.4r0-1	2009-02-12	Add EPC test cases to section 5.7.14. Update TCMT with new EPC test case mappings. Updated TP/ENC/BV-01-C MSC and verdicts. Input reviewer comments for 3.0 +HS test cases.
20	3.0.H.0 2.1.E.4	2009-02-12	Prepare for publication.
	3.0.H.0a 2.1.E.4a	2009-05-11	Fix to TSE 2626. Test cases affected: TP/ENC/BV-05-C, TP/ENC/BV-06-C, TP/ENC/BV-07-C, TP/ENC/BV-10-C, TP/ENC/BV-18-C, TP/ENC/BV-19-C
21	3.0.H.1	2009-08-16	TSE 2981: Correction to TCMT for TP/LIH/BV-133-C TSE 2941: TP/LIH/BV-127-C, TP/LIH/BV-128-C, update opcode in MSCs
	3.0.H.2r0	2009-12-01	Added TP/INF/BV-22-C for LE features
	3.0.H.2r1	2009-12-02	Fixed TCMT for new test case
22	4.0.0	2009-12-17	Prepare for publication
	4.0.1r0	2010-11-24	TSE 2978:Remove last line from TCMT for TP/RD/BV-01-C
	4.0.1r1	2011-07-13	Address reviewer (MS) comment: Apply TSE 3047 to TP/LIH/BV-61-C per TSE Approval reason
23	4.0.1	2011-07-18	Prepare for publication
	4.0.2r0	2011-11-06	TSE 4228: Update TCMT, new test cases TP/SP/BV-33-C, TP/SP/BV-34-C, TP/SP/BV-35-C, TP/SP/BV-36-C
24	4.0.2	2012-03-30	Prepare for publication
	4.0.3r0	2012-05-16	TSE: 4613: TP/INF/BV-22-C; Update MSC
25	4.0.3	2012-07-24	Prepare for publication.
	4.0.4r0	2012-11-05	TSE 4920: Added new test TP/ENC/BV-22-C (Initiate Encryption) for slave initiating encryption in section 5.5.1, TCMT entry in Encryption section. TSE 4921: Added new tests TP/ENC/BV-20-C, TP/ENC/BV-21-C, TP/ENC/BV-22-C and TP/ENC/BV-23-C in section 5.5.2, TCMT entry in Encryption section.
26	4.0.4	2012-11-16	Prepare for Publication
	4.0.5r1	2013-05-29	TSE 5068: Updated MSC for TP/LIH/BV-02-C, slave and master labels were reversed. TSE 5069: Added TP/ENC/BV-24-C to TCMT for “Encryption Pause/Resume” mapped to “LMP 6/10”.
27	4.0.5	2013-07-02	Prepare for publication
	4.0.6r0 to 4.0.6r4	2013-07-03 – 2013-09-16	Template Conversion: - Update of language to match BTI approved wording (example, fail verdicts) - Removal of Test Subgroup Objectives - Removal of sections marked “N/A” - New Pass/Fail Verdict Criteria section added

Publication Number	Revision Number	Date	Comments
	4.1.0r01	2013-09-16	BR/EDR Secure Connections CR Integration
	4.1.0r02	2013-10-01	Updated graphics from WMF to VSD--dh.
	4.1.0r03	2013-10-07	TSE 5182: Editorial cleanup of the TCMT. TSE 5189: Removed “Note: If legacy host is supported, then only MSC 2 should be executed. Otherwise, both MSCs shall be executed.” in test procedure of TP/END/BV/20-C. TSE 5200: Update MSCs for TP/ENC/BV-20-C and TP/ENC/BV-23-C. TSE 5267: Update MSCs for TP/ENC/BV-14-C and TP/ENC/BV-15-C. TSE 5302: Updated master and slave labels for MSC in TP/LIH/BV-42-C. TSE 5346: Update in Initial Condition, Test Procedure, MSC and removal of the inconclusive verdict for TP/ENC/BV-07-C. TSE 5347: Updated MSC for TP/LIH/BV-123-C to match Test Procedure. TSE 5350: Updated TP/SP/BV-01-C and TP/SP/BV-02-C.
	4.1.0r04	2013-10-09	Piconet Clock Adjust CR
28	4.1.0	2013-12-03	Prepare for Publication
	4.1.1r00	2014-04-07	TSE 5471: Updated MSC for TP/INF/BV-22-C. TSE 5530: Updated MSC, step C of the test procedure and the pass verdict for TP/XCL/BV-01-C and TP/XCL/BV-02-C. TSE 5582: Updated MSC and Pass verdict for TP/LIH/BV-135-C, TP/LIH/BV-137-C, TP/LIH/BV-139-C and TP/LIH/BV-141-C
29	4.1.1	2014-07-07	TCRL 2014-1 Publication
	4.2.0r00	2014-11-24	Revved version to align with Core 4.2 release
30	4.2.0	2014-12-04	Prepare for TCRL 2014-2 publication
	4.2.1r00	2015-10-07	TSE 6565: Resolved TBD opcodes in MSCs in Section 4.2.6 and 4.2.7; removed TBD opcodes in MSCs for TP/SP/BV-06-Cb, TP/SP/BV-08-Cb, TP/SP/BV-10-Cb, TP/SP/BV-45-Cb, TP/SP/BV-28-C, and TP/SP/BV-29-C; resolved TBD opcodes in MSCs for TP/SP/BV-54-C, TP/SP/BV-55-C, TP/SP/BV-56-C, TP/SP/BV-57-C, TP/SP/BV-58-C, TP/SP/BV-59-C, TP/SP/BV-60-C, TP/SP/BV-61-C, TP/SP/BV-62-C, and TP/SP/BV-63-C; revised two references in MSC for TP/SP/BV-47-C from TP/SP/BV-TBD01 to TP/SP/BV-41-C.
31	4.2.1	2015-12-22	Prepared for TCRL 2015-2 publication.
	4.2.2r00	2016-02-15	TSE 6889: Deleted invalid references to DSCO and DESCO in test cases TP/LIH/BV-43-C and TP/LIH/BV-103-C.
32	4.2.2	2016-07-07	Prepared for TCRL 2016-1 publication

