Security Manager Protocol (SM)

Bluetooth® Test Suite

- **Revision**: SM.TS.p18
- **Revision Date**: 2020-01-07
- **Group Prepared By**: BTI
- **Feedback Email**: bti-main@bluetooth.org
Security Manager Protocol (SM) / Test Suite

Contents

1 Scope ............................................................................................................................................... 7

2 References, Definitions, and Abbreviations ................................................................................. 8

2.1 References ...................................................................................................................................... 8

3 Test Suite Structure (TSS) ................................................................................................................ 9

3.1 Test Strategy .................................................................................................................................... 9

3.2 Test Groups ..................................................................................................................................... 10

3.2.1 Protocol ....................................................................................................................................... 10

3.2.2 STK Pairing Method .................................................................................................................. 10

3.2.3 Signing ......................................................................................................................................... 10

3.2.4 Encryption Key Size .................................................................................................................. 10

3.2.5 Key Distribution and Usage ....................................................................................................... 10

3.2.6 Slave Initiated Security ............................................................................................................. 10

3.2.7 LE Secure Connections Pairing .................................................................................................. 10

4 Test Cases ......................................................................................................................................... 11

4.1 Introduction ..................................................................................................................................... 11

4.1.1 Test Case Identification Conventions ....................................................................................... 11

4.1.2 Conformance .............................................................................................................................. 11

4.1.3 Pass/Fail Verdict Conventions .................................................................................................. 12

4.2 Setup Preambles ............................................................................................................................. 12

4.2.1 Security Manager Channel over L2CAP .................................................................................... 12

4.3 Protocol ......................................................................................................................................... 13

4.3.1 SMP Timeout ............................................................................................................................... 13

4.3.1.1 SM/MAS/PROT/BV-01-C [SMP Time Out – IUT Initiator] .................................................. 13

4.3.1.2 SM/SLA/PROT/BV-02-C [SMP Time Out – IUT Responder] .............................................. 14

4.4 STK Pairing Method ....................................................................................................................... 14

4.4.1 Just Works ................................................................................................................................... 14

4.4.1.1 SM/MAS/JW/BV-01-C [Just Works IUT Initiator – Success] .................................................. 14

4.4.1.2 SM/SLA/JW/BV-02-C [Just Works IUT Responder – Success] .............................................. 15

4.4.1.3 SM/SLA/JW/BI-03-C [Just Works IUT Responder – Handle AuthReq flag RFU correctly] .... 16

4.4.1.4 SM/MAS/JW/BV-05-C [Just Works, IUT Initiator – Pairing Failed] ..................................... 17

4.4.1.5 SM/MAS/JW/BI-04-C [Just Works IUT Initiator – Handle AuthReq flag RFU correctly] ....... 18

4.4.1.6 SM/MAS/JW/BI-01-C [Just Works, IUT Initiator – Failure] .................................................. 19

4.4.1.7 SM/SLA/JW/BI-02-C [Just Works, IUT Responder – Failure] ................................................. 20

4.4.2 Passkey Entry (PKE) .................................................................................................................. 20

4.4.2.1 SM/MAS/PKE/BV-01-C [Passkey Entry, IUT Initiator – Success] ........................................... 20

4.4.2.2 SM/SLA/PKE/BV-02-C (Passkey Entry, IUT Responder – Success) .................................... 21

4.4.2.3 SM/MAS/PKE/BV-04-C (Passkey Entry, IUT Initiator – results in Unauthenticated Success) ... 22

4.4.2.4 SM/SLA/PKE/BV-05-C [Passkey Entry, IUT Responder – Lower Tester has insufficient security for Passkey Entry] .............................................................. 22

4.4.2.5 SM/MAS/PKE/BI-01-C [Passkey Entry, IUT Initiator – Failure on Responder Side] .......... 24

4.4.2.6 SM/MAS/PKE/BI-02-C [Passkey Entry, IUT Initiator – Interrupted passkey entry by Responder Side] 24

4.4.2.7 SM/SLA/PKE/BI-03-C [Passkey Entry, IUT Responder – Failure on Initiator Side] ............. 25

4.4.3 Out of Band (OOB) .................................................................................................................... 26

4.4.3.1 SM/MAS/OOB/BV-01-C [IUT Initiator – Both sides have OOB data – Success] .................. 26

4.4.3.2 SM/SLA/OOB/BV-02-C [IUT Responder – Both sides have OOB data – Success] ............. 27

4.4.3.3 SM/MAS/OOB/BV-03-C [IUT Initiator – Only IUT has OOB data – Success] .................... 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7.2</td>
<td>Re-encrypt an encrypted link with LTK</td>
</tr>
<tr>
<td>4.7.2.1</td>
<td>SM/SLA/KDU/BV-07-C [IUT Responder - Existing encrypted link is re-encrypted using LTK]</td>
</tr>
<tr>
<td>4.8</td>
<td>Slave Initiated Security Request</td>
</tr>
<tr>
<td>4.8.1</td>
<td>Slave Initiated Pairing</td>
</tr>
<tr>
<td>4.8.1.1</td>
<td>SM/SLA/SIP/BV-01-C [Slave Initiates pairing]</td>
</tr>
<tr>
<td>4.8.1.2</td>
<td>SM/MAS/SIP/BV-02-C [Slave Initiates pairing – Master Response]</td>
</tr>
<tr>
<td>4.8.2</td>
<td>Slave Initiates Encryption</td>
</tr>
<tr>
<td>4.8.2.1</td>
<td>SM/SLA/SIE/BV-01-C [Slave initiates Encryption]</td>
</tr>
<tr>
<td>4.9</td>
<td>Pairing Methods Using LE Secure Connections</td>
</tr>
<tr>
<td>4.9.1</td>
<td>Just Works (SCJW)</td>
</tr>
<tr>
<td>4.9.1.1</td>
<td>SM/MAS/SCJW/BV-01-C [Just Works, IUT Initiator, Secure Connections – Success]</td>
</tr>
<tr>
<td>4.9.1.2</td>
<td>SM/SLA/SCJW/BV-02-C [Just Works, IUT Responder, Secure Connections – Success]</td>
</tr>
<tr>
<td>4.9.1.3</td>
<td>SM/SLA/SCJW/BV-03-C [Just Works, IUT Responder, Secure Connections – Handle AuthReq Flag RFU Correctly]</td>
</tr>
<tr>
<td>4.9.1.4</td>
<td>SM/MAS/SCJW/BV-04-C [Just Works, IUT Initiator, Secure Connections – Handle AuthReq Flag RFU Correctly]</td>
</tr>
<tr>
<td>4.9.1.5</td>
<td>SM/MAS/SCJW/BI-01-C [Just Works, IUT Initiator, Secure Connections – Pairing Failed]</td>
</tr>
<tr>
<td>4.9.1.6</td>
<td>SM/SLA/SCJW/BI-02-C [Just Works, IUT Responder, Secure Connections – Confirm Check Failure]</td>
</tr>
<tr>
<td>4.9.2</td>
<td>Passkey Entry (SCPK)</td>
</tr>
<tr>
<td>4.9.2.1</td>
<td>SM/MAS/SCPK/BV-01-C [Passkey Entry, IUT Initiator, Secure Connections – Success]</td>
</tr>
<tr>
<td>4.9.2.2</td>
<td>SM/SLA/SCPK/BV-02-C [Passkey Entry, IUT Responder, Secure Connections – Success]</td>
</tr>
<tr>
<td>4.9.2.3</td>
<td>SM/SLA/SCPK/BV-03-C [Passkey Entry, IUT Responder, Secure Connections – Handle AuthReq Flag RFU Correctly]</td>
</tr>
<tr>
<td>4.9.2.4</td>
<td>SM/MAS/SCPK/BV-04-C [Passkey Entry, IUT Initiator, Secure Connections – Handle AuthReq Flag RFU Correctly]</td>
</tr>
<tr>
<td>4.9.2.5</td>
<td>SM/MAS/SCPK/BI-01-C [Passkey Entry, IUT Initiator, Secure Connections – Pairing Failed]</td>
</tr>
<tr>
<td>4.9.2.6</td>
<td>SM/MAS/SCPK/BI-02-C [Passkey Entry, IUT Initiator, Secure Connections – Failure]</td>
</tr>
<tr>
<td>4.9.2.7</td>
<td>SM/SLA/SCPK/BI-03-C [Passkey Entry, IUT Responder, Secure Connections – Confirm Value Check Failure]</td>
</tr>
<tr>
<td>4.9.2.8</td>
<td>SM/SLA/SCPK/BI-04-C [Passkey Entry, IUT Responder, Secure Connections – Pairing Failed]</td>
</tr>
<tr>
<td>4.9.3</td>
<td>Out of Band (SCOB)</td>
</tr>
<tr>
<td>4.9.3.1</td>
<td>SM/MAS/SCOB/BV-01-C [Out of Band, IUT Initiator, Secure Connections – Success]</td>
</tr>
<tr>
<td>4.9.3.2</td>
<td>SM/SLA/SCOB/BV-02-C [Out of Band, IUT Responder, Secure Connections – Success]</td>
</tr>
<tr>
<td>4.9.3.3</td>
<td>SM/SLA/SCOB/BV-03-C [Out of Band, IUT Responder, Secure Connections – Handle AuthReq Flag RFU Correctly]</td>
</tr>
<tr>
<td>4.9.3.4</td>
<td>SM/MAS/SCOB/BV-04-C [Out of Band, IUT Initiator, Secure Connections – Handle AuthReq Flag RFU Correctly]</td>
</tr>
<tr>
<td>4.9.3.5</td>
<td>SM/MAS/SCOB/BI-01-C [Out of Band, IUT Initiator, Secure Connections – Failure]</td>
</tr>
<tr>
<td>4.9.3.6</td>
<td>SM/SLA/SCOB/BI-02-C [Out of Band, IUT Responder, Secure Connections – Failure]</td>
</tr>
<tr>
<td>4.9.3.7</td>
<td>SM/SLA/SCOB/BI-03-C [Out of Band, IUT Responder, Secure Connections – Pairing Failed]</td>
</tr>
<tr>
<td>4.9.3.8</td>
<td>SM/MAS/SCOB/BI-04-C [Out of Band, IUT Initiator, Secure Connections – Pairing Failed]</td>
</tr>
<tr>
<td>4.9.4</td>
<td>Cross Transport Key Derivation (SCCT)</td>
</tr>
<tr>
<td>4.9.4.1</td>
<td>SM/MAS/SCCT/BV-01-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections – Derive LE LTK from BR/EDR Link Key]</td>
</tr>
<tr>
<td>4.9.4.2</td>
<td>SM/SLA/SCCT/BV-02-C [Cross Transport Key Derivation, IUT Responder, Secure Connections – Derive LE LTK from BR/EDR Link Key]</td>
</tr>
<tr>
<td>4.9.4.3</td>
<td>SM/MAS/SCCT/BV-03-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections – Derive LE LTK from BR/EDR Link Key Using h6]</td>
</tr>
<tr>
<td>4.9.4.4</td>
<td>SM/SLA/SCCT/BV-04-C [Cross Transport Key Derivation, IUT Responder, Secure Connections – Derive LE LTK from BR/EDR Link Key Using h6]</td>
</tr>
</tbody>
</table>
4.9.4.5 SM/MAS/SCCT/BV-05-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections –
Derive LE LTK from BR/EDR Link Key Using h7] ........................................................................76
4.9.4.6 SM/SLA/SCCT/BV-06-C [Cross Transport Key Derivation, IUT Responder, Secure Connections –
Derive LE LTK from BR/EDR Link Key Using h7] ........................................................................76
4.9.4.7 SM/MAS/SCCT/BV-07-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections –
Derive BR/EDR Link Key from LE LTK Using h6] ........................................................................77
4.9.4.8 SM/SLA/SCCT/BV-08-C [Cross Transport Key Derivation, IUT Responder, Secure Connections –
Derive BR/EDR Link Key from LE LTK Using h6] ........................................................................78
4.9.4.9 SM/MAS/SCCT/BV-09-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections –
Derive BR/EDR Link Key from LE LTK Using h7] ........................................................................78
4.9.4.10 SM/SLA/SCCT/BV-10-C [Cross Transport Key Derivation, IUT Responder, Secure Connections –
Derive BR/EDR Link Key from LE LTK Using h7] ........................................................................79

5 Test Case Mapping ..........................................................................................................................80

6 Revision History and Contributors ..............................................................................................85
1 Scope

This Bluetooth document contains the Test Suite Structure (TSS) and Test Cases (TC) to test the Security Manager Protocol specification.

The objective of this test suite is to provide a basis for interoperability for Bluetooth devices giving a high probability of air interface interoperability between different manufacturers' Bluetooth devices.
2 References, Definitions, and Abbreviations

2.1 References

This Bluetooth document incorporates, by dated or undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For the purpose of this Bluetooth document, the definitions and abbreviations in [1], [2], and [3] apply.

[7] Implementation eXtra Information for Test (IXIT) for Security Manager
[9] Erratum 10734: Pairing Updates
3 Test Suite Structure (TSS)

3.1 Test Strategy

The test objectives are to verify functionality of the Security Manager within a LE Host and enable interoperability between LE Hosts on different devices. The testing approach is to cover mandatory and optional requirements in the protocol specification and to match these to the support of the IUT as described in the ICS proforma.

Conformance testing is the appropriate test method to meet these intents. The basis for the test approach is the general concepts and conformance testing principles defined in ISO/IEC 9646-1 and ISO/IEC 9646-2, both part of the OSI Conformance Testing Methodology and Framework (CTMF).

The conformance test equipment shall provide an implementation of the LE Controller and the parts of the Host needed to perform these test cases. For some test cases, it is necessary to stimulate the IUT from an Upper Tester. In practice, this may be implemented as a special test interface, an MMI, or another interface supported by the IUT.

The Security Manager Protocol test suite contains Valid Behavior (BV) tests complemented with Invalid Behavior (BI) tests where required. The test coverage mirrored in the test suite structure is the result of a process that started with catalogued specification requirements that were logically grouped and assessed for testability enabling coverage in defined test cases.

The test suite structure is a tree with the first level representing the protocol groups.

- Protocol
  - SMP Timeout
- STK Pairing Method
  - Just Works
  - Passkey Entry
  - Out of Band
- Encryption Key Size
  - Signing
    - Master Signing
    - Slave Signing
- Key Distribution and Usage
  - Key Distribution During Bonding
  - Re-encrypt an Encrypted Link with LTK
- Slave Initiated Security
Pairing Methods using LE Secure Connections
- Just works and Numeric Comparison
- Passkey Entry
  - Out of Band
  - Cross Transport Key Derivation

### 3.2 Test Groups

The following test groups have been defined.

#### 3.2.1 Protocol

Test the correct implementation of SMP timeout.

#### 3.2.2 STK Pairing Method

Test the correct implementation of the Just Work, Passkey and OOB pairing modes.

#### 3.2.3 Signing

Test the correct implementation of the Signing algorithm and CSRK distribution.

#### 3.2.4 Encryption Key Size

Test the correct implementation of Key Size negotiation.

#### 3.2.5 Key Distribution and Usage

Test the correct implementation of the Key distribution protocol.

#### 3.2.6 Slave Initiated Security

Within this test group the correct implementation of slave initiated security is tested.

#### 3.2.7 LE Secure Connections Pairing

Test the correct implementation of the LE Secure Connections Just Works/Numeric Comparison, Passkey, OOB pairing modes and Cross Transport Key Derivation.
4 Test Cases

4.1 Introduction

4.1.1 Test Case Identification Conventions

Test cases shall be assigned unique identifiers per the conventions in [2]. The convention used here is `<spec abbreviation>/<IUT role>/<class>/<feat>/<func>/<subfunc>/<cap>/<xx>-<nn>-<y>`.

Bolded ID parts shall appear in the order prescribed. Non-bolded ID parts (if applicable) shall appear between the bolded parts. The order of the non-bolded parts may vary from test suite to test suite, but shall be consistent within each individual test suite.

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Spec Identifier &lt;spec abbreviation&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM</td>
<td>Security Manager Protocol</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Role Identifier &lt;IUT role&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAS</td>
<td>Master Role</td>
</tr>
<tr>
<td>SLA</td>
<td>Slave Role</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identifier Abbreviation</th>
<th>Feature Identifier &lt;feat&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROT</td>
<td>Protocol</td>
</tr>
<tr>
<td>JW</td>
<td>Just Works</td>
</tr>
<tr>
<td>PKE</td>
<td>Passkey Entry</td>
</tr>
<tr>
<td>OOB</td>
<td>Out Of Band</td>
</tr>
<tr>
<td>SIGN</td>
<td>Signing</td>
</tr>
<tr>
<td>EKS</td>
<td>Encryption Key Size</td>
</tr>
<tr>
<td>SCJW</td>
<td>LE Secure Connections Numeric Comparison (including Just Works)</td>
</tr>
<tr>
<td>SCPK</td>
<td>LE Secure Connections Passkey Entry</td>
</tr>
<tr>
<td>SCOB</td>
<td>LE Secure Connections Out-of-Band</td>
</tr>
<tr>
<td>SCCT</td>
<td>LE Secure Connections Cross Transport Key Derivation</td>
</tr>
</tbody>
</table>

Table 4.1: TC Feature Naming Convention for SM

4.1.2 Conformance

When conformance is claimed, all capabilities indicated as mandatory for this Specification shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional
capabilities for which support is indicated, are subject to verification as part of the Bluetooth Qualification Program.

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 that is not excluded by the Specification, OR
- That capabilities enabled by the implementations are sustained over durations expected by the use case, OR
- That the implementation gracefully handles any quantity of data expected by the use case, OR
- That in cases where more than one valid interpretation of the Specification exist, the implementation complies with at least one interpretation and gracefully handles other interpretations, OR
- That the implementation is immune to attempted security exploits.

A single execution of each of the required tests is required in order to constitute a pass verdict. However, it is noted that in order 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 Generator, 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 errata request such that the issue may be addressed.

4.1.3 Pass/Fail Verdict Conventions

Each test case has an Expected Outcome section, which outlines all the detailed pass criteria conditions that shall be met by the IUT to merit a Pass Verdict.

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 one of the pass criteria conditions cannot be met. If this occurs the outcome of the test shall be the Fail Verdict.

4.2 Setup Preambles

The procedures defined in this section are provided for information, as they are used by test equipment in achieving the Initial Conditions in certain tests.

4.2.1 Security Manager Channel over L2CAP

- Reference
  
  [1] 3.2
  
  [6] 2.1
• Preamble Procedure
  Establish an LE transport connection between the IUT and the Lower Tester.

  Establish the Security Manager Channel over L2CAP fixed channel 0x0006 between the IUT and the Lower Tester over the LE transport.

4.3 Protocol
The test group objective is to verify the correct implementation of the SMP timeout protocol.

4.3.1 SMP Timeout
4.3.1.1 SM/MAS/PROT/BV-01-C [SMP Time Out – IUT Initiator]
• Test Purpose
  Verify that the IUT handles the lack of pairing response after 30 seconds when acting as initiator.

• Reference
  [1] 3.4

• Initial Condition
  Preamble has been executed.
  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits pairing request.
  2. Lower Tester does not respond to this pairing request.
  3. IUT timeout after 30 seconds and the procedure shall be considered to have failed.
  4. The IUT reports the failure to the Upper Tester.
  5. After additionally (at least) 10 seconds the Lower Tester responds to the pairing request.
  6. The IUT closes the connection before receiving the delayed response or does not respond to it when it is received.

• Expected Outcome
  Pass Verdict
  IUT notifies Upper Tester after the 30 seconds timeout.
  The IUT does not respond to a delayed response after the timeout, as there should be no more transactions on the channel. Alternatively the IUT does not respond to a delayed response after the timeout.

• Notes
  After the Upper Tester is alerted, the channel shall not be used until the link is reconnected.
4.3.1.2  SM/SLA/PROT/BV-02-C [SMP Time Out – IUT Responder]

- Test Purpose
  Verify that the IUT responder disconnects the link if pairing does not follow Pairing Feature Exchange within 30 seconds after receiving Pairing Request command.

- Reference
  [1] 3.4

- Initial Condition
  Preamble has been executed.
  IUT is slave. Lower Tester is master.

- Test Procedure
  1. Lower Tester transmits Pairing Request.
  2. IUT responds with Pairing Response.
  3. In phase 2, Lower Tester does not issue the expected Pairing Confirm.
  4. IUT times out 30 seconds after issued Pairing Response and reports the failure to the Upper Tester.
  5. After additionally (at least) 10 seconds the Lower Tester issues the expected Pairing Confirm.
  6. The IUT closes the connection before receiving the delayed response or does not respond to it when it is received.

- Expected Outcome
  Pass Verdict
  IUT notifies Upper Tester after the 30 seconds timeout.
  The IUT does not respond to a delayed Pairing Confirm after the timeout, as there should be no more transactions on the channel. Alternatively the IUT does not respond to a delayed response after the timeout.

4.4  STK Pairing Method
The test group objective is to verify the correct implementation of the Just Works, Passkey Entry and Out of Band pairing methods.

4.4.1  Just Works

4.4.1.1  SM/MAS/JW/BV-01-C [Just Works IUT Initiator – Success]

- Test Purpose
  Verify that the IUT performs the Just Works pairing procedure correctly as master, initiator when both sides do not require MITM protection.

- Reference
  [1] 2.3.5.1, 2.3.5.2, 5.3.1, 5.3.2
• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits Pairing Request command with:
   a. IO capability set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq Bonding Flags set to ‘00’ and the MITM flag set to ‘0’ and all the reserved bits are set to ‘0’

2. Lower Tester responds with a Pairing Response command, with:
   a. IO capability set to “KeyboardDisplay”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq Bonding Flags set to ‘00’, and the MITM flag set to ‘0’ and all the reserved bits are set to ‘0’

3. IUT and Lower Tester perform phase 2 of the just works pairing procedure and establish an encrypted link with the key generated in phase 2.

• Expected Outcome

Pass Verdict

The encryption procedure initiated by the IUT completes successfully.

The IUT can encrypt the link successfully.

4.4.1.2 SM/SLA/JW/BV-02-C [Just Works IUT Responder – Success]

• Test Purpose

Verify that the IUT is able to perform the Just Works pairing procedure correctly when acting as slave, responder.

• Reference

[1] 2.3.5.2, 2.4.6, 5.3.1 & 5.3.2

• Initial Condition

Preamble has been executed.

IUT is slave. Lower Tester is master.

• Test Procedure

1. Lower Tester transmits Pairing Request command with:
   a. IO capability set to “NoInputNoOutput”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
2. IUT responds with a Pairing Response command, with:
   a. IO capability set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)

3. IUT and Lower Tester perform phase 2 of the just works pairing and establish an encrypted link with the generated STK.

• Expected Outcome

   Pass Verdict

   The encryption procedure initiated by the master completes successfully.

   The master can encrypt the link successfully.

4.4.1.3 SM/SLA/JW/BI-03-C [Just Works IUT Responder – Handle AuthReq flag RFU correctly]

• Test Purpose

   Verify that the IUT is able to perform the Just Works pairing procedure when receiving additional bits set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as slave, responder.

• Reference

   [1] 2.3.5.2, 2.4.6, 5.3.1 & 5.3.2

• Initial Condition

   Preamble has been executed.

   IUT is slave. Lower Tester is master.

• Test Procedure

   1. Lower Tester transmits Pairing Request command with:
      a. IO Capability set to "NoInputNoOutput"
      b. OOB data flag set to 0x00 (OOB Authentication data not present)
      c. MITM set to '0' and all reserved bits are set to '1'

   2. IUT responds with a Pairing Response command, with:
      a. IO Capability set to any IO capability
      b. OOB data flag set to 0x00 (OOB Authentication data not present)
      c. All reserved bits are set to '0'

   3. IUT and Lower Tester perform phase 2 of the Just Works pairing and establish an encrypted link with the generated STK.
• Expected Outcome

Pass Verdict

The encryption procedure initiated by the Lower Tester completes successfully.

The Lower Tester can encrypt the link successfully.

4.4.1.4 SM/MAS/JW/BV-05-C [Just Works, IUT Initiator – Pairing Failed]

• Test Purpose

Verify that the IUT handles Just Works pairing failures.

• Reference

[1] 3.5.5

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits Pairing Request command with:
   a. IO capability is set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. All reserved bits are set to ‘0’

2. Lower Tester responds with a Pairing Failed command with reason code ‘0x03’ (Authentication Requirements).
3. The pairing process is aborted. IUT reports the failure to the Upper Tester.
   Run preamble to re-establish Initial Conditions

4. Execute Step 1.
5. Lower Tester responds with a Pairing Failed command with reason code ‘0x08’ (Unspecified Reason).
6. The pairing process is aborted. IUT reports the failure to the Upper Tester.
   Run preamble to re-establish Initial Conditions

7. Execute Step 1.
8. Lower Tester responds with a Pairing Failed command with reason code ‘0x05’ (Pairing Not Supported).
9. The pairing process is aborted. IUT reports the failure to the Upper Tester.
   Run preamble to re-establish Initial Conditions

10. Execute Step 1.
11. Lower Tester responds with a Pairing Failed command with reason code ‘0x09’ (Repeated Attempts).
12. The pairing process is aborted. IUT reports the failure to the Upper Tester.
• Expected Outcome

Pass Verdict

For each pairing failure, the IUT detects the failures reported by the responder and responds correctly to the Lower Tester.

For each pairing failure, the IUT aborts the pairing process and reports the failure to the Upper Tester.

4.4.1.5 SM/MAS/JW/BI-04-C [Just Works IUT Initiator – Handle AuthReq flag RFU correctly]

• Test Purpose

Verify that the IUT is able to perform the Just Works pairing procedure when receiving additional bits set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as master, initiator.

• Reference

[1] 2.3.5.2, 2.4.6, 5.3.1 & 5.3.2

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with:
   a. IO Capability set to any IO Capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. All reserved bits are set to ‘0’. For the purposes of this test the Secure Connections bit and the Keypress bits in the AuthReq bonding flag set by the IUT are ignored by the Lower Tester.

2. Lower Tester responds with a Pairing Response command, with:
   a. IO Capability set to “NoInputNoOutput”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq bonding flag set to the value indicated in the IXIT [7] for ‘Bonding Flags’ and the MITM flag set to ‘0’ and all reserved bits are set to ‘1’. The SC and Keypress bits in the AuthReq bonding flag are set to 0 by the Lower Tester for this test.

3. IUT and Lower Tester perform phase 2 of the Just Works pairing and establish an encrypted link with the generated STK.
• Expected Outcome

Pass Verdict

The encryption procedure initiated by the IUT completes successfully.

The link is encrypted successfully.

4.4.1.6 SM/MAS/JW/BI-01-C [Just Works, IUT Initiator – Failure]

• Test Purpose

Verify that the IUT handles Just Works pairing failure as initiator correctly.

• Reference

[1] 2.3.5.1, 2.3.5.2, 5.3.5.7

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with:
   a. IO capability is set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. All reserved bits are set to ‘0’

2. Lower Tester responds with a Pairing Response command with:
   a. IO capability set to “NoInputNoOutput”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq Bonding Flags set to ‘01’ and the MITM flag set to ‘0’ and all reserved bits are set to ‘0’

3. During phase 2 of the pairing procedure, the Lower Tester transmits a Pairing Confirm command with an incorrect Sconfirm value.

4. The IUT transmits a Pairing Failed command with Reason set to ‘Confirm Value Failed’ after receiving the Srandom and detecting the Sconfirm is incorrect.

5. The Lower Tester disconnects the link.

• Expected Outcome

Pass Verdict

IUT detects the incorrect confirm values and responds to the Lower Tester accordingly.
4.4.1.7 SM/SLA/JW/BI-02-C [Just Works, IUT Responder – Failure]

- Test Purpose
  Verify that the IUT handles just works pairing failure as responder correctly.

- Reference
  [1] 2.3.5.1, 2.3.5.2

- Initial Condition
  Preamble has been executed.

  IUT is slave. Lower Tester is master.

- Test Procedure
  1. Lower Tester transmits a Pairing Request command with:
     a. IO capability set to "NoInputNoOutput"
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. AuthReq bonding flag set to ‘01’, and the MITM flag set to ‘0’ and all reserved bits are set to ‘0’
  2. IUT responds with a Pairing Response command, with:
     a. IO capability set to any IO capability
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. All reserved bits are set to ‘0’
  3. During phase 2 of the just works pairing procedure, the Lower Tester transmits a Pairing Confirm command with an incorrect Mconfirm Value.
  4. IUT transmits a Pairing Failed command with Reason set to ‘Confirm Value Failed’ after receiving the Mrand and detecting the Mconfirm is incorrect.

- Expected Outcome
  Pass Verdict

  IUT detects the incorrect confirm value responds correctly to the Lower Tester.

4.4.2 Passkey Entry (PKE)

4.4.2.1 SM/MAS/PKE/BV-01-C [Passkey Entry, IUT Initiator – Success]

- Test Purpose
  Verify that the IUT performs the Passkey Entry pairing procedure correctly as initiator.

- Reference
  [1] 2.3.5.3, 5.3.2.2
• Initial Condition
  Preamble has been executed.

  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits a Pairing Request command with:
     a. IO capability set to “DisplayOnly” or “DisplayYesNo” or “KeyboardOnly” or “KeyboardDisplay”
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. AuthReq bonding flag set to ‘00’, and the MITM flag set to ‘0’
  2. Lower Tester responds with a Pairing Response command, with:
     a. IO capability set to “KeyboardOnly”
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. AuthReq bonding flag set to ‘00’, and the MITM flag set to ‘1’ and all reserved bits are set to ‘0’
  3. During the phase 2 pairing, the IUT displays the 6-digit passkey while the Lower Tester prompts user to enter the 6-digit passkey. If the IUT IO capabilities are “KeyboardOnly” the passkey is not displayed and both IUT and Lower Tester enter the same 6-digit passkey.
  4. IUT and Lower Tester use the same 6-digit passkey.
  5. IUT and Lower Tester perform phase 2 of the Passkey Entry pairing procedure and establish an encrypted link with the key generated in phase 2.

• Expected Outcome
  Pass Verdict

  The IUT can encrypt the link successfully.

4.4.2.2 SM/SLA/PKE/BV-02-C (Passkey Entry, IUT Responder – Success)

• Test Purpose
  Verify that the IUT performs the Passkey Entry pairing procedure correctly as responder.

• Reference
  [1] 2.3.5.3, 5.3.2.2

• Initial Condition
  Preamble has been executed.

  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Lower Tester initiates a Pairing Request command with:
     a. IO capability set to “KeyboardDisplay”
b. OOB data flag set to 0x00 (OOB Authentication data not present)

c. AuthReq bonding flag set to the value indicated in the IXIT [7] for ‘Bonding Flags’, and the MITM flag set to ‘1’ and all reserved bits are set to ‘0’

2. IUT responds with a Pairing Response command, with:
   a. IO capability set to “KeyboardOnly” or “KeyboardDisplay” or “DisplayYesNo” or “DisplayOnly”
   b. OOB data flag set to 0x00
   c. All reserved bits are set to ‘0’

3. During the phase 2 passkey pairing process, Lower Tester displays the 6-digit passkey while the IUT prompts user to enter the 6-digit passkey. If the IO capabilities of the IUT are “DisplayYesNo” or “DisplayOnly” the IUT displays the 6-digit passkey while the Lower Tester enters the 6-digit passkey.

4. IUT and Lower Tester use the same pre-defined 6-digit passkey.

5. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

• Expected Outcome

Pass Verdict

The master can encrypt the link successfully.

4.4.2.3 SM/MAS/PKE/BV-04-C [Passkey Entry, IUT Initiator – results in Unauthenticated Success]

• Test Purpose

Verify that the IUT performs the Passkey Entry pairing procedure correctly as initiator and pairing is successful if the Lower Tester only supports IO capabilities resulting in an Unauthenticated key.

• Reference

[1] 2.3.5.1, 5.3.2.1

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with:
   a. IO capability set to “DisplayOnly” or “DisplayYesNo” or “KeyboardOnly” or “KeyboardDisplay”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. All reserved bits are set to ‘0’

2. Lower Tester responds with a Pairing Response command, with:
   a. IO capability set to “NoInputNoOutput”
3. IUT and Lower Tester perform phase 2 of the Just Works pairing and establish an encrypted link with the generated STK.

- Expected Outcome

**Pass Verdict**

The IUT can encrypt the link successfully.

### 4.4.2.4 SM/SLA/PKE/BV-05-C [Passkey Entry, IUT Responder – Lower Tester has insufficient security for Passkey Entry]

- **Test Purpose**

Verify that the IUT that supports the Passkey Entry pairing procedure as responder correctly handles an initiator with insufficient security to result in an Authenticated key, yielding an unauthenticated key.

- **Reference**

[1] 2.3.5.1, 5.3.2.1

- **Initial Condition**

Preamble has been executed.

IUT is slave. Lower Tester is master.

- **Test Procedure**

1. Lower Tester initiates a Pairing Request command with:
   a. IO capability set to “NoInputNoOutput”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq bonding flag set to ‘00’, and the MITM flag set to ‘0’ and all reserved bits are set to ‘0’

2. IUT responds with a Pairing Response command, with:
   a. IO capability set to “KeyboardOnly” or “KeyboardDisplay” or “DisplayYesNo” or “DisplayOnly”
   b. OOB data flag set to 0x00 and the MITM flag set to ‘1’ and all reserved bits are set to ‘0’
   c. Alternatively, the IUT may respond with Pairing Failed command with reason code set to ‘Authentication Requirements’.

3. IUT and Lower Tester perform phase 2 of the Just Works pairing and establish an encrypted link with the generated STK.

- **Expected Outcome**

**Pass Verdict**

The master can encrypt the link successfully
ALT: IUT responds with Pairing Failed command with reason code set to ‘Authentication Requirements’.

4.4.2.5 SM/MAS/PKE/BI-01-C [Passkey Entry, IUT Initiator – Failure on Responder Side]

- Test Purpose
  Verify that the IUT handles the invalid Passkey Entry pairing procedure correctly as initiator.

- Reference
  [1] 2.3.5.3, 5.3.2.2

- Initial Condition
  Preamble has been executed.
  IUT is master. Lower Tester is slave.

- Test Procedure
  1. IUT transmits a Pairing Request command with:
     a. IO capability set to “DisplayOnly” or “DisplayYesNo” or “KeyboardOnly” or “KeyboardDisplay”
     b. OOB data flag set to 0x00 and all the reserved bits are set to ‘0’
  2. Lower Tester responds with a Pairing Response command, with:
     a. IO capability set to “KeyboardOnly”
     b. OOB data flag set to 0x00 and MITM bit set to ‘1’
  3. During the phase 2 pairing, IUT displays the 6-digit passkey while the Lower Tester enters a different 6-digit passkey. If the IUT IO capabilities are “KeyboardOnly” then both IUT and Lower Tester enter different passkeys.
  4. IUT and Lower Tester perform phase 2 of the LE pairing.
  5. Lower Tester transmits ‘Pairing Random’ (SrAnd) command even though the passkey entry was incorrect.
  6. IUT responds with ‘Pairing Failed’ command.