Publication Number	Revision Number	Date	Comments
	5.0.0r00	2016-07-08	Integrated Slot Availability Mask CRr06 for Core Specification v5.0 release
	5.0.0r01	2016-06-29	Issue 6888: “the MSB of CLK27” changed to “CLK_{27}.” Replaced “the first LMP_SAM_switch PDU” with “an LMP_SAM_switch PDU" in test case TP/SAM/BV-01-C. Removed all references to “D_{SAM}.” Updated SAM_Submaps values for PDUs "$N_{SAM\text{-}SM}$ = 0" in TP/SAM/BI-05-C.
33	5.0.0	2016-12-13	Approved by BTI. Prepared for TCRL 2016-2 publication.
	5.0.1r00	2017-05-18	Converted to new Test Case ID conventions as defined in TSTO v4.1.
34	5.0.1	2017-07-05	Approved by BTI. Prepared for TCRL 2017-1 publication.
	5.0.2r00	2017-10-12	TSE 9779: Revised LMP/SAM/BV-02-C test procedure.
	5.0.2r01	2017-10-18	TSE 9873: Revised LMP/AUT/BV-13-C and LMP/ENC/BV-37-C – 38-C MSCs. Missing Underscores in HCI_Change_Connection_Link_Key_Complete Event.
	5.0.2r02	2017-10-19	TSE 9928: ESR 11 - Change Flags field to Unused in QoS Setup Complete and Flow Specification Complete. Revised MSCs for LMP/LIH/BV-39-C, LMP/LIH/BV-40-C, LMP/LIH/BV-41-C, LMP/LIH/BV-42-C, LMP/LIH/BV-81-C by removing reference to Flags where QoS Setup Complete Event is mentioned.
35	5.0.2	2017-12-07	Approved by BTI. Prepared for TCRL 2017-2 publication.
	5.0.3r00	2018-01-05	Template update.
	5.0.3r02	2018-02-22	TSE 10263 (rating 1): Replaced MSC for LMP/INF/BV-21-C.
	5.0.3r02	2018-02-22	TSE 10238 (rating 1): Replaced Verifying MSC for LMP/LIH/BV-22-C.
	5.0.3r03	2018-03-02	TSE 10342 (rating 3): Changed initial condition for LMP/AUT/BV-14-C to 17-C to the exact wording used for the P-256 Simple Pairing tests. TSE 10341 (rating 3): For Section 4.2.7 (Secure Simple Pairing P256) MSC, moved the HCI_Write_Secure_Connections_Host_Support command from after to before HCI_Write_Scan_Enable. TSE 10257 (rating 3): For LMP/INF/BV-16-C MSC, added an optional if exchange to HCI_Read_Remote_Extended_Features. TSE 10256 (rating 3): Changed initial condition for LMP/INF/BV-05-C from “Pairing / Authentication and features request has to be carried out.” to “ACL connection has been established between the IUT and the Lower Tester.”

Publication Number	Revision Number	Date	Comments
	5.0.3r04	2018-03-13	TSE 9710 (rating 3): Clarified LMP QoS PDU generation: replaced MSCs in LMP/LIH/BV-14-C, 15-C, 17-C, 18-C, 39-C, 40-C, LMP/SAM/BV-08-C; revised test procedure in LMP/LIH/BV-39-C, 40-C; and replaced polling figure in LMP/SAM/BV-08-C.
	5.0.3r05	2018-03-20	TSE 10469 (rating 3): Fixed problematic exchanges: replaced LMP/SP/BV-26-C MSC b, 27-C and MSC b.
	5.0.3r06	2018-03-22	TSE 10468 (rating 1): Changed 28 to 31 in MSC caption for LMP/ENC/BV-31-C. TSE 10456 (rating 3): Added “with Keypress notification” to headings for LMP/SP/BV-33-C to 36-C. Added missing ends of test procedures to LMP/SP/BV-35-C and 36-C MSC b.
	5.0.3r07	2018-04-12	TSE 10471 (rating 3): Updated LMP/SP/BV-01-C Test Procedure to allow only MSC2.
	5.0.3r08	2018-04-13	TSE 10472 (rating 4): Removed MSC 1 from test case LMP/SP/BV-05-C and revised MSC 2 to take out the notion of it being alternative 2. Updated the Pass verdict, removed Note in the intro, and changed the cross reference in the Initial Condition. TSE 10537 (rating 3): Invert direction of first LMP_sniff_subrating_req and LMP_sniff_subrating_res arrows for the LMP/LIH/BV-119-C test procedure MSC.
	5.0.3r09	2018-05-09	TSE 10256 (rating 3): Changed initial condition to reference Default Settings for test case LMP/INF/BV-05-C.
	5.0.3r10	2018-05-10	TSE 10658 (rating 1): Replaced MSC for test case LMP/LIH/BV-16-C.
	5.0.3r11	2018-05-15	TSE 9782 (rating 2): Updated MSCs and figures for test cases LMP/LIH/BV-14-C, 15-C, 17-C, 18-C, 39-C, 40-C; and LMP/SAM/ BV-08-C.
	5.0.3r12	2018-06-05	TSE 10470 (rating 3): Replaced MSC 2 in test procedure for test case LMP/AUT/BV-06-C.
	5.0.3r13	2018-06-08	TSE 10257: fixed integration error. In the last HCI event in the MSC, changed the page number from 0x00 to 0x01.
	5.0.3r14	2018-06-12	TSE 6810 (rating 1): Revised pass verdict for test cases LMP/SP/BV-20-C, 26-C, and 27-C. TSE 10257: fixed integration and formatting error in MSC.
	5.0.3r15	2018-06-20	Core E10734 Pairing Updates TS CR: Added new test cases LMP/SP/BI-01-C to 12-C, and added them to the TCMT. Updated Section 4.3.5 Pass/Inconclusive/Fail Verdict Conventions with inconclusive verdict conventions.
36	5.0.3	2018-07-02	Approved by BTI. Prepared for 2018-1 publication.