- Expected Outcome
  Pass Verdict

  IUT detects that the ‘Pairing Random’ value from the Lower Tester is incorrect and sends ‘Pairing Failed’ command to the Lower Tester.

4.4.2.6 SM/MAS/PKE/BI-02-C [Passkey Entry, IUT Initiator – Interrupted passkey entry by Responder Side]

- Test Purpose
  Verify that the IUT handles the interrupted passkey entry by the responder.

- Reference
  [1] 2.3.5.3, 5.3.2.2
• Initial Condition
  Preamble has been executed.
  
  IUT is master. Lower Tester is slave.

• Test Procedure
  
  1. IUT transmits a Pairing Request command with:

     a. IO capability set to “DisplayOnly” or “DisplayYesNo” or “KeyboardOnly” or “KeyboardDisplay”
     b. OOB data flag set to 0x00 and all reserved bits are set to '0'

  2. Lower Tester responds with a Pairing Response command, with:

     a. IO capability set to “KeyboardOnly”
     b. OOB data flag set to 0x00 and MITM bit set to '1' and all the reserved bits are set to '0'

  3. During the phase 2 pairing, if IO capability is set to “DisplayOnly”, “DisplayYesNo” or “KeyboardDisplay” the IUT displays the 6-digit passkey. If the IUT IO capabilities are “KeyboardOnly” the passkey is not displayed and both IUT and Lower Tester enter the same 6-digit passkey.

  4. Emulating interrupted passkey entry the Lower Tester issues a Pairing Failed command with reason code set to ‘0x01’ (Passkey Entry Failed).

  5. The pairing process is aborted. IUT reports the failure to the Upper Tester.

• Expected Outcome
  
  Pass Verdict
  
  IUT detects the Pairing Failed from the Lower Tester and reports the failure to the Upper Tester.

4.4.2.7 SM/SLA/PKE/BI-03-C [Passkey Entry, IUT Responder – Failure on Initiator Side]

• Test Purpose
  
  Verify that the IUT handles the invalid passkey entry pairing procedure correctly as responder.

• Reference
  
  [1] 2.3.5.3, 5.3.2.2

• Initial Condition
  
  Preamble has been executed.
  
  IUT is slave. Lower Tester is master.

• Test Procedure

  1. Lower Tester initiates a Pairing Request command with:

     a. IO capability set to “KeyboardOnly”
     b. OOB data flag set to 0x00 and MITM bit set to ‘1’ and all the reserved bits are set to '0'
2. IUT responds with a Pairing Response command, with:
   a. IO capability set to “DisplayOnly” or “DisplayYesNo” or “KeyboardDisplay” or “KeyboardOnly”
   b. OOB data flag set to 0x00 and all the reserved bits are set to ‘0’

3. IUT and Lower Tester use different 6-digit passkey.

4. During the phase 2 pairing, IUT displays 6-digit passkey while the Lower Tester enters different 6-digit passkey. If IUT IO capabilities are “KeyboardOnly” the passkey is not displayed and IUT and Lower Tester enter different 6-digit passkeys.

5. IUT and Lower Tester perform phase 2 of the LE pairing.

   • Expected Outcome
     Pass Verdict
     IUT detects the ‘Pairing Confirm’ value from the Lower Tester is incorrect and sends ‘Pairing Failed’ command to the Lower Tester.

4.4.3 Out of Band (OOB)

4.4.3.1 SM/MAS/OOB/BV-01-C [IUT Initiator – Both sides have OOB data – Success]

   • Test Purpose
     Verify that the IUT performs the OOB pairing procedure correctly as initiator.

   • Reference
     [1] 2.3.5.4, 5.3.2.3

   • Initial Condition
     Preamble has been executed.

     IUT is master. Lower Tester is slave.

   • Test Procedure
     1. IUT transmits a Pairing Request command with OOB data flag set to 0x01.
     2. Lower Tester responds with a Pairing Response command with OOB data flag set to 0x01.
     3. IUT and Lower Tester use the same 128-bit value as OOB data.
     4. IUT and Lower Tester perform phase 2 of the pairing process and establish an encrypted link with the key generated in phase 2.

   • Expected Outcome
     Pass Verdict
     The IUT can encrypt the link successfully.

   • Notes
     OOB data are exchanged out of band.
4.4.3.2 SM/SLA/OOB/BV-02-C [IUT Responder – Both sides have OOB data – Success]

- Test Purpose
  Verify that the IUT performs the OOB pairing procedure correctly as responder.

- Reference
  [1] 2.3.5.3, 5.3.2.2

- Initial Condition
  Preamble has been executed.
  IUT is slave. Lower Tester is master.

- Test Procedure
  1. Lower Tester initiates a Pairing Request command with OOB data flag set to 0x01.
  2. IUT responds with a Pairing Response command with OOB data flag set to 0x01.
  3. IUT and Lower Tester use the same 128 bit value as OOB data.
  4. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

- Test Condition
  IUT and Lower Tester use the same OOB data values in this test case.

- Expected Outcome
  Pass Verdict
  The master can encrypt the link successfully.

4.4.3.3 SM/MAS/OOB/BV-03-C [IUT Initiator – Only IUT has OOB data – Success]

- Test Purpose
  Verify that the IUT performs pairing correctly as initiator if the responder does not have OOB data.

- Reference
  [1] 2.3.5.3, 5.3.2.2

- Initial Condition
  Preamble has been executed.
  IUT is master. Lower Tester is slave.

- Test Procedure
  1. IUT transmits a Pairing Request command with:
     a. IO capability set to “DisplayOnly” or “DisplayYesNo” or “KeyboardOnly” or “KeyboardDisplay”
     b. OOB data flag set to 0x01
2. Lower Tester responds with a Pairing Response command, with:
   a. IO capability set to “KeyboardOnly”
   b. OOB data flag set to 0x00 and MITM bit set to ‘1’

3. IUT generates a random 6-digit passkey between 000,000 and 999,999.
4. During the phase 2 pairing, IUT displays the 6-digit passkey while the Lower Tester enters the same 6-digit passkey. If the IUT IO capabilities are “KeyboardOnly” the passkey is not displayed and both IUT and Lower Tester enter the same 6-digit passkey.
5. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

- **Expected Outcome**
  
  **Pass Verdict**

  The IUT can encrypt the link successfully.

4.4.3.4 SM/SLA/OOB/BV-04-C [IUT Responder – Only IUT has OOB data – Success]

- **Test Purpose**
  
  Verify that the IUT performs the pairing procedure correctly as responder if only the IUT has OOB data.

- **Reference**

  [1] 2.3.5.3, 5.3.2.2

- **Initial Condition**

  Preamble has been executed.

  IUT is slave. Lower Tester is master.

- **Test Procedure**

  1. Lower Tester initiates a Pairing Request command with:
     a. IO capability set to “KeyboardDisplay”
     b. OOB data flag set to 0x00 and MITM bit set to ‘1’

  2. IUT responds with a Pairing Response command, with:
     a. IO capability set to “KeyboardOnly” or “KeyboardDisplay” or “DisplayOnly” or “DisplayYesNo”
     b. OOB data flag set to 0x01 and MITM bit set to ‘1’

  3. Lower Tester has a pre-defined 6-digit passkey.

  4. During the phase 2 pairing, the Lower Tester displays the 6-digit passkey while the user of IUT enters the same 6-digit passkey. If the IO capabilities of the IUT are “DisplayYesNo” or “DisplayOnly” the IUT displays the 6-digit passkey while the Lower Tester enters the 6-digit passkey.

  5. IUT and the Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
• Expected Outcome

Pass Verdict

The master can encrypt the link successfully.

4.4.3.5 SM/MAS/OOB/BV-05-C [IUT Initiator – Only Lower Tester has OOB data – Success]

• Test Purpose

Verify that the IUT performs the OOB pairing procedure correctly as initiator if only the Lower Tester has OOB data.

• Reference

[1] 2.3.5.3, 5.3.2.2

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with:
   a. IO capability set to “DisplayOnly” or “DisplayYesNo”, or “KeyboardOnly” or “KeyboardDisplay”
   b. OOB data flag set to 0x00

2. Lower Tester responds with a Pairing Response command, with:
   a. IO capability set to “KeyboardOnly”
   b. OOB data flag set to 0x01 and MITM bit set to ‘1’

3. IUT generates a random pre-defined 6-digit passkey between 000,000 and 999,999 and begins phase 2 pairing.

4. During the phase 2 pairing, IUT displays the 6-digit passkey while the Lower Tester enters the same 6-digit passkey. If IUT has IO capabilities set to “KeyboardOnly” the passkey is not displayed and both initiator and responder input the same 6-digit passkey.

5. IUT and the Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

• Expected Outcome

Pass Verdict

The IUT can encrypt the link successfully.
4.4.3.6 SM/SLA/OOB/BV-06-C [IUT Responder – Only Lower Tester has OOB data – Success]

- **Test Purpose**
  Verify that the IUT performs the pairing procedure correctly as responder if only the Lower Tester has OOB data.

- **Reference**
  [1] 2.3.5.3, 5.3.2.2

- **Initial Condition**
  Preamble has been executed.

  IUT is slave. Lower Tester is master.

- **Test Procedure**
  1. Lower Tester initiates a Pairing Request command with:
     a. IO capability set to “KeyboardDisplay”.
     b. AuthReq bonding flag set to the value indicated in the IXIT [7] for ‘Bonding Flags’, and the MITM flag set to ‘1’ and all reserved bits are set to ‘0’.
     c. OOB data flag set to 0x01.
  2. IUT responds with a Pairing Response command, with:
     a. IO capability set to “KeyboardOnly” or “KeyboardDisplay” or “DisplayOnly” or “DisplayYesNo”
     b. OOB data flag set to 0x00
  3. Alternatively, the IUT may respond with Pairing Failed command with reason code set to ‘OOB Not Available’ or ‘Authentication Requirements’.
  4. Lower Tester has a pre-defined 6-digit passkey.
  5. During the phase 2 pairing, the Lower Tester displays the 6-digit passkey while the user of the IUT enters the same 6-digit passkey.
  6. IUT and the Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2. If the IO capabilities of the IUT are “DisplayYesNo” or “DisplayOnly” the IUT displays the 6-digit passkey while the Lower Tester enters the 6-digit passkey.

- **Expected Outcome**
  **Pass Verdict**

  The master can encrypt the link successfully.

  **ALT:** IUT responds with Pairing Failed, with reason code set to ‘OOB Not Available’ or ‘Authentication Requirements’. 
4.4.3.7 SM/MAS/OOB/BV-07-C (IUT Initiator – Only Lower Tester has OOB data – Unauthenticated Success)

- Test Purpose
  Verify that the IUT performs the OOB pairing procedure correctly as initiator if only the Lower Tester has OOB data and the IUT does not require MITM protection.

- Reference
  [1] 2.3.5.1, 5.3.2.1

- Initial Condition
  Preamble has been executed.
  IUT is master. Lower Tester is slave.

- Test Procedure
  1. IUT transmits a Pairing Request command with:
     a. IO capability set to any IO capability
     b. OOB data flag set to 0x00
  2. Lower Tester responds with a Pairing Response command, with:
     a. IO capability set to “NoInputNoOutput”
     b. OOB data flag set to 0x01 and MITM bit set to ‘0’
  3. IUT and Lower Tester perform phase 2 of the just works pairing procedure and establish an encrypted link with the key generated in phase 2.

- Expected Outcome
  Pass Verdict
  The IUT can encrypt the link successfully.

4.4.3.8 SM/SLA/OOB/BV-08-C [IUT Responder – Only Lower Tester has OOB data – Lower Tester also supports Just Works]

- Test Purpose
  Verify that the IUT performs the pairing procedure correctly as responder if only the Lower Tester has OOB data and supports the Just Works pairing method.

- Reference
  [1] 2.3.5.1, 5.3.2.1

- Initial Condition
  Preamble has been executed.
  IUT is slave. Lower Tester is master.
• Test Procedure
  1. Lower Tester initiates a Pairing Request command with:
     a. IO capability set to "NoInputNoOutput"
     b. OOB data flag set to 0x01 and MITM bit set to ‘0’
  2. IUT responds with a Pairing Response command, with:
     a. IO capability set to any IO capability
     b. OOB data flag set to 0x00
  3. Alternatively, IUT may respond with Pairing Failed command with reason code set to ‘OOB Not Available’ or ‘Authentication Requirements’.
  4. IUT and Lower Tester perform phase 2 of the just works pairing and establish an encrypted link with the generated STK.

• Expected Outcome
  Pass Verdict
  The master can encrypt the link successfully.

  ALT: IUT responds with Pairing Failed with reason code set to ‘OOB Not Available’ or ‘Authentication Requirements’.

4.4.3.9 SM/MAS/OOB/BV-09-C [IUT Initiator – Only IUT has OOB data – Unauthenticated Success]

• Test Purpose
  Verify that the IUT performs pairing correctly as initiator if the responder does not have OOB data and the IUT does not require MITM protection.

• Reference
  [1] 2.3.5.1, 5.3.2.1

• Initial Condition
  Preamble has been executed.
  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits a Pairing Request command with:
     a. IO capability set to any IO capability
     b. OOB data flag set to 0x01
  2. Lower Tester responds with a Pairing Response command, with:
     a. IO capability set to “NoInputNoOutput”
     b. OOB data flag set to 0x00 and MITM bit set to ‘0’
  3. IUT and Lower Tester perform phase 2 of the Just Works pairing procedure and establish an encrypted link with the key generated in phase 2.
• Expected Outcome
   Pass Verdict
   The IUT can encrypt the link successfully.

4.4.3.10 SM/SLA/OOB/BV-10-C [IUT Responder – Only IUT has OOB data – Lower Tester also supports Just Works]

• Test Purpose
   Verify that the IUT performs the pairing procedure correctly as responder if only the IUT has OOB data and the Lower Tester supports the Just Works pairing method.

• Reference
   [1] 2.3.5.1, 5.3.2.1

• Initial Condition
   Preamble has been executed.
   IUT is slave. Lower Tester is master.

• Test Procedure
   1. Lower Tester initiates a Pairing Request command with:
      a. IO capability set to "NoInputNoOutput"
      b. OOB data flag set to 0x00 and MITM bit set to ‘0’
   2. IUT responds with a Pairing Response command, with:
      a. IO capability set to any IO capability
      b. OOB data flag set to 0x01
   3. Alternatively, IUT may respond with Pairing Failed command with reason code set to ‘Authentication Requirements’.
   4. IUT and Lower Tester perform phase 2 of the just works pairing and establish an encrypted link with the generated STK.

• Expected Outcome
   Pass Verdict
   The master encrypts the link successfully or in the alternate case the IUT responds with the Pairing Failed commend with the reason code set to ‘Authentication Requirements’.

4.4.3.11 SM/MAS/OOB/BI-01-C [IUT Initiator – Both sides have different OOB data – Failure]

• Test Purpose
   Verify that the IUT initiates OOB pairing procedure and handles the failure correctly.

• Reference
   [1] 2.3.5.3, 5.3.2.2
• Initial Condition
Preamble has been executed.

IUT and Lower Tester have different 128 bit OOB data.

IUT is master. Lower Tester is slave.

• Test Procedure
1. IUT transmits Pairing Request command with OOB data flag set to 0x01 and its MITM bit set to ‘1’.
2. Lower Tester responds with a Pairing Response command, with OOB data flag to set 0x01 and MITM bit set to ‘1’.
3. IUT detects the mismatch of confirm value. IUT sends Pairing Failed and the Lower Tester initiates disconnect.

• Expected Outcome
Pass Verdict

IUT detects the mismatch of confirm value, sends ‘Pairing Failed’ and the Lower Tester disconnects the link.

4.4.3.12 SM/SLA/OOB/B1-02-C [IUT Responder – Both sides have different OOB data – Failure]

• Test Purpose
Verify that the IUT responds to OOB pairing procedure and handles the failure correctly.

• Reference
[1] 2.3.5.3, 5.3.2.2

• Initial Condition
Preamble has been executed.

IUT is slave. Lower Tester is master.

IUT and Lower Tester have different 128 bit OOB data.

IUT OOB data can be anything but the same value as the OOB data in the Lower Tester.

• Test Procedure
1. Lower Tester initiates Pairing Request command with OOB data flag set to 0x01 and its MITM bit set to ‘1’.
2. IUT responds with Pairing Response command with OOB data flag set to 0x01 and MITM bit set to ‘1’.
3. IUT detects the mismatch of confirm value, sends Pairing Failed and notifies the Upper Tester.
• Expected Outcome
  Pass Verdict
  
  IUT detects the mismatch of confirm value and notifies the Upper Tester.

4.5 Encryption Key Size

The test group objective is to verify the correct implementation of the encryption key size negotiation procedure.

4.5.1 Encryption Key Size Negotiation

4.5.1.1 SM/MAS/EKS/BV-01-C [IUT initiator, Lower Tester Maximum Encryption Key Size = Min_Encryption_Key_Length]

• Test Purpose
  Verify that the IUT uses correct key size during encryption as initiator.

• Reference
  [1] 2.3.4

• Initial Condition
  Preamble has been executed.

  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits pairing request.
  2. Lower Tester responds with Pairing Response command with Maximum Encryption Key Size field set to Min_Encryption_Key_Length’.
  3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

• Expected Outcome
  Pass Verdict
  
  The IUT can encrypt the link successfully.

• Notes
  The value of Min_Encryption_Key_Length is specified in the IXIT [7].

4.5.1.2 SM/SLA/EKS/BV-02-C [IUT Responder, Lower Tester Maximum Encryption Key Size = Min_Encryption_Key_Length]

• Test Purpose
  Verify that the IUT uses correct key size during encryption as responder.

• Reference
  [1] 2.3.4
• Initial Condition

Preamble has been executed.

IUT is slave. Lower Tester is master.

• Test Procedure

1. Lower Tester initiates Pairing Request command with Maximum Encryption Key Size field set to Min_Encryption_Key_Length’.
2. IUT responds with Pairing Response command.
3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

• Expected Outcome

Pass Verdict

The Lower Tester can encrypt the link successfully.

• Notes

The value of Min_Encryption_Key_Length is specified in the IXIT [7].

4.5.1.3 SM/MAS/EKS/BI-01-C [IUT initiator, Lower Tester Maximum Encryption Key Size < Min_Encryption_Key_Length]

• Test Purpose

Verify that the IUT checks that the resultant encryption key size is not smaller than the minimum key size.

• Reference

[1] 2.3.4

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command.
2. Lower Tester responds with a Pairing Response command with Maximum Encryption Key Size field set to Min_Encryption_Key_Length -1. The value of Min_Encryption_Key_Length used should be determined by the value supported on the IUT and given by IXIT [7] value.
3. IUT transmits the Pairing Failed command.

• Expected Outcome

Pass Verdict

- IUT transmits Pairing Failed command.
- If the IUT supports a value of Min_Encryption_Key_Length greater than the minimum defined value for the encryption key length parameter in the specification, the IUT transmits the Pairing Failed comment with error code “Encryption Key Size”.

- If the IUT supports only the minimum defined values for the encryption key length parameter in the specification, the IUT transmits the Pairing Failed command and may respond with error code “Invalid Parameters”.

4.5.1.4 SM/SLA/EKS/BI-02-C [IUT Responder, Lower Tester Maximum Encryption Key Size < Min_Encryption_Key_Length]

- Test Purpose
  Verify that the IUT uses correct key size during encryption as responder.

- Reference
  [1] 2.3, 2.3.4

- Initial Condition
  Preamble has been executed.

  IUT is slave. Lower Tester is master.

- Test Procedure
  1. Lower Tester initiates Pairing Request command with Maximum Encryption Key Size field set to Min_Encryption_Key_Length - 1.
  2. IUT transmits the Pairing Failed command.

- Expected Outcome
  Pass Verdict

  IUT detects that encryption key size is smaller than the minimum key size parameter for the IUT and responds with Pairing Failed command.

  If the IUT supports a value of Maximum Encryption Key Size greater than the minimum defined value for the encryption key length parameter in the Specification the IUT transmits the Pairing Failed command with error code “Encryption Key Size”.

  If the IUT supports only the minimum defined value for the encryption key length parameter the IUT transmits the Pairing Failed command and may respond with error code “Invalid Parameters”.

4.6 Signing

The test group objective is to verify the correct implementation of the generation and verification of MAC with signed data.

4.6.1 Signing of Data

4.6.1.1 SM/MAS/SIGN/BV-01-C [IUT transfers signed data – Success]

- Test Purpose
  Verify that the IUT has implemented the signing algorithm correctly for data transferring.
• Reference
  [1] 2.4.5

• Initial Condition
Preamble has been executed.
Pairing has been executed and IUT has distributed CSRK as requested by the Lower Tester.
A new link has been established with no encryption.
SignCounter is set to 0.

• Test Procedure
IUT transfers a pre-defined packet with signed MAC and SignCounter.

• Expected Outcome
Pass Verdict
IUT has correct MAC in the signed data.

4.6.1.2 SM/MAS/SIGN/BV-03-C [IUT receives signed data – Success]

• Test Purpose
Verify that the IUT has implemented the signing algorithm correctly for data receiving.

• Reference
  [1] 2.4.5

• Initial Condition
Preamble has been executed.
Pairing has been executed and Lower Tester has distributed CSRK as requested by the IUT.
A new link has been established with no encryption.
SignCounter is set to 0.

• Test Procedure
Lower Tester transfers a pre-defined packet with signed MAC and SignCounter.
IUT has verified the MAC with signed data correctly.

• Expected Outcome
Pass Verdict
IUT has verified the MAC with signed data correctly.
IUT has forwarded the signed data to the Upper Tester correctly.
4.6.1.3 SM/MAS/SIGN/BI-01-C [IUT receives signed data – Failure]

• Test Purpose
Verify that the IUT has implemented the signing algorithm correctly to detect a failure in signed data.

• Reference
[1] 2.4.5

• Initial Condition
Preamble has been executed.

Pairing has been executed and Lower Tester has distributed CSRK as requested by the IUT.

A new link has been established with no encryption.

• Test Procedure
Lower Tester transfers a pre-defined packet with incorrectly signed MAC.

IUT has detected the incorrectly signed MAC and ignores the received PDU.

• Expected Outcome
Pass Verdict

IUT has detected the incorrectly signed MAC and ignores the received PDU.

Upper Tester may be notified.

4.7 Key Distribution and Usage
The test group objective is to verify the correct implementation of key distribution and usage.

4.7.1 Key Distribution during bonding

4.7.1.1 SM/SLA/KDU/BV-01-C [LE Legacy Pairing, IUT Responder – Lower Tester sets EncKey bit – Success]

• Test Purpose
Verify that the IUT sends LTK, EDIV and Rand after a successful pairing.

• Reference
[1] 3.6.1

• Initial Condition
Preamble has been executed.

IUT is slave. Lower Tester is master.
• Test Procedure

1. Lower Tester initiates a Pairing Request command, with the SC bit of AuthReq set to ‘0’ and with the EncKey bit of ‘responder key distribution’ set to ‘1’, and the IdKey and SignKey bits set to ‘0’.
2. IUT responds with a Pairing Response command with the EncKey bit of ‘responder key distribution’ set to ‘1’, and the IdKey and SignKey bits set to ‘0’.
3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
4. IUT distributes the LTK using the Encryption Information command followed by EDIV and RAND using the Master Identification command.

• Expected Outcome

Pass Verdict

The IUT sets the EncKey bit of ‘responder key distribution’ to 1, and sets the IdKey and SignKey bits to ‘0’ in the Pairing Response command.

IUT distributes LTK using the Encryption Information command followed by EDIV and Rand using the Master Identification command.

IUT does not distribute any other key information to the Lower Tester.

4.7.1.2 SM/SLA/KDU/BV-02-C [LE Legacy Pairing, IUT Responder – Lower Tester sets IdKey bit – Success]

• Test Purpose

Verify that the IUT sends IRK after a successful pairing.

• Reference

[1] 3.6.1

• Initial Condition

Preamble has been executed.

IUT is slave. Lower Tester is master.

• Test Procedure

1. Lower Tester initiates a Pairing Request command, with the SC bit of AuthReq set to ‘0’ and with the IdKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, SignKey bits set to ‘0’.
2. IUT responds with a Pairing Response command with IdKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, SignKey bits set to ‘0’.
3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
4. IUT distributes the IRK using the Identity information and Identity Address Information command correctly with proper order. IUT does not distribute any other keys.
• Expected Outcome

Pass Verdict

The IUT distributes IRK using the Identity Information command and Identity Address Information command correctly with proper order. IUT does not distribute any other keys.

4.7.1.3 SM/SLA/KDU/BV-03-C [LE Legacy Pairing, IUT Responder – Lower Tester sets SignKey bit – Success]

• Test Purpose

Verify that the IUT sends CSRK after a successful pairing.

• Reference

[1] 3.6.1

• Initial Condition

Preamble has been executed.

IUT is slave. Lower Tester is master.

• Test Procedure

1. Lower Tester initiates a Pairing Request command, with the SC bit of AuthReq set to ‘0’ and with the SignKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, IdKey bits set to ‘0’.
2. IUT responds with a Pairing Response command with the SignKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, IdKey bits set to ‘0’.
3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
4. IUT distributes CSRK using the Signing Information command and does not distribute any other keys.

• Expected Outcome

Pass Verdict

The IUT distributes CSRK using the Signing Information command and does not distribute any other keys.

4.7.1.4 SM/MAS/KDU/BV-04-C [LE Legacy Pairing, IUT Initiator – Lower Tester sets SignKey bit – Success]

• Test Purpose

Verify that the IUT sends CSRK after a successful pairing.

• Reference

[1] 3.6.1
• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with SignKey bit of Initiator Key Distribution set to ‘1’. IUT may set EncKey, IdKey of Initiator Key Distribution field, and the bits in Responder Key Distribution field to either ‘1’ or ‘0’.

2. Lower Tester responds with a Pairing Response command with the SC bit of AuthReq set to ‘0’ and with the SignKey bit of Initiator Key Distribution set to ‘1’ and EncKey, IdKey bits set to ‘0’. The EncKey, IdKey and SignKey bits of the Responder Key Distribution are all set to ‘0’.

3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

4. IUT distributes the CSRK using the Signing Information command. IUT does not distribute any other keys to the Lower Tester.

• Expected Outcome

Pass Verdict

The IUT distributes CSRK using the Signing Information command. IUT does not distribute any other keys to the Lower Tester.

4.7.1.5 SM/MAS/KDU/BV-05-C [LE Legacy Pairing, IUT Initiator – Lower Tester sets Idkey bit – Success]

• Test Purpose

Verify that the IUT sends IRK using correct procedures.

• Reference

[1] 3.6.1

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with the IdKey bit of “Initiator Key Distribution” field set to ‘1’. IUT may set the EncKey and SignKey bits of “Initiator Key Distribution” and the bits in “Responder Key Distribution” field to either ‘1’ or ‘0’.

2. Lower Tester responds with Pairing Response command with the SC bit of AuthReq set to ‘0’ and with the IdKey bit of “Initiator Key Distribution” field set to ‘1’, the EncKey and SignKey bits of “Initiator Key Distribution” are set to ‘0’. Lower Tester also sets all bits in “Responder Key Distribution” field to ‘0’.

3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
4. IUT distributes IRK using Identity Information command followed by its public or random address using Identity Address Information command. IUT does not distribute any other keys to the Lower Tester.

- Expected Outcome
  - Pass Verdict

  The IUT distributes IRK using the Identity Information command followed by its public device or static random address using Identity Address Information.

  IUT does not distribute any other keys to the Lower Tester.

4.7.1.6 SM/MAS/KDU/BV-06-C [LE Legacy Pairing, IUT Initiator – Lower Tester sets EncKey bit – Success]

- Test Purpose
  - Verify that the IUT sends LTK and EDIV using correct procedures.

- Reference
  [1] 3.6.1

- Initial Condition
  - Preamble has been executed.

  IUT is master. Lower Tester is slave.

- Test Procedure
  1. IUT transmits a Pairing Request command with EncKey bit of “Initiator Key Distribution” field set to ‘1’. IUT may set the IdKey and SignKey bits of “Initiator Key Distribution” field and the bits of “Responder Key Distribution” field to ‘1’ or ‘0’.
  2. Lower Tester responds with a Pairing Response command with the SC bit of AuthReq set to ‘0’ and with the EncKey bit of “Initiator Key Distribution” field set to ‘1’, the IdKey and SignKey bits set to ‘0’. Lower Tester also sets all bits in “Responder Key Distribution” field to ‘0’.
  3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
  4. IUT distributes LTK using the Encryption Information command followed by EDIV and RAND using the Master Identification command. IUT does not distribute any other key information to the Lower Tester.

- Expected Outcome
  - Pass Verdict

  The IUT distributes LTK using the Encryption Information command followed by EDIV and Rand using the Master Identification command.

  IUT does not distribute any other key information to the Lower Tester.
4.7.1.7 SM/SLA/KDU/BV-08-C [LE Secure Connections Pairing, IUT Responder – Lower Tester sets IdKey bit – Success]

- Test Purpose
  Verify that the IUT sends IRK after a successful pairing using LE Secure Connections.

- Reference
  [8] 3.6.1

- Initial Condition
  Preamble has been executed.
  IUT is slave. Lower Tester is master.

- Test Procedure
  1. Lower Tester initiates a Pairing Request command, with the SC bit of AuthReq set to ‘1’ and with the IdKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, SignKey bits set to ‘0’.
  2. IUT responds with a Pairing Response command with the SC bit of AuthReq set to ‘1’ and with IdKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, SignKey bits set to ‘0’.
  3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
  4. IUT distributes the IRK using the Identity information and Identity Address Information command correctly with proper order. IUT does not distribute any other keys.

- Expected Outcome
  Pass Verdict
  The IUT distributes IRK using the Identity Information command and Identity Address Information command correctly with proper order. IUT does not distribute any other keys.

4.7.1.8 SM/SLA/KDU/BV-09-C [LE Secure Connections Pairing, IUT Responder – Lower Tester sets SignKey bit – Success]

- Test Purpose
  Verify that the IUT sends CSRK after a successful pairing using LE Secure Connections.

- Reference
  [8] 3.6.1

- Initial Condition
  Preamble has been executed.
  IUT is slave. Lower Tester is master.

- Test Procedure
  1. Lower Tester initiates a Pairing Request command, with the SC bit of AuthReq set to ‘1’ and with the SignKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, IdKey bits set to ‘0’.
2. IUT responds with a Pairing Response command with the SC bit of AuthReq set to ‘1’ and with the SignKey bit of ‘responder key distribution’ set to ‘1’ and EncKey, IdKey bits set to ‘0’.

3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

4. IUT distributes CSRK using the Signing Information command and does not distribute any other keys.

• Expected Outcome

Pass Verdict

The IUT distributes CSRK using the Signing Information command and does not distribute any other keys.

4.7.1.9 SM/MAS/KDU/BV-10-C [LE Secure Connections Pairing, IUT Initiator – Lower Tester sets IdKey bit – Success]

• Test Purpose

Verify that the IUT sends IRK after a successful pairing using LE Secure Connections.

• Reference

[8] 3.6.1

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with the SC bit of AuthReq set to ‘1’ and with the IdKey bit of “Initiator Key Distribution” field set to ‘1’. IUT may set the EncKey and SignKey bits of “Initiator Key Distribution” and the bits in “Responder Key Distribution” field to either ‘1’ or ‘0’.

2. Lower Tester responds with Pairing Response command with the SC bit of AuthReq set to ‘1’ and with the IdKey bit of “Initiator Key Distribution” field set to ‘1’, the EncKey and SignKey bits of “Initiator Key Distribution” are set to ‘0’. Lower Tester also sets all bits in “Responder Key Distribution” field to ‘0’.

3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.

4. IUT distributes IRK using Identity Information command followed by its public or random address using Identity Address Information command. IUT does not distribute any other keys to the Lower Tester.

• Expected Outcome

Pass Verdict

The IUT distributes IRK using the Identity Information command followed by its public device or static random address using Identity Address Information.

IUT does not distribute any other keys to the Lower Tester.
4.7.1.10 SM/MAS/KDU/BV-11-C [LE Secure Connections Pairing, IUT Initiator – Lower Tester sets SignKey bit – Success]

- **Test Purpose**
  Verify that the IUT sends CSRK after a successful pairing using LE Secure Connections.

- **Reference**
  [8] 3.6.1

- **Initial Condition**
  Preamble has been executed.
  IUT is master. Lower Tester is slave.

- **Test Procedure**
  1. IUT transmits a Pairing Request command with the SC bit of AuthReq set to ‘1’ and with SignKey bit of Initiator Key Distribution set to ‘1’. IUT may set EncKey, IdKey of Initiator Key Distribution field, and the bits in Responder Key Distribution field to either ‘1’ or ‘0’.
  2. Lower Tester responds with a Pairing Response command with the SC bit of AuthReq set to ‘1’ and with the SignKey bit of Initiator Key Distribution set to ‘1’ and EncKey, IdKey bits set to ‘0’. The EncKey, IdKey and SignKey bits of the Responder Key Distribution are all set to ‘0’.
  3. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the key generated in phase 2.
  4. IUT distributes the CSRK using the Signing Information command. IUT does not distribute any other keys to the Lower Tester.

- **Expected Outcome**
  **Pass Verdict**
  The IUT distributes CSRK using the Signing Information command. IUT does not distribute any other keys to the Lower Tester.

4.7.1.11 SM/SLA/KDU/BI-01-C [LE Secure Connections Pairing, IUT Responder – Lower Tester sends invalid public key]

- **Test Purpose**
  Verify that the IUT detects an invalid public key from the LT.

- **Reference**
  [8], [9] 2.3.5.6.1

- **Initial Condition**
  Preamble has been executed.
  IUT is slave. Lower Tester is master.

  FKC is the number of failed pairing attempts before the Upper Tester generates a new key pair as defined in the IXIT [7] entry and is used in Table 4.2.
The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.

- Test Procedure

![Test Procedure Diagram]

Figure 4.1: SM/SLA/KDU/BI-01-C [LE Secure Connections Pairing, IUT Responder – Lower Tester sends invalid public key]

Execute Steps 1–5 for each round in Table 4.2, repeating the number of times as specified in Table 4.2.