Publication Number	Revision Number	Date	Comments
	5.0.4r01-r10	2018-07-26 - 2018-11-08	TSE 10810 (rating 1): Revised MSCs using the old HCI_IO_Capability_Response command instead of the correct HCI_IO_Capability_Request_Reply command. Affects these MSCs: LMP/SP/BV-07-C to 29-C, and 33-C to 36-C. Core Minor Enhancements Batch 1 Test CRr10-clean: Added “Flow Specification” section and new test case LMP/LIH/BV-88-C to spec text and TCMT. Issue 11122: Deleted v5.1 MEP19 test case LMP/LIH/BV-88-C and removed test case mapping. TSE 10539 (rating 3): Updated MSC and test procedure steps for test cases LMP/ENC/BV-30-C and 47-C. TSE 10553 (rating 3): Updated MSC, test procedure steps, and pass verdict for test cases LMP/ENC/BV-29-C, 32-C, 46-C, and 49-C. TSE 10677 (rating 4): Added new test cases LMP/SAM/BV-09-C and LMP/SAM/BV-10-C and updated TCMT with new test cases. TSE 10753 (rating 3): Updated MSCs for test cases LMP/LIH/BV-121-C and 122-C. TSE 10890 (rating 3): Updated MSCs for sections 4.2.2 Encryption, and 4.2.6 AES-CCM Encryption. TSE 10753 (rating 3): Updated MSCs for test cases LMP/LIH/BV-121-C and 122-C. TSE 10890 (rating 3): Updated MSCs for sections 4.2.2 Encryption, and 4.2.6 AES-CCM Encryption. TSE 10929 (rating 3): Updated MSCs and pass verdict for test cases LMP/AUT/BV-03-C, 04-C, and LMP/SP/BV-05-C. Updated MSCs for test cases LMP/AUT/BV-05-C and 06-C. Updated pass verdict for test cases LMP/SP/BV-01-C and 02-C. Updated TCMT for test cases LMP/AUT/BV-03-C, 05-C, and 06-C. TSE 10953 (rating 3): Updated initial condition for test cases LMP/AFH/BV-05-C to 08-C. TSE 10960 (rating 1): Updated initial condition for test cases LMP/SAM/BV-06-C and 07-C. TSE 10995 (rating 3): Updated MSC in sections 4.2.1 Authentication, 4.2.2 Encryption, and 4.2.6 AES-CCM Encryption. Updated initial condition for test cases LMP/AUT/BV-04-C and 06-C. Updated initial condition, test procedure, and MSCs for test cases LMP/ENC/BV-05-C, 06-C, and 22-C. TSE 11037 (rating 1): Updated pass verdict for test case LMP/ENC/BV-22-C. TSE 11047 (rating 1): Added new test case LMP/AUT/BI-01-C and updated the TCMT with the new test case. TSE 11125 (rating 1): Changed section title of test cases LMP/LIH/BV-51-C to "Accept SCO Closure as Slave" and LMP/LIH/BV-59-C to "Accept SCO Closure as Master".

Publication Number	Revision Number	Date	Comments
			TSE 11126 (rating 2): In TCMT, updated mapping for test case LMP/LIH/BV-34-C. TSE 10995 (rating 3): Fixed paragraph breaks between the initial condition and test procedure for test cases LMP/ENC/BV-05-C, 06-C, and 22-C. TSE 11208 (rating 1) Updated description of LMP/LIH/BV-46-C. Missing to update the Table of Contents. Updated version on document front page. TSE 10553 (rating 3): Added “messages” as postfix to LMP_Ping_Res to test procedure steps and pass verdict for test cases LMP/ENC/BV-29-C, 32-C, 46-C, and 49-C.
	5.1.0r00-r01	2018-11-13 - 2018-11-28	Updated revision number from 5.0.4 to 5.1.0 to align with the adoption of Core Specification version 5.1. TSE 11313 (rating 4): Updated MSC for AUT/BI-01-C with PIN Code Request Event.
37	5.1.0	2018-12-07	Approved by BTI. Prepared for TCRL 2018-2 publication.
	5.1.1r00–r10	2019-03-27 – 2019-07-18	TSE 11462 (rating 1): Updated initial condition for test cases LMP/SP/BI-01-C to 06-C to add the specific IXIT statement “TSPX_new_key_failed_count.” TSE 11139 (rating 2): Updated Item column in TCMT for test cases LMP/ENC/BV-03-C and -12-C and LMP/LIH/BV-03-C, -12-C, -20-C, -34-C, -60-C, -71-C, -72-C, -83-C, and -116-C. TSE 11566 (rating 3): Updated old MSCs with new MSCs for test cases LMP/ENC/BV-16-C, -17-C, -21-C, -24-C, and -41-C – -44-C. TSE 11478 (rating 3): Updated old MSC with new MSC for test case LMP/INF/BV-22-C. TSE 10882 (rating 3): Updated old MSCs with new MSCs for test cases LMP/ENC/BV-18-C and -19-C. TSE 11297 (rating 4): Deleted test cases LMP/LIH/BV-22-C – -27-C, -29-C, -32-C, -34-C, and -75-C and updated TCMT accordingly. TSE 11393 (rating 4): Deleted test case LMP/AFH/BV-07-C and updated TCMT accordingly. TSE 11946 (rating 1): Updated the MSC in the “Simple Pairing” section to address an issue with the Event_Mask. TSE 11920 (rating 1): Updated initial condition for test case LMP/ENC/BV-01-C. TSE 11821 (rating 3): Added an Inconclusive Verdict item for test cases LMP/SP-BI-05-C and -06-C. TSE 11479 (rating 3): Updated MSC for test case LMP/SP/BV-47-C.