1. The Upper Tester sends an HCI_Reset command to the IUT.
2. The Lower Tester initiates a Pairing Request command, with the SC bit of AuthReq set to ‘1’.
3. The IUT responds with a Pairing Response command with the SC bit of AuthReq set to ‘1’.
4. The IUT and Lower Tester perform the Public Key Exchange. The Lower Tester generates a new valid private/public key pair and modifies the keys as specified in Table 4.2. The Lower Tester verifies that these new coordinates are not on the curve before sending them; if accidentally the new coordinates are valid, then the generation procedure is repeated. The resulting invalid Public Key is sent over the air.
5. The Lower Tester continues the pairing procedure using the public key value sent over the air until the IUT fails the pairing procedure. In Authentication Stage 2, the Lower Tester either uses the computed DHKey or DHKey = 0 as specified in Table 4.2.

<table>
<thead>
<tr>
<th>Round</th>
<th>Key Size</th>
<th>Invalid Key Type</th>
<th>Repeat # of times</th>
<th>Lower Tester DHKey</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P-256</td>
<td>Generate valid public key and set y-coordinate = 0</td>
<td>If FKC = 0, then run 1 time, otherwise Max(20*FKC, 20)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>P-256</td>
<td>Generate valid public key and set y-coordinate = 0</td>
<td>1</td>
<td>Computed DHKey</td>
</tr>
<tr>
<td>3</td>
<td>P-256</td>
<td>Generate valid public key and flip a bit in y-coordinate</td>
<td>1</td>
<td>Computed DHKey</td>
</tr>
<tr>
<td>Round</td>
<td>Key Size</td>
<td>Invalid Key Type</td>
<td>Repeat # of times</td>
<td>Lower Tester DHKey</td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>--------------------------------------</td>
<td>------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>4</td>
<td>P-256</td>
<td>Public Key coordinates (0, 0)</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 4.2: Invalid Public Key generation for each round*

- **Expected Outcome**

  **Pass Verdict**

  The IUT fails the pairing procedure any time after receiving the invalid public key. If the IUT sends a Pairing Failed message then any reason code is allowed.

  **Fail Verdict**

  The IUT successfully completes the pairing procedure.

4.7.1.12 SM/MAS/KDU/BI-01-C [LE Secure Connections Pairing, IUT Initiator – Lower Tester sends invalid public key]

- **Test Purpose**

  Verify that the IUT detects an invalid public key from the LT.

- **Reference**

  [8], [9] 2.3.5.6.1

- **Initial Condition**

  Preamble has been executed.

  IUT is master. Lower Tester is slave.

  FKC is the number of failed pairing attempts before the Upper Tester generates a new key pair as defined in the IXIT [7] entry and is used in Table 4.2.

  The Lower Tester generates and uses only private/public key pairs where bit 0 of the private key is set to 0.
• **Test Procedure**

![Diagram](image)

**Figure 4.2: SM/MAS/KDU/BI-01-C [LE Secure Connections Pairing, IUT Initiator – Lower Tester sends invalid public key]**

Execute Steps 1–5 for each round in Table 4.2, repeating the number of times as specified in Table 4.2.

1. The Upper Tester sends an HCI_Reset command to the IUT.
2. The IUT transmits a Pairing Request command with the SC bit of AuthReq set to ‘1’.
3. The Lower Tester responds with Pairing Response command with the SC bit of AuthReq set to ‘1’. Lower Tester also sets all bits in “Responder Key Distribution” field to ‘0’.
4. The IUT and Lower Tester perform the Public Key Exchange. The Lower Tester generates a new valid private/public key pair and modifies the keys as specified in Table 4.2. The Lower Tester verifies that these new coordinates are not on the curve before sending them; if accidentally the new coordinates are valid, then the generation procedure is repeated. The resulting invalid Public Key is sent over the air.
5. The Lower Tester continues the pairing procedure using the public key value sent over the air until the IUT fails the pairing procedure. In Authentication Stage 2, the Lower Tester either uses the computed DHKey or DHKey = 0 as specified in Table 4.2.

• **Expected Outcome**

**Pass Verdict**

The IUT fails the pairing procedure any time after receiving the invalid public key. If the IUT sends a Pairing Failed message then any reason code is allowed.

**Fail Verdict**

The IUT successfully completes the pairing procedure.
4.7.2 Re-encrypt an encrypted link with LTK

4.7.2.1 SM/SLA/KDU/BV-07-C [IUT Responder - Existing encrypted link is re-encrypted using LTK]

- Test Purpose
  Verify that the IUT correctly handles a requested encrypted session setup to use the distributed LTK, EDIV and Rand values when the key distribution phase has completed.

- Reference
  [1] 3.6.1

- Initial Condition
  Lower Tester and IUT have completed SM/SLA/KDU/BV-01-C [LE Legacy Pairing, IUT Responder – Lower Tester sets EncKey bit – Success] and have not disconnected the link.

  IUT is slave. Lower Tester is master.

- Test Procedure
  Lower Tester re-encrypts the link using the LTK EDIV and RAND values distributed by the IUT.

- Expected Outcome
  Pass Verdict

  The Lower Tester can re-encrypt the link successfully, i.e., the IUT sends an encrypted LL_START_ENC_RSP packet with the correct MIC, which is acknowledged by the Lower Tester.

4.8 Slave Initiated Security Request

The test group objective is to verify the correct implementation of the slave initiated security request.

4.8.1 Slave Initiated Pairing

4.8.1.1 SM/SLA/SIP/BV-01-C [Slave Initiates pairing]

- Test Purpose
  Verify that the IUT is able to initiate a pairing as a slave.

- Reference
  [1] 2.4.6

- Initial Condition
  Preamble has been executed.

  IUT and Lower Tester both have IO capability set as “KeyboardDisplay”.

  IUT is slave. Lower Tester is master.

  IUT is not bonded with the Lower Tester.
• Test Procedure
  1. Upper Tester commands the IUT to send ‘security request’ with MITM as ‘0’.
  2. Upon receiving the security request from the IUT, the Lower Tester initiates Just Works pairing.

• Test Condition
  It must be guaranteed that the IUT is able to send security request if requested via Upper Tester.

• Expected Outcome
  Pass Verdict
  IUT sends Security Request with no-MITM authentication requirement.
  Just Works Pairing has completed successfully.

4.8.1.2 SM/MAS/SIP/BV-02-C [Slave Initiates pairing – Master Response]

• Test Purpose
  Verify that the IUT, as master, is able to respond to slave initiated pairing.

• Reference
  [1] 2.4.6

• Initial Condition
  Preamble has been executed.
  IUT and Lower Tester both have IO capability set as “KeyboardDisplay”.
  IUT is master. Lower Tester is slave.
  IUT is not bonded with the Lower Tester.

• Test Procedure
  1. Lower Tester sends ‘security request’ with MITM as ‘0’ to IUT.
  2. Upon receiving the security request from the Lower Tester, the IUT initiates Just Works pairing or the IUT responds to the request with a Pairing Failure Response with the reason field set to ‘Pairing Not Supported.’

• Expected Outcome
  Pass Verdict
  Just Works Pairing has completed successfully, or
  The IUT response to the request with a Pairing Failure Response with the reason set to ‘Pairing Not Supported.’
4.8.2 Slave Initiated Encryption

4.8.2.1 SM/SLA/SIE/BV-01-C [Slave initiates Encryption]

• Test Purpose
  Verify that the IUT is able to initiate encryption as a slave.

• Reference
  [1] 2.4.6, 5.3.1.1

• Initial Condition
  Lower Tester and IUT have been bonded with exchanged security information with security property
  of MITM protection not required.

  Lower Tester and IUT both maintained the bond information.

  Lower Tester and IUT currently have established link layer connection without encryption and SMP
  fixed channel is ready.

  IUT and Lower Tester both have IO capability set as “NoInputNoOutput”.

  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Upper Tester commands the IUT to send ‘security request’ with MITM as ‘0’.
  2. Lower Tester starts link encryption procedure with bonded security information, and link is
     encrypted successfully.

• Test Condition
  It must be guaranteed that the IUT is able to send security request if requested via Upper Tester.

• Expected Outcome
  Pass Verdict
  IUT sends Security Request with required authentication requirement.

  Encryption procedure with LTK is performed correct.

4.9 Pairing Methods Using LE Secure Connections

4.9.1 Just Works (SCJW)

4.9.1.1 SM/MAS/SCJW/BV-01-C [Just Works, IUT Initiator, Secure Connections – Success]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections performs the Just Works or Numeric
  Comparison pairing procedure correctly as initiator.
• Reference

[8] 2.3.5.1, 2.3.5.6

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits Pairing Request command with:
   a. IO capability set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq Bonding Flags set to ‘00’, the MITM flag set to either ‘0’ for Just Works or ‘1’ for Numeric Comparison, Secure Connections flag set to ‘1’ and all the reserved bits are set to ‘0’

2. Lower Tester responds with a Pairing Response command, with:
   a. IO capability set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq Bonding Flags set to ‘00’, the MITM flag set to ‘0’, Secure Connections flag set to ‘1’ and all the reserved bits are set to ‘0’

3. IUT and Lower Tester perform phase 2 of the Just Works or Numeric Comparison pairing procedure according to the MITM flag and IO capabilities, and establish an encrypted link with the LTK generated in phase 2.

The test is repeated by the IUT to test all supported combinations of [8] Section 2.3.5.1, Table 2.8 which do not result in passkey entry.

• Expected Outcome

Pass verdict

The encryption procedure initiated by the IUT completes successfully.

The IUT can encrypt the link successfully using LE Secure Connections.

4.9.1.2  SM/SLA/SCJW/BV-02-C [Just Works, IUT Responder, Secure Connections – Success]

• Test Purpose

Verify that the IUT supporting LE Secure Connections is able to perform the Just Works or Numeric Comparison pairing procedure correctly when acting as responder.

• Reference

[8] 2.3.5.1, 2.3.5.6.2
• Initial Condition
Preamble has been executed.

IUT is slave. Lower Tester is master.

• Test Procedure
1. Lower Tester transmits Pairing Request command with:
   a. IO capability set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq Bonding Flags set to '00', MITM flag set to '0', Secure Connections flag set to '1'
      and all reserved bits are set to '0'

2. IUT responds with a Pairing Response command, with:
   a. IO capability set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq Bonding Flags set to '00', MITM flag set to either '0' for Just Works or '1' for
      Numeric Comparison, Secure Connections flag set to '1' and all reserved bits are set to
      '0'

3. IUT and Lower Tester perform phase 2 of the Just Works or Numeric Comparison pairing
   procedure according to the MITM flag and IO capabilities, and establish an encrypted link with the
   LTK generated in phase 2.

The test is repeated by the IUT to test all supported combinations of [8] Section 2.3.5.1, Table 2.8
which do not result in passkey entry.

• Expected Outcome
Pass verdict

The encryption procedure initiated by the Lower Tester completes successfully.

The IUT and the Lower Tester can encrypt the link successfully using LE Secure Connections.

4.9.1.3 SM/SLA/SCJW/BV-03-C [Just Works, IUT Responder, Secure Connections –
Handle AuthReq Flag RFU Correctly]

• Test Purpose
Verify that the IUT is able to perform the Just Works pairing procedure when receiving additional bits
set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as slave,
responder.

• Reference
[8] 2.3.5.1, 2.3.5.6.2

• Initial Condition
Preamble has been executed.

IUT is slave. Lower Tester is master.
• Test Procedure

1. Lower Tester transmits Pairing Request command with:
   a. IO Capability set to “NoInputNoOutput”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. MITM set to ‘0’ and all reserved bits are set to a random value.

2. IUT responds with a Pairing Response command, with:
   a. IO Capability set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. All reserved bits are set to ‘0’

3. IUT and Lower Tester perform phase 2 of the Just Works pairing and establish an encrypted link with the generated LTK.

   The test is repeated by the IUT until all possible combinations of reserved bits from the Lower Tester is tested.

• Expected Outcome

   Pass verdict

   The encryption procedure initiated by the Lower Tester completes successfully.

   The IUT and the Lower Tester can encrypt the link successfully.

4.9.1.4 SM/MAS/SCJW/BV-04-C [Just Works, IUT Initiator, Secure Connections – Handle AuthReq Flag RFU Correctly]

• Test Purpose

   Verify that the IUT is able to perform the Just Works pairing procedure when receiving additional bits set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as master, initiator.

• Reference

   [8] 2.3.5.1, 2.3.5.6.2

• Initial Condition

   Preamble has been executed.

   IUT is master. Lower Tester is slave.

• Test Procedure

   1. IUT transmits a Pairing Request command with:
      a. IO Capability set to any IO Capability
      b. OOB data flag set to 0x00 (OOB Authentication data not present)
      c. All reserved bits are set to ‘0’
2. Lower Tester responds with a Pairing Response command, with:
   a. IO Capability set to “NoInputNoOutput”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq bonding flag set to the value indicated in the IXIT [7] for ‘Bonding Flags’ and the
      MITM flag set to ‘0’ and all reserved bits are set to a random value.

3. IUT and the Lower Tester perform phase 2 of the Just Works pairing and establish an encrypted
   link with the generated LTK.
   The test is repeated by the IUT until all possible combinations of reserved bits of the Lower Tester are
   tested. If IUT supports Secure Connections the SC flag shall be set to 1 when testing all combinations
   of the reserved bits.

   • Expected Outcome
     
     Pass Verdict
     
     The encryption procedure initiated by the IUT completes successfully.
     The link is encrypted successfully.

4.9.1.5 SM/MAS/SCJW/BI-01-C [Just Works, IUT Initiator, Secure Connections – Pairing Failed]

   • Test Purpose
     
     Verify that the IUT supporting LE Secure Connections handles Just Works or Numeric Comparison
     pairing failures.

   • Reference
     
     [8] 3.5.5, 2.3.5.6.2

   • Initial Condition
     
     Preamble has been executed.
     
     IUT is master. Lower Tester is slave.

   • Test Procedure
     
     1. IUT transmits Pairing Request command with:
        a. IO capability is set to any IO capability
        b. OOB data flag set to 0x00 (OOB Authentication data not present)
        c. Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’

     2. Lower Tester responds with a Pairing Failed command with reason code ‘0x03’ (Authentication
        Requirements).

     3. The pairing process is aborted. IUT reports the failure to the Upper Tester.
        Run preamble to re-establish Initial Conditions

     4. Execute Step 1.
5. Lower Tester responds with a Pairing Failed command with reason code ‘0x08’ (Unspecified Reason).

6. The pairing process is aborted. IUT reports the failure to the Upper Tester.

Run preamble to re-establish Initial Conditions

7. Execute Step 1.
8. Lower Tester responds with a Pairing Failed command with reason code ‘0x05’ (Pairing Not Supported).
9. The pairing process is aborted. IUT reports the failure to the Upper Tester.

Run preamble to re-establish Initial Conditions

10. Execute Step 1.
11. Lower Tester responds with a Pairing Failed command with reason code ‘0x09’ (Repeated Attempts).
12. The pairing process is aborted. IUT reports the failure to the Upper Tester.
14. Lower Tester transmits Pairing Response command with:
   a. IO capability is set to any IO capability
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’

15. Lower Tester responds with a Pairing Failed command in phase 2 with reason code ‘0x0C (Numeric Comparison Failed).
16. The pairing process is aborted. IUT reports the failure to the Upper Tester.

• Expected Outcome

Pass verdict

For each pairing failure, the IUT detects the failures reported by the responder and responds correctly to the Lower Tester.

For each pairing failure, the IUT aborts the pairing process and reports the failure to the Upper Tester.

4.9.1.6 SM/SLA/SCJW/BJI-02-C [Just Works, IUT Responder, Secure Connections – Confirm Check Failure]

• Test Purpose

Verify that the IUT supporting LE Secure Connections handles Just Works pairing failure as responder correctly, when the Lower Tester does not confirm “OK”.

• Reference

[8] 2.3.5.1, 2.3.5.6.2

• Initial Condition

Preamble has been executed.

IUT is slave. Lower Tester is master.
• Test Procedure

1. Lower Tester transmits a Pairing Request command with:
   - a. IO capability set to “NoInputNoOutput”
   - b. OOB data flag set to 0x00 (OOB Authentication data not present)
   - c. AuthReq bonding flag set to ‘01’, and the MITM flag set to ‘0’, Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’

2. IUT responds with a Pairing Response command, with:
   - a. IO capability set to any IO capability
   - b. OOB data flag set to 0x00 (OOB Authentication data not present)
   - c. Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’

3. During phase 2 of the just works pairing procedure, the Lower Tester transmits a Pairing Failed command with (Confirm Value Failed).

• Expected Outcome

   Pass verdict

   IUT aborts the pairing.

4.9.2 Passkey Entry (SCPK)

4.9.2.1 SM/MAS/SCPK/BV-01-C [Passkey Entry, IUT Initiator, Secure Connections – Success]

• Test Purpose

Verify that the IUT supporting LE Secure Connections performs the Passkey Entry pairing procedure correctly as master, initiator.

• Reference

[8] 2.3.5.1, 2.3.5.6.3

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with:
   - a. IO capability set to “DisplayOnly” or “KeyboardOnly”
   - b. OOB data flag set to 0x00 (OOB Authentication data not present)
   - c. AuthReq bonding flag set to ‘00’, the MITM flag set to ‘0’ and Secure Connections flag set to ‘1’. Keypress bit is set to ‘1’ if supported

2. Lower Tester responds with a Pairing Response command, with:
   - a. IO capability set to “KeyboardOnly”
b. OOB data flag set to 0x00 (OOB Authentication data not present)

c. AuthReq bonding flag set to ‘00’, the MITM flag set to ‘1’, Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’. Keypress bit is set to ‘1’ if supported by the IUT.

3. During the phase 2 pairing, the IUT displays the 6-digit passkey while the Lower Tester prompts user to enter the 6-digit passkey. If the IUT’s IO capabilities are “KeyboardOnly” the passkey is not displayed and both IUT and Lower Tester enter the same 6-digit passkey. If Keypress bit is set, pairing keypress notifications are sent by the Lower Tester.

4. IUT and Lower Tester use the same 6-digit passkey.

5. IUT and Lower Tester perform phase 2 of the Passkey Entry pairing procedure and establish an encrypted link with the LTK generated in phase 2.

• Expected Outcome

Pass verdict

The IUT can encrypt the link successfully using LE Secure Connections.

• Notes

This test also covers the use of the keypress bit.

4.9.2.2 SM/SLA/SCPK/BV-02-C [Passkey Entry, IUT Responder, Secure Connections – Success]

• Test Purpose

Verify that the IUT supporting LE Secure Connections is able to perform the Passkey Entry pairing procedure correctly when acting as slave, responder.

• Reference

[8] 2.3.5.1, 2.3.5.6.3

• Initial Condition

Preamble has been executed.

IUT is slave. Lower Tester is master.

• Test Procedure

1. Lower Tester initiates a Pairing Request command with:

   a. IO capability set to “KeyboardDisplay”

   b. OOB data flag set to 0x00 (OOB Authentication data not present)

   c. AuthReq bonding flag set to the value indicated in the IXIT [7] for ‘Bonding Flags’, and the MITM flag set to ‘1’ Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’

2. IUT responds with a Pairing Response command, with:

   a. IO capability set to “KeyboardOnly” or “KeyboardDisplay” or “DisplayYesNo” or “DisplayOnly”

   b. OOB data flag set to 0x00 (OOB Authentication data not present)
c. Secure Connections flag set to ‘1’. Keypress bit is set to ‘1’ if supported by IUT

3. During the phase 2 passkey pairing process, Lower Tester displays the 6-digit passkey while the IUT prompts user to enter the 6-digit passkey. If the IO capabilities of the IUT are “DisplayYesNo” or “DisplayOnly” the IUT displays the 6-digit passkey while the Lower Tester enters the 6-digit passkey. If Keypress bit is set, pairing keypress notifications are send by the IUT.

4. IUT and Lower Tester use the same pre-defined 6-digit passkey.
5. IUT and Lower Tester perform phase 2 of the LE pairing and establish an encrypted link with the LTK generated in phase 2.

The test is repeated where the Lower Tester also sets the Keypress bit to ‘1’ if supported by the IUT in step 1c.

• Expected Outcome
  Pass verdict
  The master can encrypt the link successfully with LE Secure Connections.
  IUT only sends keypress notification if supported by the Lower Tester.

• Notes
  This test also covers the use of the keypress bit.

4.9.2.3 SM/SLA/SCPK/BV-03-C [Passkey Entry, IUT Responder, Secure Connections – Handle AuthReq Flag RFU Correctly]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections is able to perform the Passkey Entry pairing procedure when receiving additional bits set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as slave, responder.

• Reference
  [8] 2.3.5.1, 2.3.5.6.3

• Initial Condition
  Preamble has been executed.

  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Lower Tester transmits Pairing Request command with:
     a. IO Capability set to "KeyboardOnly"
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. MITM set to ‘1’ and all reserved bits are set to a random value
  2. IUT responds with a Pairing Response command, with:
     a. IO Capability set to “KeyboardOnly” or “DisplayOnly”
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
c. All reserved bits are set to ‘0’

3. IUT and Lower Tester perform phase 2 of the Passkey Entry pairing and establish an encrypted link with the generated LTK.

Test is repeated until all possible combinations of reserved bits of the Lower Tester are tested.

• Expected Outcome

Pass Verdict

The encryption procedure initiated by the Lower Tester completes successfully.

The Lower Tester can encrypt the link successfully.

4.9.2.4 SM/MAS/SCPK/BV-04-C [Passkey Entry, IUT Initiator, Secure Connections – Handle AuthReq Flag RFU Correctly]

• Test Purpose

Verify that the IUT supporting LE Secure Connections is able to perform the Passkey Entry pairing procedure when receiving additional bits set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as master, initiator.

• Reference

[8] 2.3.5.1, 2.3.5.6.3

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with:
   a. IO Capability set to “DisplayOnly” or “DisplayYesNo” or “KeyboardOnly” or “KeyboardDisplay”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. All reserved bits are set to ‘0’

2. Lower Tester responds with a Pairing Response command, with:
   a. IO Capability set to “KeyboardOnly”
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq bonding flag set to the value indicated in the IXIT [7] for ‘Bonding Flags’ and the MITM flag set to ‘1’ and all reserved bits are set to a random value.

3. IUT and the Lower Tester perform phase 2 of the Passkey Entry pairing and establish an encrypted link with the generated LTK.

The test is repeated by the IUT until all possible combinations of reserved bits of the Lower Tester are tested. If IUT supports LE Secure Connections, the SC flag shall be set to 1 when testing all combinations of the reserved bits.
• Expected Outcome
  Pass Verdict

  The encryption procedure initiated by the IUT completes successfully.

  The link is encrypted successfully.

4.9.2.5  SM/MAS/SCPK/BI-01-C [Passkey Entry, IUT Initiator, Secure Connections – Pairing Failed]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections handles Passkey Entry pairing failures.

• Reference
  [8] 2.3.5.1, 2.3.5.6.3

• Initial Condition
  Preamble has been executed.

  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits Pairing Request command with:
     a. IO capability is set to “KeyboardOnly” or “DisplayOnly” or “DisplayYesNo” or “DisplayOnly”
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’
  2. Lower Tester responds with a Pairing Failed command with reason code ‘0x03’ (Authentication Requirements).
  3. The pairing process is aborted. IUT reports the failure to the Upper Tester.

  Run preamble to re-establish Initial Conditions

  4. Execute Step 1.
  5. Lower Tester responds with a Pairing Failed command with reason code ‘0x08’ (Unspecified Reason).
  6. The pairing process is aborted. IUT reports the failure to the Upper Tester.

  Run preamble to re-establish Initial Conditions

  7. Execute Step 1.
  8. Lower Tester responds with a Pairing Failed command with reason code ‘0x05’ (Pairing Not Supported).
  9. The pairing process is aborted. IUT reports the failure to the Upper Tester.

  Run preamble to re-establish Initial Conditions

  10. Execute Step 1.
11. Lower Tester responds with a Pairing Failed command with reason code '0x09' (Repeated Attempts).
12. The pairing process is aborted. IUT reports the failure to the Upper Tester.
14. Lower Tester transmits Pairing Response command with:
   a. IO capability is set to "KeyboardOnly"
   b. OOB data flag set to 0x00 (OOB Authentication data not present)
   c. AuthReq bonding flag set to the value indicated in the IXIT [7] for 'Bonding Flags', and the MITM flag set to '1', Secure Connections flag set to '1' and all reserved bits are set to '0'.
15. Lower Tester responds with a Pairing Failed command in phase 2 with reason code '0x01 (Passkey Entry Failed).
16. The pairing process is terminated. IUT reports the failure to the Upper Tester.

• Expected Outcome
  
  Pass verdict

For each pairing failure, the IUT detects the failures reported by the responder and responds correctly to the Lower Tester.

For each pairing failure, the IUT aborts the pairing process and reports the failure to the Upper Tester.

4.9.2.6 SM/MAS/SCPK/BI-02-C [Passkey Entry, IUT Initiator, Secure Connections – Failure]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections handles Passkey Entry pairing failure as initiator correctly.

• Reference
  [8] 2.3.5.1, 2.3.5.6.3

• Initial Condition
  Preamble has been executed.

  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits a Pairing Request command with:
     a. IO capability set to “KeyboardOnly” or “DisplayOnly”
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. AuthReq bonding flag set to ‘01’, Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’
  2. Lower Tester responds with a Pairing Response command, with:
     a. IO capability set to “KeyboardOnly”
b. OOB data flag set to 0x00 (OOB Authentication data not present)
c. Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’
d. MITM set to ‘1’

3. During phase 2 of the pass key entry pairing procedure, the Lower Tester transmits an incorrect Pairing Confirm Value.
4. The IUT detects the incorrect confirm value and sends a Pairing Failed command with ‘0x04’ (Confirm Value Failed).

• Expected Outcome
   Pass verdict
   IUT terminates the pairing.

4.9.2.7 SM/SLA/SCPK/BI-03-C [Passkey Entry, IUT Responder, Secure Connections – Confirm Value Check Failure] 

• Test Purpose
   Verify that the IUT supporting LE Secure Connections handles Passkey Entry pairing failure with confirm value check as responder correctly.

• Reference
   [8] 2.3.5.1, 2.3.5.6.3

• Initial Condition
   Preamble has been executed.

   IUT is slave. Lower Tester is master.

• Test Procedure
   1. Lower Tester transmits a Pairing Request command with:
      a. IO capability set to “KeyboardOnly”
      b. OOB data flag set to 0x00 (OOB Authentication data not present)
      c. AuthReq bonding flag set to ‘01’, and the MITM flag set to ‘1’, Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’
   2. IUT responds with a Pairing Response command, with:
      a. IO capability set to “KeyboardOnly” or “KeyboardDisplay” or “Display YesNo” or “DisplayOnly”
      b. OOB data flag set to 0x00 (OOB Authentication data not present)
      c. Secure Connections flag set to ‘1’ and all reserved bits are set to ‘0’
   3. During phase 2 of the pass key entry pairing procedure, the Lower Tester transmits an incorrect Pairing Confirm Value.
   4. The IUT detects the incorrect confirm value and sends a Pairing Failed command with ‘0x04’ (Confirm Value Failed).
• Expected Outcome
  
  Pass verdict
  
  IUT terminates the pairing.

4.9.2.8  SM/SLA/SCP/BI-04-C [Passkey Entry, IUT Responder, Secure Connections – Pairing Failed]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections handles Passkey Entry pairing failures.

• Reference
  [8] 2.3.5.1, 2.3.5.6.3

• Initial Condition
  Preamble has been executed.

  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Lower Tester transmits Pairing Request command with:
     a. IO capability is set to "KeyboardOnly" or "DisplayOnly"
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. Secure Connections flag set to '1' and all reserved bits are set to '0'
  2. IUT transmits Pairing Response command with:
     a. IO capability is set to "KeyboardOnly"
     b. OOB data flag set to 0x00 (OOB Authentication data not present)
     c. Secure Connections flag set to '1' and all reserved bits are set to '0'
     d. MITM set to '1'
  3. Lower Tester responds with a Pairing Failed command with reason code '0x03' (Authentication Requirements).
  4. The pairing process is aborted. IUT reports the failure to the Upper Tester.

Run preamble to re-establish Initial Conditions

5. Execute Step 1 and 2.

6. Lower Tester responds with a Pairing Failed command with reason code '0x08' (Unspecified Reason).

7. The pairing process is aborted. IUT reports the failure to the Upper Tester.

Run preamble to re-establish Initial Conditions

8. Execute Step 1 and 2.

9. Lower Tester responds with a Pairing Failed command with reason code '0x05' (Pairing Not Supported).

10. The pairing process is aborted. IUT reports the failure to the Upper Tester.
Run preamble to re-establish Initial Conditions

11. Execute Step 1 and 2.
12. Lower Tester responds with a Pairing Failed command with reason code ‘0x09’ (Repeated Attempts).
13. The pairing process is aborted. IUT reports the failure to the Upper Tester.
14. Execute Step 1 and 2.
15. Lower Tester responds with a Pairing Failed command in phase 2 with reason code ‘0x01’ (Passkey Entry Failed).
16. The pairing process is terminated. IUT reports the failure to the Upper Tester.

• Expected Outcome

Pass verdict

For each pairing failure, the IUT detects the failures reported by the initiator and responds correctly to the Lower Tester.

For each pairing failure, the IUT terminates the pairing process and reports the failure to the Upper Tester.

4.9.3 Out of Band (SCOB)

4.9.3.1 SM/MAS/SCOB/BV-01-C [Out of Band, IUT Initiator, Secure Connections – Success]

• Test Purpose

Verify that the IUT supporting LE Secure Connections performs the Out-of-Band pairing procedure correctly as master, initiator.

• Reference

[8] 2.3.5.1, 2.3.5.6.4

• Initial Condition

Preamble has been executed.

IUT is master. Lower Tester is slave.

• Test Procedure

1. IUT transmits a Pairing Request command with OOB data flag set to either 0x00 or 0x01, and Secure Connections flag set to ‘1’.
2. Lower Tester responds with a Pairing Response command with Secure Connections flag set to ‘1’ and OOB data flag set to either 0x00 or 0x01.
3. IUT uses the 128-bit value generated by the Lower Tester as the confirm value. Similarly, the Lower Tester uses the 128-bit value generated by the IUT as the confirm value.
4. IUT and Lower Tester perform phase 2 of the pairing process and establish an encrypted link with an LTK generated using the OOB data in phase 2.

The test is repeated with OOB data flag combinations set to {0x01, 0x01}, {0x01, 0x00} and {0x00, 0x01}
• Expected Outcome
  Pass verdict
  The IUT can encrypt the link successfully as a Secure Connection.
  The IUT indicates successful Secure Connections pairing to the Upper Tester.

• Notes
  OOB data are exchanged out of band.

4.9.3.2  SM/SLA/SCOB/BV-02-C [Out of Band, IUT Responder, Secure Connections – Success]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections is able to perform the Out-of-Band pairing procedure correctly when acting as slave, responder.

• Reference
  [8] 2.3.5.1, 2.3.5.6.4

• Initial Condition
  Preamble has been executed.
  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Lower Tester transmits a Pairing Request command with OOB data flag set to either 0x00 or 0x01, and Secure Connections flag set to ‘1’.
  2. IUT responds with a Pairing Response command with Secure Connections flag set to ‘1’ and OOB data flag set to either 0x00 or 0x01.
  3. IUT uses the 128-bit value generated by the Lower Tester as the confirm value. Similarly, the Lower Tester uses the 128-bit value generated by the IUT as the confirm value.
  4. IUT and Lower Tester perform phase 2 of the pairing process and establish an encrypted link with an LTK generated using the OOB data in phase 2.

  The test is repeated with OOB data flag combinations set to {0x01, 0x01}, {0x01, 0x00} and {0x00, 0x01}

• Expected Outcome
  Pass verdict
  The Initiator can encrypt the link successfully as Secure Connections.
  The IUT indicates successful Secure Connections pairing to the Upper Tester.
4.9.3.3 SM/SLA/SCOB/BV-03-C [Out of Band, IUT Responder, Secure Connections – Handle AuthReq Flag RFU Correctly]

- **Test Purpose**
  Verify that the IUT supporting LE Secure Connections is able to perform the Out-of-Band pairing procedure when receiving additional bits set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as slave, responder.

- **Reference**
  [1] 2.3.5.2, 2.4.6, 5.3.1 & 5.3.2
  [8] 2.3.5.1, 2.3.5.6.4

- **Initial Condition**
  Preamble has been executed.
  
  IUT is slave. Lower Tester is master.

- **Test Procedure**
  1. Lower Tester transmits Pairing Request command with:
     a. IO Capability set to any IO capability
     b. OOB data flag set to 0x01 (OOB Authentication data from remote device present)
     c. MITM set to ‘0’, Secure Connections flag is set to ‘1’, and all reserved bits are set to a random value.

  2. IUT responds with a Pairing Response command, with:
     a. IO Capability set to any IO capability
     b. OOB data flag set to 0x01 (OOB Authentication data present)
     c. Secure Connections flag is set to ‘1’, All reserved bits are set to ‘0’

  3. IUT and Lower Tester perform phase 2 of the OOB authenticated pairing and establish an encrypted link with the generated LTK.
  
  Test is repeated until all possible combinations of reserved bits of the Lower Tester are tested.

- **Expected Outcome**
  
  **Pass verdict**
  
  The encryption procedure initiated by the Lower Tester completes successfully.
  
  The IUT and the Lower Tester can encrypt the link successfully.
4.9.3.4 SM/MAS/SCOB/BV-04-C [Out of Band, IUT Initiator, Secure Connections – Handle AuthReq Flag RFU Correctly]

- **Test Purpose**
  
  Verify that the IUT supporting LE Secure Connections is able to perform the Out-of-Band pairing procedure when receiving additional bits set in the AuthReq flag. Reserved For Future Use bits are correctly handled when acting as master, initiator.

- **Reference**
  
  [8] 2.3.5.1, 2.3.5.6.4

- **Initial Condition**
  
  Preamble has been executed.

  IUT is master. Lower Tester is slave.

- **Test Procedure**
  
  1. IUT transmits Pairing Request command with:
    
    a. IO Capability set to any IO capability
    
    b. OOB data flag set to 0x01 (OOB Authentication data present)
    
    c. MITM set to '0', Secure Connections flag is set to '1', and all reserved bits are set to '0'

  2. Lower Tester responds with a Pairing Response command, with:
    
    a. IO Capability set to any IO capability
    
    b. OOB data flag set to 0x01 (OOB Authentication data present)
    
    c. Secure Connections flag is set to '1', and all reserved bits are set to a random value.

  3. IUT and Lower Tester perform phase 2 of the OOB authenticated pairing and establish an encrypted link with the generated LTK.

  Test is repeated until all possible combinations of reserved bits of the Lower Tester are tested.