Publication Number	Revision Number	Date	Comments
			Incorporated changes associated with Key Negotiation specification erratum 11838: Added “Key Size Negotiation as Slave” section with test cases for E0 and AES encryption types for IUT in Acceptor and Initiator roles. Added “Key Size Negotiation as Master” section with test cases for E0 and AES encryption types. (New TCID numbers LMP/ENC/BV-51-C – -56-C.) Incorporated changes associated with Key Negotiation specification erratum 11838: Updated to indicate if the IUT enforces a minimum encryption key size of 56 bits; that has a range of 7–16 octets (updated "Key Size Negotiation as Slave" section, containing test cases LMP/ENC/BV-51-C, -52-C, -55-C, and -56-C (swapped out MSC, updated test Step 5b); updated "Key Size Negotiation as Master" section, containing test cases LMP/ENC/BV-53-C and -54-C (swapped out MSC, updated test Step 5)).
38	5.1.1	2019-08-01	Approved by BTI. Prepared for TCRL 2019-1 publication.
	p39r00–r04	2019-09-06 – 2019-12-03	TSE 12260 (rating 1): Replaced MSC for test case LMP/ENC/BV-12-C. TSE 12155 (rating 1): Replaced MSC for test case LMP/AUT/BI-01-C. TSE 12293 (rating 4): Deleted test case LMP/LIH/BV-132-C and updated TCMT accordingly. TSE 12110 (rating 1): Fixed references to align with changes made in erratum 11876. TSE 12761 (rating 2): Updated the TCMT only for test cases LMP/SP/BI-01-C – -12-C to include LMP 2/19a and removed SUM ICS 21/17 from the mapping. Revised document numbering convention, setting last release publication of 5.1.1 as p38; added publication number column to Revision History. Updated Contributors list and alphabetized new entries.
39	p39	2020-01-07	Approved by BTI on 2019-12-22. Prepared for TCRL 2019-2 publication.
	p40r00–r19	2020-01-24 – 2021-06-21	TSE 11492 (rating 2): Added a test condition and an inconclusive verdict and updated the MSC for test case LMP/LIH/BV-61-C. TSE 12153 (rating 4): Deleted test cases LMP/LIH/BV-135-C, -137-C, -139-C, and 141-C. Added new section "Rejecting SCO Connection request when AES-CCM encryption is enabled" featuring new test cases LMP/LIH/BI-01-C and -02-C. Updated TCMT accordingly. TSE 12801 (rating 3): Updated test steps and pass verdict for test case LMP/LIH/BV-121-C to better align with spec.

Publication Number	Revision Number	Date	Comments
			TSE 12840 (rating 2): Updated test procedure for test cases LMP/LIH/BV-111-C and LMP/LIH/BV-116-C to align with spec after Erratum 12590. TSE 12857 (rating 1): Renamed instances of “simple pairing” to “secure simple pairing” as appropriate, including some instances in the TCMT. Updated TCRL naming with “Secure” when the test title changed. TSE 13088 (rating 4): To address an issue with needing to allow for poll/null packets as required in SCO and eSCO tests, updated the test purpose, MSC, test procedure, and/or Pass verdict for TCs LMP/LIH/BV-43-C, -46-C, -50-C, -53-C, -54-C, -56-C, -100-C – -107-C, and -110-C – -113-C. TSE 13199 (rating 2): Updated test steps for test case LMP/SAM/BV-02-C. TSE 13240 (rating 4): To address changes stemming from E12908, added a reference to LMP v5.3, added new TCs LMP/AUT/BI-04-C – -07-C, LMP/ENC/BI-01-C and -02-C, and LMP/LIH/BI-04-C, -05-C, and /BV-151-C. Updated TCMT accordingly. TSE 13266 (rating 4): Added new test cases LMP/AUT/BV-18-C – -23-C and LMP/AUT/BI-02-C and -03-C to address role switch during secure authentication. Updated TCMT accordingly. TSE 13267 (rating 4): Added new test cases LMP/AUT/BV-24-C – -33-C to address checking that authentication actually authenticates. Updated TCMT accordingly. TSE 13340 (rating 2): Replaced the MSCs for TCs LMP/LIH/BV-108-C and -109-C. TSE 13350 (rating 1): Replaced MSC for TC LMP/INF/BV-16-C to align with requirements of E13293. TSE 13354 (rating 2): Replaced the MSCs for TCs LMP/ENC/BV-11-C and -13-C. TSE 13380 (rating 4): Added new TCs LMP/LIH/BV-142-C – -144-C to address role switch rejection. Updated TCMT accordingly. TSE 13436 (rating 2): Updated MSC, test procedure, and Pass verdict for TC LMP/AFH/BV-04-C. TSE 13439 (rating 3): Updated test procedure for TC LMP/LIH/BV-19-C. TSE 14648 (rating 1): Updated parameter names, where found, to new format with underscores and capitalization fixed. Also added tables to Abbreviations section to inform regarding such definitions.