- **Expected Outcome**
  
  **Pass verdict**

  The encryption procedure initiated by the IUT completes successfully.

  The IUT can encrypt the link successfully.

4.9.3.5 SM/MAS/SCOB/BI-01-C [Out of Band, IUT Initiator, Secure Connections – Failure]

- **Test Purpose**
  
  Verify that the IUT supporting LE Secure Connections handles Out-of-Band pairing failure as initiator correctly.

- **Reference**
  
  [8] 2.3.5.1, 2.3.5.6.4
• **Initial Condition**

Preamble has been executed.

IUT is master. Lower Tester is slave.

• **Test Procedure**

1. IUT transmits Pairing Request command with:
   a. IO capability is set to any value
   b. OOB data flag set to 0x01 (OOB Authentication data from remote device present)
   c. Secure Connections flag set to '1' and all reserved bits are set to '0'

2. Lower Tester responds with a Pairing Failed command with reason code '0x03' (Authentication Requirements).

3. The pairing process is aborted. IUT reports the failure to the Upper Tester.

   Run preamble to re-establish Initial Conditions

4. Execute Step 1.

5. Lower Tester responds with a Pairing Failed command with reason code '0x08' (Unspecified Reason).

6. The pairing process is aborted. IUT reports the failure to the Upper Tester.

   Run preamble to re-establish Initial Conditions

7. Execute Step 1.

8. Lower Tester responds with a Pairing Failed command with reason code '0x05' (Pairing Not Supported).

9. The pairing process is aborted. IUT reports the failure to the Upper Tester.

   Run preamble to re-establish Initial Conditions

10. Execute Step 1.

11. Lower Tester responds with a Pairing Failed command with reason code '0x09' (Repeated Attempts).

12. The pairing process is aborted. IUT reports the failure to the Upper Tester.


14. Lower Tester transmits Pairing Response command with:
   a. IO capability is set to any value
   b. OOB data flag set to 0x01 (OOB Authentication data present)
   c. Secure Connections flag set to '1' and all reserved bits are set to '0'

15. Lower Tester responds with a Pairing Failed command in phase 2 with reason code '0x02 (OOB Not Available).

16. The pairing process is terminated. IUT reports the failure to the Upper Tester.

• **Expected Outcome**

  Pass verdict

  For each pairing failure, the IUT detects the failures reported by the responder and responds correctly to the Lower Tester.
For each pairing failure, the IUT terminates the pairing process and reports the failure to the Upper Tester.

4.9.3.6 SM/SLA/SCOB/BI-02-C [Out of Band, IUT Responder, Secure Connections – Failure]

- **Test Purpose**
  Verify that the IUT supporting LE Secure Connections handles Out-of-Band pairing failure as responder correctly.

- **Reference**
  [8] 2.3.5.1, 2.3.5.6.4

- **Initial Condition**
  Preamble has been executed.
  IUT is slave. Lower Tester is master.

- **Test Procedure**
  1. Lower Tester transmits Pairing Request command with:
     a. IO capability is set to any value
     b. OOB data flag set to 0x01 (OOB Authentication data present)
     c. Secure Connections flag set to '1' and all reserved bits are set to '0'
  2. IUT transmits Pairing Response command with:
     a. IO capability is set to any value
     b. OOB data flag set to 0x01 (OOB Authentication data from remote device present)
     c. Secure Connections flag set to '1' and all reserved bits are set to '0'
  3. Lower Tester responds with a Pairing Failed command with reason code '0x03' (Authentication Requirements).
  4. The pairing process is aborted. IUT reports the failure to the Upper Tester.
  Run preamble to re-establish Initial Conditions
  5. Execute steps 1 and 2.
  6. Lower Tester responds with a Pairing Failed command with reason code '0x08' (Unspecified Reason).
  7. The pairing process is aborted. IUT reports the failure to the Upper Tester.
  Run preamble to re-establish Initial Conditions
  8. Execute steps 1 and 2.
  9. Lower Tester responds with a Pairing Failed command with reason code '0x05' (Pairing Not Supported).
  10. The pairing process is aborted. IUT reports the failure to the Upper Tester.
  Run preamble to re-establish Initial Conditions
  11. Execute steps 1 and 2.
12. Lower Tester responds with a Pairing Failed command with reason code ‘0x09’ (Repeated Attempts).
13. The pairing process is aborted. IUT reports the failure to the Upper Tester.
14. Execute steps 1 and 2.
15. Lower Tester responds with a Pairing Failed command in phase 2 with reason code ‘0x02 (OOB Not Available).
16. The pairing process is terminated. IUT reports the failure to the Upper Tester.

• Expected Outcome

Pass verdict

For each pairing failure, the IUT detects the failures reported by the initiator and responds correctly to the Lower Tester.

For each pairing failure, the IUT terminates the pairing process and reports the failure to the Upper Tester.

4.9.3.7 SM/SLA/SCOB/BI-03-C [Out of Band, IUT Responder, Secure Connections – Pairing Failed]

• Test Purpose

Verify that the IUT supporting LE Secure Connections handles Out-of-Band pairing failures.

• Reference

[8] 2.3.5.1, 2.3.5.6.4

• Initial Condition

Preamble has been executed.

Lower Tester has sent the wrong OOB data to IUT.

IUT is slave. Lower Tester is master.

• Test Procedure

1. Lower Tester transmits Pairing Request command with OOB data flag set to 0x01 and Secure Connections flag set to ‘1’.
2. Responder responds with a Pairing Response command, with OOB data flag to set 0x01 and Secure Connections flag set to ‘1’.
3. The IUT detects the incorrect confirm value and sends Pairing Failed (“Confirm Value Failed”) and the Lower Tester initiates disconnect.

• Expected Outcome

Pass verdict

IUT detects the mismatch of confirm value, sends ‘Pairing Failed’ and the Lower Tester disconnects the link.
4.9.3.8 SM/MAS/SCOB/BI-04-C [Out of Band, IUT Initiator, Secure Connections – Pairing Failed]

- Test Purpose
  Verify that the IUT supporting LE Secure Connections handles Out-of-Band pairing failures.

- Reference
  [8] 2.3.5.1, 2.3.5.6.4

- Initial Condition
  Preamble has been executed.
  Lower Tester has sent the wrong OOB data to IUT.
  IUT is master. Lower Tester is slave.

- Test Procedure
  1. IUT transmits Pairing Request command with OOB data flag set to 0x01 and Secure Connections flag set to ‘1’
  2. Lower Tester responds with a Pairing Response command, with OOB data flag to set 0x01 and Secure Connections flag set to ‘1’
  3. The IUT detects the incorrect confirm value and sends Pairing Failed (“Confirm Value Failed”) and the Lower Tester initiates disconnect.

- Expected Outcome
  Pass verdict
  IUT detects the mismatch of confirm value, sends ‘Pairing Failed’ and the Lower Tester disconnects the link.

4.9.4 Cross Transport Key Derivation (SCCT)

4.9.4.1 SM/MAS/SCCT/BV-01-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections – Derive LE LTK from BR/EDR Link Key]

- Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive LE LTK from the BR/EDR Link Key.

- Reference
  [8] 2.3.5.7.2

- Initial Condition
  IUT and Lower Tester have paired over BR/EDR.
  IUT is master. Lower Tester is slave.
• Test Procedure
  1. IUT transmits Pairing Request command with the EncKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  2. Lower Tester responds with a Pairing Response command, with the EncKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the BR/EDR transport.
  5. The IUT and the Lower Tester connect on the LE transport and encrypt the link using the derived LTK.

• Expected Outcome
  Pass verdict

  IUT derives the LE LTK from the BR/EDR Link Key.

4.9.4.2 SM/SLA/SCCT/BV-02-C [Cross Transport Key Derivation, IUT Responder, Secure Connections – Derive LE LTK from BR/EDR Link Key]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive LE LTK from the BR/EDR Link Key.

• Reference
  [8] 2.3.5.7.2

• Initial Condition
  IUT and Lower Tester have paired over BR/EDR.

  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Lower Tester transmits Pairing Request command with EncKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  IUT responds with a Pairing Response command, with the EncKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR
  2. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  3. The Lower Tester or IUT disconnects the BR/EDR transport.
  4. The IUT and the Lower Tester connect on the LE transport and encrypt the link using the derived LTK.

• Expected Outcome
  Pass verdict

  IUT derives the LE LTK from the BR/EDR Link Key.
4.9.4.3 SM/MAS/SCCT/BV-03-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections – Derive LE LTK from BR/EDR Link Key Using h6]

- **Test Purpose**
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h6.

- **Reference**
  [8] 2.3.5.7, 2.4.2.5

- **Initial Condition**
  IUT is master. Lower Tester is slave.

- **Test Procedure**
  1. IUT transmits Pairing Request command with the CT2 bit in the AuthReq field set to ‘1’, and the EncKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  2. Lower Tester responds with a Pairing Response command with the CT2 bit in the AuthReq field set to ‘0’, and the EncKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the BR/EDR transport.
  5. The IUT and the Lower Tester connect on the LE transport and encrypt the link using the derived LTK.

- **Expected Outcome**
  Pass verdict
  IUT derives the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h6.

4.9.4.4 SM/SLA/SCCT/BV-04-C [Cross Transport Key Derivation, IUT Responder, Secure Connections – Derive LE LTK from BR/EDR Link Key Using h6]

- **Test Purpose**
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h6.

- **Reference**
  [8] 2.3.5.7, 2.4.2.5

- **Initial Condition**
  IUT is slave. Lower Tester is master.

- **Test Procedure**
  1. Lower Tester transmits Pairing Request command with the CT2 bit in the AuthReq field set to ‘0’, and the EncKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
2. IUT responds with a Pairing Response command with the CT2 bit in the AuthReq field set to ‘1’, and the EncKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
4. The Lower Tester or IUT disconnects the BR/EDR transport.
5. The IUT and the Lower Tester connect on the LE transport and encrypt the link using the derived LTK.

• Expected Outcome
  Pass verdict

IUT derives the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h6.

4.9.4.5 SM/MAS/SCCT/BV-05-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections – Derive LE LTK from BR/EDR Link Key Using h7]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h7.

• Reference
  [8] 2.3.5.7, 2.4.2.5

• Initial Condition
  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits Pairing Request command with the CT2 bit in the AuthReq field set to ‘1’, and the EncKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  2. Lower Tester responds with a Pairing Response command with the CT2 bit in the AuthReq field set to ‘1’, and the EncKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the BR/EDR transport.
  5. The IUT and the Lower Tester connect on the LE transport and encrypt the link using the derived LTK.

• Expected Outcome
  Pass verdict

IUT derives the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h7.

4.9.4.6 SM/SLA/SCCT/BV-06-C [Cross Transport Key Derivation, IUT Responder, Secure Connections – Derive LE LTK from BR/EDR Link Key Using h7]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h7.
• Reference
  [8] 2.3.5.7, 2.4.2.5

• Initial Condition
  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Lower Tester transmits Pairing Request command with the CT2 bit in the AuthReq field set to ‘1’, and the EncKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  2. IUT responds with a Pairing Response command with the CT2 bit in the AuthReq field set to ‘1’, and the EncKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over BR/EDR.
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the BR/EDR transport.
  5. The IUT and the Lower Tester connect on the LE transport and encrypt the link using the derived LTK.

• Expected Outcome
  Pass verdict

  IUT derives the LE LTK from the BR/EDR Link Key using Link Key Conversion Function h7.

4.9.4.7  SM/MAS/SCCT/BV-07-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections – Derive BR/EDR Link Key from LE LTK Using h6]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h6.

• Reference
  [8] 2.3.5.7, 2.4.2.4

• Initial Condition
  IUT is master. Lower Tester is slave.

• Test Procedure
  1. IUT transmits Pairing Request command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘1’, and the LinkKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over LE.
  2. Lower Tester responds with a Pairing Response command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘0’, and the LinkKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over LE.
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the LE transport.
  5. The IUT and the Lower Tester connect on the BR/EDR transport and encrypt the link using the derived Link Key.
- Expected Outcome
  Pass verdict

  IUT derives the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h6.

4.9.4.8 SM/SLA/SCCT/BV-08-C [Cross Transport Key Derivation, IUT Responder, Secure Connections – Derive BR/EDR Link Key from LE LTK Using h6]

- Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h6.

- Reference
  [8] 2.3.5.7, 2.4.2.4

- Initial Condition
  IUT is slave. Lower Tester is master.

- Test Procedure
  1. Lower Tester transmits Pairing Request command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘0’, and the LinkKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over LE.
  2. IUT responds with a Pairing Response command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘1’, and the LinkKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over LE.
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the LE transport.
  5. The IUT and the Lower Tester connect on the BR/EDR transport and encrypt the link using the derived Link Key.

- Expected Outcome
  Pass verdict

  IUT derives the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h6.

4.9.4.9 SM/MAS/SCCT/BV-09-C [Cross Transport Key Derivation, IUT Initiator, Secure Connections – Derive BR/EDR Link Key from LE LTK Using h7]

- Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h7.

- Reference
  [8] 2.3.5.7, 2.4.2.4

- Initial Condition
  IUT is master. Lower Tester is slave.
• Test Procedure
  1. IUT transmits Pairing Request command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘1’, and the LinkKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over LE.
  2. Lower Tester responds with a Pairing Response command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘1’, and the LinkKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over LE.
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the LE transport.
  5. The IUT and the Lower Tester connect on the BR/EDR transport and encrypt the link using the derived Link Key.

• Expected Outcome
  Pass verdict

IUT derives the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h7.

4.9.4.10 SM/SLA/SCCT/BV-10-C [Cross Transport Key Derivation, IUT Responder, Secure Connections – Derive BR/EDR Link Key from LE LTK Using h7]

• Test Purpose
  Verify that the IUT supporting LE Secure Connections and being a BR/EDR/LE device can derive the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h7.

• Reference
  [8] 2.3.5.7, 2.4.2.4

• Initial Condition
  IUT is slave. Lower Tester is master.

• Test Procedure
  1. Lower Tester transmits Pairing Request command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘1’, and the LinkKey bit in the Initiator Key Distribution/Generation field set to ‘1’ on SMP over LE.
  2. IUT responds with a Pairing Response command with the SC bit in the AuthReq field set to ‘1’, the CT2 bit in the AuthReq field set to ‘1’, and the LinkKey bit in the Responder Key Distribution/Generation field set to ‘1’ on SMP over LE.
  3. The IUT optionally distributes the negotiated keys such as the IRK, CSRK.
  4. The Lower Tester or IUT disconnects the LE transport.
  5. The IUT and the Lower Tester connect on the BR/EDR transport and encrypt the link using the derived Link Key.

• Expected Outcome
  Pass verdict

IUT derives the BR/EDR Link Key from the LE LTK using Link Key Conversion Function h7.
5 Test Case Mapping

The Test Case Mapping Table (TCMT) maps test cases to specific capabilities in the ICS. Profiles, protocols and services may define multiple roles, and it is possible that a product may implement more than one role. The product shall be tested in all roles for which support is declared in the ICS document.

The columns for the TCMT are defined as follows:

**Item:** contains an y/x reference, where y corresponds to the table number and x corresponds to the feature number as defined in the ICS Proforma for SM [3]. If the item is defined with Protocol, Profile or Service abbreviation before y/x, the table and feature number referenced are defined in the abbreviated ICS proforma document.

**Feature:** recommended to be the primary feature defined in the ICS being tested or may be the test case name.

**Test Case(s):** the applicable test case identifiers required for Bluetooth Qualification if the corresponding y/x references defined in the Item column are supported.