Publication Number	Revision Number	Date	Comments
			TSE 14809 (rating 4): Updated title of section from “Control of Multi-slot Packets – Slave” to “Multi-slot Packets – Slave”, made a minor edit to the test group subobjectives, and added new TCs LMP/LIH/BV-145-C and -146-C; updated title of section from “Control of Multi-slot Packets – Master” to “Multi-slot Packets – Master”, made a minor edit to the test group subobjectives, and added new TCs LMP/LIH/BV-147-C and -148-C. Updated TCMT accordingly. TSE 14898 (rating 1): Removed TC LMP/INF/BV-14-C and updated TCMT accordingly. TSE 14906 (rating 4): Added new reference to Core 4.2 or later GAP; added new TCs LMP/AUT/BV-34-C and LMP/SP/BV-66-C. Updated TCMT accordingly. TSE 14942 (rating 3): Updated several rows in the TCMT to call out the new LMP 2/19b item. TSE 14986 (rating 1): Updated MSCs of TCs LMP/INF/BV-08-C and -09-C to align with new parameter naming. TSE 14997 (rating 2): Updated test cases LMP/INF/BV-10-C, -11-C, -16-C, and -17-C to support full testing of LMP feature bits. TSE 15043 (rating 1): Fixed typo in test purpose of TCs LMP/LIH/BV-01-C and -02-C. TSE 15053 (rating 4): To address E15031 regarding an issue with forcing authentication even if a link has been authenticated or a link key is stored, added new TCs LMP/AUT/BV-36-C and -37-C. Updated TCMT accordingly. TSE 15089 (rating 3): Updated MSC and Pass Verdict of TC LMP/INF/BV-22-C to address changes required in E15039. TSE 15092 (rating 2): Updated several rows in the TCMT to remove items that are no longer used in the LMP ICS. TSE 15175 (rating 2): Updated TCs LMP/INF/BV-18-C and -19-C and deleted TCs LMP/INF/BV-20-C and -21-C to address mistaken SSP testing. Updated TCMT accordingly. TSE 15257 (rating 3): To address an issue involved with the random generation of the 128-bit nonce noted in E11424, added a reference for "Specification of the Bluetooth System, Volume 2, Part H (Security Specification), Version 5.3 or later" and updated the Test Purpose, Reference, Test Procedure, and Pass Verdict of TCs LMP/SP/BV-06-C and -07-C. TSE 15452 (rating 1): To address Erratum 15352, globally changed “Master” to “Central” and “Slave” to “Peripheral”. TSE 15488 (rating 1): To address Erratum 15334, globally changed “Master Slave” language as it relates to role switches.

Publication Number	Revision Number	Date	Comments
			TSE 15490 (rating 1): To address Erratum 15328, globally changed “HCI Master Link Key” to “HCI Link Key Selection” and “HCI Master Link Key Complete” to “HCI Link Key Type Changed”. TSE 15494 (rating 4): To address E13169, created new TC LMP/AUT/BV-35-C so that the Peripheral IUT properly handles the Lower Tester sending an SRES immediately after an AU_RAND. Updated TCMT accordingly. TSE 15561 (rating 4): To address an issue with a lack of LMP test cases for mutual legacy authentication in all role combinations, added a new “Legacy Authentication Procedures” section containing new TCs LMP/AUT/BV-38-C and -39-C. Updated TCMT accordingly. TSE 15589 (rating 1): To address Erratum 15531, globally changed “master clock” to “Central’s clock” and “current master clock” to “Central’s current clock”. TSE 15608 (rating 4): To address E15553, to confirm behavior when the Role Switch or Link Key Selection command either has no change or fails, added new TCs LMP/ENC/BV-57-C and -58-C and LMP/LIH/BV-149-C, and updated TCs LMP/LIH/BV-143-C and LMP/LIH/BV-03-C. Updated TCMT accordingly. TSE 15621 (rating 4): To address E12093, confirming that LMP rejects an SCO connection attempt with AES-CCM encryption, updated Initial Condition and MSC for the following TCs: LMP/ENC/BV-29-C, -30-C, and -37-C – -47-C, and LMP/LIH/BV-134-C, 136-C, 138-C, and 140-C. Added new TC LMP/LIH/BI-03-C. Updated TCMT accordingly. TSE 15936 (rating 4): To address E15834, LMP_CLK_ADJ not allowed in APB-C logical link, updated Test Procedure, test steps, and Pass Verdict for TCs LMP/XCL/BV-01-C and -03-C and added new TCs LMP/LIH/BV-150-C and LMP/XCL/BV-04-C. Updated TCMT accordingly. TSE 16109 (rating 3): Updated the Test Purpose and Reference sections and added an Inconclusive Verdict for TCs LMP/XCL/BV-01-C and -02-C. TSE 16484 (rating 1): Replaced MSCs for TCs LMP/ENC/BV-26-C and -33-C to fix a typo. TSE 17094 (rating 1): (Note: There are two CR files for this TSE.) Changed TCMT entries as follows: for LMP/AFH/BV-01-C – -03-C, changed to “LMP 26/2” from “LMP 2/16”; for LMP/AUT/BV-34-C, changed “LMP 2/19” to “LMP 2/19b”. Also changed TCMT entries as follows: for LMP/AFH/BV-06-C, added “Peripheral” to feature name and removed “LMP 26/3”; moved LMP/AFH/BV-05-C to its own row with appropriate item/feature info.

Publication Number	Revision Number	Date	Comments
			TSE 17095 (rating 1): Updated TCMT entry for LMP/AUT/BI-01-C to remove SUM ICS 21/9 and 21/13 and to correct styles throughout the TCMT to "SUM ICS" per updated template conventions. Incorporated Host_To_Controller_Encryption_Key_Control_Enhancements_TEST_CR_r06: Modified "Key_Size Negotiation as Slave" subsection of "Encryption" section by adding initial conditions, replacing the MSC, adding and updating test steps, updating heading text in the TC Config table, and adding a pass verdict (affects test cases LMP/ENC/BV-51-C, -52-C, -55-C, and -56-C; modified "Key_Size Negotiation as Master" subsection of "Encryption" section by adding initial conditions, replacing the MSC, adding and updating test steps, updating heading text in the TC Config table, and adding a pass verdict (affects test cases LMP/ENC/BV-53-C and -54-C). Updated TCMT. Template-related and consistency checker editorials.
40	p40	2021-07-13	Approved by BTI on 2021-06-27. Prepared for TCRL 2021-1 publication.
	p41r00–r09	2021-08-18 – 2021-12-29	TSE 16917 (rating 3): Updated test procedures and pass verdict for LMP/SAM/BV-01-C, test procedures for LMP/SAM/BI-03-C and -04-C, and test procedure and MSC for LMP/SAM/BV-09-C. TSE 17006 (rating 2): To address issues with test cases added under TSE 13240: deleted LMP/AUT/BI-05-C and LMP/LIH/BI-05-C; updated Initial Condition and MSC for LMP/ENC/BI-01-C and -02-C; and updated MSC and TCMT entry for LMP/LIH/BI-04-C. Updated TCMT accordingly. TSE 17161 (rating 4): To address the need for separate TCIDs for devices that support HCI Set Min Encryption Key Size, updated the MSC and test procedure: for the section containing LMP/ENC/BV-51-C, -52-C, -55-C, and -56-C, adding new tests LMP/ENC/BV-59-C – -62-C, and for the section containing LMP/ENC/BV-53-C and -54-C, adding new tests LMP/ENC/BV-63-C and -64-C. Updated TCMT accordingly. TSE 17209 (rating 2): Updated MSC and TCMT entry for TC LMP/ENC/BV-12-C. TSE 17217 (rating 2): Removed SUM ICS 23/3 from the TCMT and replaced it with LMP 1/2 for LMP/LIH/BV-03-C, -12-C, -20-C, -60-C, -71-C, -72-C, -83-C, and -116-C. Removed LMP/ENC/BV-03-C. TSE 17235 (rating 2): Updated initial condition and TCMT entries for TCs LMP/AUT/BI-06-C and -07-C. TSE 17299 (rating 2): Updated title, test purpose, initial condition, and MSC to fix an integration error in TC LMP/LIH/BV-149-C.

Publication Number	Revision Number	Date	Comments
			TSE 17382 (rating 1): Fixed typo in TC LMP/AUT/BV-24-C. TSE 17533 (rating 4): To address an issue with nugatory start and stop encryption, added new test case LMP/ENC/BI-03-C to the Encryption - Peripheral section and added new section Encryption - Both Connected Roles, with new test cases LMP/ENC/BI-04-C – -09-C. Updated TCMT accordingly. TSE 17535 (rating 3): To address an issue with rejected role switches, updated MSC and Pass verdict for LMP/LIH/BV-142-C – -144-C and -149-C and added role name to test description where needed. Performed editorial work, including making consistency checker fixes and aligning the copyright page with v2 of the DNMD.
41	p41	2022-01-25	Approved by BTI on 2021-12-27. Prepared for TCRL 2021-2 publication.
	p42r00–r05	2022-01-31 – 2022-05-12	TSE 18003 (rating 2): To correct the order of LMP_in_rand, replaced the MSC in the test procedure for LMP/AUT/BI-04-C. TSE 18178 (rating 2): Updated the initial condition, MSC, test procedure, and pass verdict for LMP/LIH/BV-19-C. TSE 18338 (rating 2): Updated the MSC for LMP/LIH/BV-151-C. TSE 18385 (rating 2): Added “Fields and Bits Reserved for Future Use” section. Template-related editorials.
42	p42	2022-06-28	Approved by BTI on 2022-05-31. Prepared for TCRL 2022-1 publication.
	p43r00–r09	2022-07-27 – 2022-12-05	TSE 18304 (rating 2): Updated Initial Condition, MSC, test steps and Pass verdict for LMP/ENC/BV-57-C and -58-C. TSE 18324 (rating 2): Combined LMP/LIH/BV-143-C and -149-C into one table-based section. Updated the TCMT accordingly. TSE 18412 (rating 2): Updated to remove the “Optional” aspects of BB 11/1 “Broadcast Messages” which are Mandatory to support. Updated MSCs, Test Conditions, and Pass verdicts for the following test cases: LMP/ENC/BV-01-C, -05-C, -22-C, -25-C, -26-C, -33-C, -34-C, and -50-C and LMP/ENC/BI-01-C and -02-C. TSE 18461 (rating 4): Added new test case LMP/LIH/BI-06-C; updated TCMT accordingly. TSE 18462 (rating 4): Added new sections with new TCs LMP/LIH/BI-07-C – -09-C. Updated TCMT accordingly. TSE 19141 (rating 4): Per E17532, added new TC LMP/LIH/BI-10-C; updated TCMT accordingly. Template-related editorials.

Publication Number	Revision Number	Date	Comments
43	p43	2023-02-07	Approved by BTI on 2022-12-28. Prepared for TCRL 2022-2 publication.
	p44r00	2023-04-05	TSE 22656 (rating 2): To address the incorrect use of the ID field in LMP_CLK_ADJ, updated the MSC and test steps and added a Test Condition to LMP/LIH/BI-09-C. Minor editorials to align with most recent template guidance.
44	p44	2023-06-29	Approved by BTI on 2023-06-05. Prepared for TCRL 2023-1 publication.
	p45r00–r04	2023-08-07 – 2024-05-09	TSE 23170 (rating 1): Per E23034, added an acronyms table for “SSR”. TSE 23307 (rating 2): Added a new section to the TSS for “HCI command and event version”. TSE 24051 (rating 3): Per E23579, updated the test purpose and MSC for LMP/LIH/BV-14-C and LMP/LIH/BV-57-C and the test purpose, test steps, and MSC for LMP/SAM/BV-01-C. Combined LMP/LIH/BV-100-C – -102-C into a table-based section with an updated test purpose, test steps, and MSC. TSE 24093 (rating 1): Replaced SUM ICS references in the TCMT with CORE ICS references, affecting LMP/AUT/BI-01-C. TSE 24979 (rating 4): Added new TC LMP/ENC/BV-65-C. Updated the TCMT accordingly. TSE 25144 (rating 1): Corrected the caption for the LMP/ENC/BV-44-C MSC.
45	p45	2024-07-01	Approved by BTI on 2024-05-22. Prepared for TCRL 2024-1 publication.
	p46r00–r05	2024-07-10 – 2024-07-25	TSE 22968 (rating 4): Per E24507 and E22495, added new TC LMP/AUT/BV-41-C. Updated the TCMT accordingly. TSE 24670 (rating 4): Per E15466, added new TC LMP/LIH/BV-152-C. Updated the TCMT accordingly. TSE 24683 (rating 4): To address EC 24617 (“Security changes - batch 1”), added new TCs LMP/AUT/BI-08-C, LMP/AUT/BV-40-C, and LMP/SP/BI-13-C and -14-C and modified the LMP/SP/BI-01-C rounds table to include “Generate valid public key and set x-coordinate same as IUT’s”; updated the TCMT accordingly. TSE 25388 (rating 2): Deleted LMP/LIH/BV-73-C and updated the Test Purpose, test steps, Initial Condition, and Pass verdict for LMP/LIH/BV-126-C; updated the TCMT accordingly. TSE 25389 (rating 2): Reorganized the TCMT to better group TCIDs related to SCO Links and to remove references to deleted LMP ICS Table 21.

Publication Number	Revision Number	Date	Comments
			TSE 25503 (rating 1): Removed LMP/ENC/BV-65-C; updated the TCMT accordingly. TSE 25510 (rating 1): Updated the MSC for the section containing LMP/LIH/BV-100-C – -102-C. Incorporated integration review feedback and made consistency checker editorial updates.
46	p46	2024-09-04	Approved by BTI on 2024-08-14. Prepared for TCRL 2024-2 publication.
	p47r00–r02	2024-10-31 – 2024-11-14	TSE 24803 (rating 2): Corrected the MSC and the Pass verdict for LMP/LIH/BV-83-C. TSE 24977 (rating 3): Updated the MSCs and the Pass verdict for LMP/SP/BV-12-C. TSE 24978 (rating 4): Added new TCs LMP/AUT/BV-42-C – -44-C to close a gap in testing. Updated the TCMT accordingly. TSE 25386 (rating 2): Updated the initial condition and the MSC and corrected capitalization in the test procedure and Pass verdict for the section containing LMP/LIH/BI-01-C and -02-C. Deleted LMP/LIH/BI-03-C. Updated the TCMT accordingly. TSE 25760 (rating 1): Corrected the Test Purpose wording for LMP/ENC/BV-57-C and LMP/ENC/BV-58-C. TSE 25924 (rating 1): Corrected the MSC and the Pass verdict for LMP/AUT/BI-01-C. TSE 26011 (rating 3): Corrected the test procedure/MSC and Pass verdict for LMP/ENC/BV-58-C. TSE 26024 (rating 2): Corrected the test procedure for LMP/AUT/BI-08-C, including removing the rounds table. TSE 26110 (rating 2): Combined LMP/SP/BI-02-C and -08-C, LMP/SP/BI-04-C and -10-C, and LMP/SP/BI-06-C and -12-C into table-based sections, with updated test procedure/MSC. TSE 26298 (rating 2): Per E24460, updated the MSC for LMP/AFH/BV-04-C. TSE 26377 (rating 2): Corrected the test procedure/MSC and Pass verdict for LMP/AUT/BV-40-C and added a Fail verdict. TSE 26429 (rating 2): Corrected the TCMT entry for LMP/AUT/BV-41-C.
47	p47	2025-02-18	Approved by BTI on 2024-12-26. Prepared for TCRL 2025-1 publication.

Publication Number	Revision Number	Date	Comments
	p48r00–r12	2025-01-27 – 2025-04-02	TSE 25194 (rating 2): To address an issue with HCI_Host_Buffer_Size not being needed, combined standalone tests LMP/LIH/BV-43-C – -46-C into a table-based section, standalone tests LMP/LIH/BV-104-C and -105-C into a table-based section, and standalone tests LMP/LIH/BV-53-C and -54-C into a table-based section, with updates to test name, test purpose, MSC, test procedure, and Pass verdict. Updated LMP/LIH/BV-55-C – -57-C; the section containing LMP/LIH/BV-100-C – -102-C; LMP/LIH/BV-103-C, -106-C, and -107-C; and LMP/LIH/BV-110-C – -113-C, with updates to the MSC and Pass verdict and with deletion of the Test Condition. Swapped out MSCs for LMP/LIH/BV-114-C and -115-C. TSE 25201 (rating 1): To accommodate ES-25131, renamed all “Connection Rejected” errors to “Rejected”. Additionally, updated instances of “package” to “packet” and corrected references format. TSE 26257 (rating 4): To accommodate ES-17738, updated the sections containing LMP/AUT/BV-25-C, -27-C, -29-C – -33-C and added new TCs LMP/AUT/BV-45-C – -51-C. Updated the TCMT accordingly. TSE 26974 (rating 3): Corrected the MSC and test procedure for TCs LMP/AUT/BV-42-C – -44-C. TSE 27419 (rating 2): Corrected the TCMT entry affecting LMP/ENC/BV-30-C and -47-C and combined -46-C, -48-C, and -49-C with the entry for -29-C, -31-C, and -32-C.
48	p48	2025-05-06	Approved by BTI on 2025-04-16. Prepared for TCRL 2025-2 publication.
	p49r00–r07	2025-07-21 – 2025-09-18	Converted the LMP TS into a multi-file document. Converted cross-references to plain text where an inter-file reference was involved and nonfunctional. TSE 25309 (rating 2): Revamped the LMP TS, primarily converting standalone TCs to table-driven tests where feasible. TSE 25310 (rating 2): Updated PDU names throughout the LMP TS to better align with the latest Core spec. TSE 27094 (rating 1): Corrected the TCMT entries for LMP/ENC/BI-05-C, -06-C, -08-C, and -09-C. TSE 27117 (rating 4): To accommodate the changes needed for E24489, added new TCs LMP/SP/BV-67-C and -68-C and a reference to Core v6.2. Updated the TCMT accordingly. TSE 27257 (rating 2): Replaced BB entries in the TCMT with LMP items where possible. TSE 27589 (rating 3): Updated the test steps and MSC for LMP/AUT/BV-42-C.

Publication Number	Revision Number	Date	Comments
			TSE 28050 (rating 4): As part of TSE 25309 overhaul, added new TCs LMP/AUT/BV-52-C – -57-C and LMP/INF/BV-23-C and -24-C. Updated the TCMT accordingly.
49	p49	2025-11-04	Approved by BTI on 2025-10-05. Prepared for TCRL pkg101 publication.
	p50r00–r11	2025-11-25 – 2026-03-09	TSE 18890 (rating 2): To support the changes needed for ES-23102, updated the MSC and/or test steps of LMP/AUT/BV-03-C; the section containing LMP/AUT/BV-04-C and -52-C; the section containing LMP/AUT/BV-06-C and -53-C; LMP/AUT/BV-24-C; the section containing LMP/AUT/BV-25-C, -54-C, -45-C, and -55-C; the section containing LMP/AUT/BV-27-C, -56-C, -46-C, and -57-C; LMP/AUT/BV-12-C; LMP/AUT/BV-13-C; LMP/AUT/BV-28-C; the section containing LMP/AUT/BV-29-C and -47-C; and LMP/SP/BV-05-C. TSE 24021 (rating 3): In support of ES-24402, updated the test steps for LMP/AUT/BV-42-C – -44-C. TSE 25387 (rating 2): Removed LMP/LIH/BV-134-C, -136-C, -138-C, and -140-C and updated the TCMT accordingly. These tests are to be repurposed under the HCI layer. TSE 27120 (rating 4): To support ES-26039, add new TCs LMP/ENC/BV-66-C – -90-C. Updated the TCMT accordingly. TSE 27801 (rating 3): Corrected the MSC, test steps, and Pass verdict for LMP/AUT/BI-08-C. TSE 27873 (rating 3): To accommodate ES-26043, added a line item with “The IUT always generates a unique public key.” to the Pass verdicts of LMP/SP/BV-06-C, -07-C, -12-C, and -13-C; added that same line to the Pass verdict of the section containing LMP/SP/BV-18-C, -20-C, -22-C, and -70-C – -72-C and the section containing LMP/SP/BV-19-C, -21-C, -23-C, and -73-C – -75-C along with associated test procedure and MSC updates. TSE 28002 (rating 3): Replaced the MSC for LMP/ENC/BV-58-C. TSE 28192 (rating 2): Corrected MSC and test step for the section containing LMP/AUT/BV-32-C and -50-C. TSE 28193 (rating 2): Corrected MSC, test step, and Pass verdict for the section containing LMP/ENC/BV-01-C, -26-C, and -33-C. TSE 28194 (rating 2): Corrected MSC and test steps for the section containing LMP/ENC/BV-16-C and -41-C.

Publication Number	Revision Number	Date	Comments
			TSE 28343 (rating 2): To accommodate changes needed for ES-17791, for the section containing LMP/SP/BV-18-C, -20-C, and -22-C, added a Random Key column to the TCC, added a version number to the TC descriptions, updated the test steps and the Pass verdict, and added new TCs under TSE 28811; for the section containing LMP/SP/BV-19-C, -21-C, and -23-C, added a Random Key column to the TCC, added a version number to the TC descriptions, updated the test steps and the Pass verdict, and added new TCs under TSE 28811; for the section containing LMP/SP/BV-54-C, -56-C, and -58-C, added a Random Key column to the TCC, added a version number to the TC descriptions, updated the test steps, MSCs, and the Pass verdict, and added new TCs under TSE 28811; for the section containing LMP/SP/BV-55-C, -57-C, and -59-C, added a Random Key column to the TCC, added a version number to the TC descriptions, updated the test steps, MSCs, and the Pass verdict, and added new TCs under TSE 28811. Updated the TCMT for existing test cases to reflect version. TSE 28350 (rating 4): Per ES-27464, added new TC LMP/SP/BV-69-C. Updated the TCMT accordingly. TSE 28476 (rating 1): Changed “host” to “Host” where necessary. TSE 28506 (rating 2): Updated the TCMT logic for LMP/INF/BV-22-C. TSE 28811 (rating 4): This TSE captures only Category 4 changes made for the otherwise Category 2 TSE 28343. Added new TCs LMP/SP/BV-70-C – -81-C. Updated the TCMT accordingly.
50	p50	2026-05-05	Approved by BTI on 2026-04-07. Prepared for TCRL pkg103 publication.

Acknowledgments

Name	Company
Joakim Linde	Apple
Dejan Berec	Bluetooth SIG, Inc.
Nathan Burns	Bluetooth SIG, Inc.
Gene Chang	Bluetooth SIG, Inc.
Jeff Drake	Bluetooth SIG, Inc.
Alicia Courtney	Broadcom
Prasanna Desai	Broadcom
Shawn Ding	Broadcom
Steven Hall	Broadcom
Farooq Hameed	Broadcom
Robert Hulvey	Broadcom

Name	Company
Knut Odman	Broadcom
Angel Polo	Broadcom
Erik Rivard	Broadcom
Mayank Batra	CSR
Joe Decuir	CSR
Tim Howes	CSR
Ian Jones	CSR
Sean Mitchell	CSR
Ross O'Connor	CSR
Steven Singer	CSR
Dishant Srivastava	CSR
Jonathan Tanner	CSR
Steven Wenham	CSR
Fabien Duvoux	Ellisys
Kyle Penri-Williams	Ellisys
Clement Vacheron	Ellisys
Leif Wilhelmsson	Ericsson
Magnus Eriksson	Intel
Marcel Holtmann	Intel
Sharon Yang	Intel
Josselin de la Broise	Marvell
L. C. Ko	MediaTek
Huanchun Ye	MediaTek
Krishna Singala	Mindtree
Lily Chen	NIST
Kaisa Nyberg	Nokia
Tsuyoshi Okada	Panasonic Corporation
Niclas Granqvist	Polar
Olaf Hirsch	Qualcomm Atheros
Joel Linsky	Qualcomm Atheros
Cameron McDonald	Qualcomm Atheros
Brian Redding	Qualcomm Atheros
Magnus Sommansson	Qualcomm Technologies International, Ltd.
Jean-Philippe Lambert	RivieraWaves
Rasmus Abildgren	Samsung Electronics
Clive D. W. Feather	Samsung Electronics
Kyong-Sok Seo	Samsung Electronics Co. Ltd
Andrew Estrada	Sony Corporation
Masahiko Seki	Sony Corporation
Jorgen van Parijs	ST Ericsson
Yves Wernaers	ST-Ericsson

Name	Company
Alon Cheifetz	Texas Instruments
Alon Paycher	Texas Instruments
Rod Kimmell	X6D, Inc.