<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Role</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(SM 1/1 OR SM 1/2) AND SM 4/2 AND SM 6/1</td>
<td>Signing – Generation</td>
<td>SM/MAS/SIGN/BV-01-C</td>
</tr>
<tr>
<td>(SM 1/1 OR SM 1/2) AND SM 4/2 AND SM 6/2</td>
<td>Signing – Resolving</td>
<td>SM/MAS/SIGN/BV-03-C, SM/MAS/SIGN/BI-01-C</td>
</tr>
<tr>
<td>SM 1/1</td>
<td>Initiator tests</td>
<td>SM/MAS/PROT/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 7/1</td>
<td>Master Key Distribution - Encryption Key bit</td>
<td>SM/MAS/KDU/BV-06-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 7/2</td>
<td>Master Key Distribution - Identity Key bit</td>
<td>SM/MAS/KDU/BV-05-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 7/3</td>
<td>Master Key Distribution - Signing Key bit</td>
<td>SM/MAS/KDU/BV-04-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 7/2 AND SM 2/5</td>
<td>Master Key Distribution - Identity Key bit, LE Secure Connections</td>
<td>SM/MAS/KDU/BV-10-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 7/3 AND SM 2/5</td>
<td>Master Key Distribution - Signing Key bit, LE Secure Connections</td>
<td>SM/MAS/KDU/BV-11-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 3/1</td>
<td>Initiate Encryption key size negotiation</td>
<td>SM/MAS/EKS/BV-01-C, SM/MAS/EKS/BI-01-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/1 AND (NOT SM 2/1)</td>
<td>Initiate Just Works pairing with no MITM</td>
<td>SM/MAS/JW/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/1</td>
<td>Initiate Just Works pairing</td>
<td>SM/MAS/JW/BL-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SM/MAS/JW/BV-05-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SM/MAS/JW/BL-04-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/2</td>
<td>Initiate Passkey Entry pairing</td>
<td>SM/MAS/PKE/BL-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SM/MAS/PKE/BL-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SM/MAS/PKE/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/2 AND SM 4/1</td>
<td>Initiate pairing – Unauthenticated key</td>
<td>SM/MAS/PKE/BV-04-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/3</td>
<td>Initiate OOB pairing</td>
<td>SM/MAS/OOB/BL-01-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SM/MAS/OOB/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/3 AND SM 4/2</td>
<td>Initiate pairing, only IUT has OOB data</td>
<td>SM/MAS/OOB/BV-03-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/2 AND (NOT SM 4/3)</td>
<td>Initiate pairing, only Lower Tester has OOB data</td>
<td>SM/MAS/OOB/BV-05-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/1 AND (NOT SM 4/3)</td>
<td>Initiate pairing, only Lower Tester has OOB data</td>
<td>SM/MAS/OOB/BV-07-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 4/3 AND SM 4/1</td>
<td>Initiate pairing, only IUT has OOB data</td>
<td>SM/MAS/OOB/BV-09-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 5/4</td>
<td>Slave Initiated Security – Master response</td>
<td>SM/MAS/SIP/BV-02-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/1</td>
<td>Just Works, IUT Initiator, Secure Connections – success</td>
<td>SM/MAS/SCJW/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/1</td>
<td>Just Works, IUT Initiator, Secure Connections – Handle AuthReq flag RFU correctly</td>
<td>SM/MAS/SCJW/BV-04-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/1</td>
<td>Just Works, IUT Initiator, Secure Connections – Pairing Failed</td>
<td>SM/MAS/SCJW/BL-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/2</td>
<td>Passkey Entry, IUT Initiator, Secure Connections – success</td>
<td>SM/MAS/SCPK/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/2</td>
<td>Passkey Entry, IUT Initiator, Secure Connections – Handle AuthReq flag RFU correctly</td>
<td>SM/MAS/SCPK/BV-04-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/2</td>
<td>Passkey Entry, IUT Initiator, Secure Connections – Pairing Failed</td>
<td>SM/MAS/SCPK/BL-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/2</td>
<td>Passkey Entry, IUT Initiator, Secure Connections – failure</td>
<td>SM/MAS/SCPK/BL-02-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/3</td>
<td>Out of Band, IUT Initiator, Secure Connections – success</td>
<td>SM/MAS/SCOB/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/3</td>
<td>Out of Band, IUT Initiator, Secure Connections – Pairing Failed</td>
<td>SM/MAS/SCOB/BI-04-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/3</td>
<td>Out of Band, IUT Initiator, Secure Connections – Handle AuthReq flag RFU correctly</td>
<td>SM/MAS/SCOB/BV-04-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5 AND SM 4/3</td>
<td>Out of Band, IUT Initiator, Secure Connections – failure</td>
<td>SM/MAS/SCOB/BI-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND SM 2/5</td>
<td>Master Key Distribution - LE Secure Connections - Invalid Public Key</td>
<td>SM/MAS/KDU/BI-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND GAP 35/9 AND GAP 41/2b AND (NOT SM 5/5)</td>
<td>Cross Transport Key Derivation, IUT Initiator, Secure Connections</td>
<td>SM/MAS/SCCT/BV-01-C</td>
</tr>
<tr>
<td>SM 1/1 AND GAP 35/9 AND GAP 41/2b AND SM 5/5</td>
<td>Cross Transport Key Derivation, IUT Initiator, Secure Connections, Link Key Conversion Function h7, Derive LE LTK from BR/EDR</td>
<td>SM/MAS/SCCT/BV-03-C</td>
</tr>
<tr>
<td>SM 1/1 AND GAP 35/9 AND GAP 41/2a AND SM 5/5</td>
<td>Cross Transport Key Derivation, IUT Initiator, Secure Connections, Link Key Conversion Function h7, Derivation of BR/EDR Link Key from LE LTK</td>
<td>SM/MAS/SCCT/BV-07-C</td>
</tr>
<tr>
<td>SM 1/1 AND GAP 35/9 AND GAP 41/2a AND SM 5/5</td>
<td>Cross Transport Key Derivation, IUT Initiator, Secure Connections, Link Key Conversion Function h7, Derivation of BR/EDR Link Key from LE LTK</td>
<td>SM/MAS/SCCT/BV-09-C</td>
</tr>
</tbody>
</table>

### Slave Role

<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Test Case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM 1/2</td>
<td>Responder tests</td>
<td>SM/SLA/PROT/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 5/3</td>
<td>Slave Initiated Security</td>
<td>SM/SLA/SIE/BV-01-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 5/3</td>
<td></td>
<td>SM/SLA/SIP/BV-01-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 7/1</td>
<td>Slave Key Distribution - Encryption Key bit</td>
<td>SM/SLA/KDU/BV-01-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 7/2</td>
<td>Slave Key Distribution - Identity Key bit</td>
<td>SM/SLA/KDU/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 7/3</td>
<td>Slave Key Distribution - Signing Key bit</td>
<td>SM/SLA/KDU/BV-03-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 3/1</td>
<td>Encryption Key size negotiation – Respond</td>
<td>SM/SLA/EKS/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 3/1</td>
<td></td>
<td>SM/SLA/EKS/BI-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/1</td>
<td>Respond to Just Works pairing</td>
<td>SM/SLA/JW/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/1</td>
<td></td>
<td>SM/SLA/JW/BI-03-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/1 AND SM 2/2</td>
<td>Respond to Just Works pairing with Unauthenticated no MITM protection</td>
<td>SM/SLA/JW/BI-02-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/2</td>
<td>Respond to Pass key Entry pairing</td>
<td>SM/SLA/PKE/BI-03-C SM/SLA/PKE/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/2 AND SM 4/1</td>
<td>Respond to pairing – Unauthenticated key</td>
<td>SM/SLA/PKE/BV-05-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/3</td>
<td>Respond to OOB pairing – both sides have OOB data</td>
<td>SM/SLA/OOB/BI-02-C SM/SLA/OOB/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/3 AND SM 4/2</td>
<td>Respond to pairing – IUT has OOB data</td>
<td>SM/SLA/OOB/BV-04-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 5/2</td>
<td>Re-encrypt an encrypted link with LTK</td>
<td>SM/SLA/KDU/BV-07-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 7/2 AND SM 2/5</td>
<td>Slave Key Distribution - Identity Key bit, LE Secure Connections</td>
<td>SM/SLA/KDU/BV-08-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 7/3 AND SM 2/5</td>
<td>Slave Key Distribution - Signing Key bit, LE Secure Connections</td>
<td>SM/SLA/KDU/BV-09-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/2 AND (NOT SM 4/3)</td>
<td>Respond to OOB pairing where IUT lacks OOB data</td>
<td>SM/SLA/OOB/BV-06-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/1 AND (NOT SM 4/3)</td>
<td>Respond to OOB pairing where IUT lacks OOB data</td>
<td>SM/SLA/OOB/BV-08-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 4/3 AND SM 4/1</td>
<td>Respond to pairing – IUT has OOB data</td>
<td>SM/SLA/OOB/BV-10-C</td>
</tr>
<tr>
<td>SM 1/2 and SM 2/5 and SM 4/1</td>
<td>Just Works, IUT Responder, Secure Connections – success</td>
<td>SM/SLA/SCJW/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 4/1</td>
<td>Just Works, IUT Responder, Secure Connections – Handle AuthReq flag RFU correctly</td>
<td>SM/SLA/SCJW/BV-03-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 4/1</td>
<td>Just Works, IUT Responder, Secure Connections – Confirm Check Failure</td>
<td>SM/SLA/SCJW/BI-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 4/2</td>
<td>Passkey Entry, IUT Responder, Secure Connections – success</td>
<td>SM/SLA/SCPK/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 4/2</td>
<td>Passkey Entry, IUT Responder, Secure Connections – Handle AuthReq flag RFU correctly</td>
<td>SM/SLA/SCPK/BV-03-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 4/3</td>
<td>Out of Band, IUT Responder, Secure Connections – success</td>
<td>SM/SLA/SCOB/BV-02-C</td>
</tr>
<tr>
<td>Item</td>
<td>Feature</td>
<td>Test Case(s)</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 4/3</td>
<td>Out of Band, IUT Responder, Secure Connections – Handle AuthReq flag RFU correctly</td>
<td>SM/SLA/SCOB/BV-03-C</td>
</tr>
<tr>
<td>SM1/2 AND SM 2/5</td>
<td>Slave Key Distribution - LE Secure Connections - Invalid Public Key</td>
<td>SM/SLA/KDU/BI-01-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND (NOT SM 5/5) AND GAP 43/2b</td>
<td>Cross Transport Key Derivation, IUT Responder, Secure Connections</td>
<td>SM/SLA/SCCT/BV-02-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 5/5 AND GAP 43/2b</td>
<td>Cross Transport Key Derivation, IUT Responder, Secure Connections, Link Key Conversion Function h7, Derivation of LE LTK from BR/EDR Link Key</td>
<td>SM/SLA/SCCT/BV-04-C, SM/SLA/SCCT/BV-06-C</td>
</tr>
<tr>
<td>SM 1/2 AND SM 2/5 AND SM 5/5 AND GAP 43/2a</td>
<td>Cross Transport Key Derivation, IUT Responder, Secure Connections, Link Key Conversion Function h7, Derivation of BR/EDR Link Key from LE LTK</td>
<td>SM/SLA/SCCT/BV-08-C, SM/SLA/SCCT/BV-10-C</td>
</tr>
</tbody>
</table>

*Table 5.1: Test Case Mapping*
## 6 Revision History and Contributors

### Revision History

<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.0.0d18–d24</td>
<td>2010-06-03 – 2010-06-24</td>
<td>Step 4 – 7 in TP/JW/BV-04-C (slave initiation removed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Second Pass Verdict in TP/JW/BV-04-C removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Corrected Mconfirm / Sconfirm confusion in SM/MAS/JW/BI-01-C and SM/SLA/JW/BI-02-C (legacy test case IDs TP/JW/BI-01-C and TP/JW/BI-02-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SM/SLA/OOB/BI-02-C (legacy test case ID TP/OOB/BI-02-C) changed from IUT disconnects link to notifies the Upper Tester</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>References to LL transitions in verdicts universally removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/SIGN/BV-01-C split into two test cases:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SM/MAS/SIGN/BV-01-C (legacy test case ID (new)TP/SIGN/BV-01-C) and TP/SIGN/BV-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Old TP/SIGN/BV-02-C split into two test cases:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SM/MAS/SIGN/BV-03-C (legacy test case ID TP/SIGN/BV-03-C) and TP/SIGN/BV-04-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TP/SIGN/BI-01-C split into two test cases:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SM/MAS/SIGN/BI-01-C (legacy test case ID (new)TP/SIGN/BI-01-C) and TP/SIGN/BI-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Outdated MSC in SM/SLA/SIE/BV-01-C (legacy test case ID TP/SIE/BV-01-C) removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SM/SLA/KDU/BV-07-C (legacy test case ID TP/KDU/BV-07-C), Step 1 removed since it was redundant</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SM/MAS/JW/BV-05-C (legacy test case ID TP/JW/BV-05-C), Change MITM(IUT) from '0' to '1' and MITM(Lower Tester) '1' to '0'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SM/MAS/KDU/BV-04-C (legacy test case ID TP/KDU/BV-04-C), Pass Verdict corrected Identity Information Command&quot; changed to &quot;Sign Information Command</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Removed TP/JW/BV-03-C due to redundancy with SM/MAS/JW/BV-01-C (legacy test case ID TP/JW/BV-01-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Removed TP/JW/BV-04-C due to redundancy with SM/SLA/JW/BV-02-C (legacy test case ID TP/JW/BV-02-C)</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added new SM/MAS/SIP/BV-02-C (legacy test case ID TP/SIP/BV-02-C) (Slave Initiates pairing – Master Response)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TCMT update in response to finding when integrating to the TPG</td>
</tr>
<tr>
<td>4.0.1r0</td>
<td></td>
<td>2010-09-23</td>
<td>TSE 3674: Superseded by TSE 3957: Edits to TP step 2A for SM/MAS/JW/BV-05-C (legacy test case ID TP/JW/BV-05-C).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 3940: TP/SIGN/BV-01-C, TP/SIGN/BI-01-C, TP/SIGN/BV-03-C—Delete test cases</td>
</tr>
<tr>
<td>4.0.1r1</td>
<td></td>
<td>2011-01-13</td>
<td>TSE 4189 Correct spelling errors TP/JW/BV-02-C, TP/JW/BI-01-C, TP/JW/BI-02-C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4190 Remove SM/SLA/EKS/BV-02-C (legacy test case ID TP/EKS/BV-02-C) from TCMT</td>
</tr>
<tr>
<td>4.0.1r2</td>
<td></td>
<td>2011-02-07</td>
<td>Input reviewer’s comments on page 25, 29, 31, 43. Fix numbering for SM/SLA/EKS/BI-02-C , SM/MAS/SIP/BV-02-C (legacy test case IDs TP/EKS/BI-02-C, TP/SIP/BV-02-C)</td>
</tr>
<tr>
<td>4.0.1r3</td>
<td></td>
<td>2011-06-27</td>
<td>TSE 4424: SM/SLA/EKS/BV-02-C (legacy test case ID TP/EKS/BV-02-C); TSE 4190 backed out as a result; reorganized EKS test cases BV, then BI</td>
</tr>
<tr>
<td>1</td>
<td>4.0.1</td>
<td>2011-07-18</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td>4.0.2r0</td>
<td></td>
<td>2011-12-14</td>
<td>TSE 3856: Change to TCMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4576: Repeat of TSE 3940 (Remove test cases TP/SIGN/BV-02-C, TP/SIGN/BV-04-C, TP/SIGN/BI-02-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4312: See TSE 4569.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4328: SM/MAS/KDU/BV-04-C (legacy test case ID TP/KDU/BV-04-C): TCMT change</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4538: SM/MAS/OOB/BV-05-C (legacy test case ID TP/OOB/BV-05-C): Update Test procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4569: SM/MAS/PROT/BV-01-C, SM/SLA/PROT/BV-02-C (legacy test case IDs TP/PROT/BV-01-C, TP/PROT/BV-02-C): Supersedes TSE 4312:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4570: SM/SLA/JW/BV-02-C (legacy test case ID TP/JW/BV-02-C): Update Test procedure</td>
</tr>
<tr>
<td>4.0.2r1</td>
<td>2012-01-19</td>
<td></td>
<td>TSE 3856: SM/MAS/KDU/BV-04-C (legacy test case ID TP/KDU/BV-04-C) needs resolution via comment</td>
</tr>
<tr>
<td>4.0.2r2</td>
<td>2012-02-06</td>
<td></td>
<td>Fixed TCMT with conflict to TSE 3856</td>
</tr>
<tr>
<td>2</td>
<td>4.0.2</td>
<td>2012-03-30</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td></td>
<td>4.0.3r0</td>
<td>2012-05-21</td>
<td>TSE 4754: SM/MAS/SIP/BV-02-C (legacy test case ID TP/SIP/BV-02-C): Add text to line 2 of Test Procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 4668: SM/MAS/OOB/BV-05-C (legacy test case ID TP/OOB/BV-05-C): Fix TCMT</td>
</tr>
<tr>
<td></td>
<td>4.0.3r1</td>
<td>2012-05-22</td>
<td>TSE 4754, additional correction in pass verdict.</td>
</tr>
<tr>
<td>3</td>
<td>4.0.3</td>
<td>2012-07-31</td>
<td>Prepare for publication.</td>
</tr>
<tr>
<td></td>
<td>4.0.4r0</td>
<td>2012-09-04</td>
<td>TSE 4867: Changed Master Key Distribution – Encryption Key bit, and Signing Key bit, test cases were inverted.</td>
</tr>
<tr>
<td>4</td>
<td>4.0.4</td>
<td>2012-11-12</td>
<td>Prepare for Publication</td>
</tr>
<tr>
<td></td>
<td>4.0.5r1</td>
<td>2013-05-31</td>
<td>TSE 4585/4590/4984: Updated Just Works section test cases: SM/MAS/JW/BV-01-C, SM/SLA/JW/BV-02-C,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated Passkey Entry (PKE) section test cases: SM/MAS/PKE/BV-01-C, SM/SLA/PKE/BV-02-C,</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Master Role Section</td>
<td></td>
<td></td>
<td>Updated SM/MAS/JW/BV-01-C (legacy test case ID TP/JW/BV-01-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/MAS/PKE/BV-04-C (legacy test case ID TP/PKE/BV-04-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Removed SM/MAS/OOB/BV-03-C from SM/MAS/OOB/BI-01-C and SM/MAS/OOB/BV-01-C (legacy test case IDs TP/OOB/BV-03-C from TP/OOB/BI-01-C and TP/OOB/BV-01-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/MAS/OOB/BV-03-C (legacy test case ID TP/OOB/BV-03-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/MAS/OOB/BV-07-C (legacy test case ID TP/OOB/BV-07-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/MAS/OOB/BV-09-C (legacy test case ID TP/OOB/BV-09-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added TP/JW/BV-06-C row.</td>
</tr>
<tr>
<td>Slave Role Section</td>
<td></td>
<td></td>
<td>Updated mapping for SM/SLA/JW/BI-02-C (legacy test case ID TP/JW/BI-02-C)</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Removed TP/PKE/BV-03-C from SM/SLA/PKE/BI-03-C and SM/SLA/PKE/BV-02-C (legacy test case IDs TP/PKE/BI-03-C and TP/PKE/BV-02-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/SLA/PKE/BV-05-C (legacy test case ID TP/PKE/BV-05-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Removed SM/SLA/OOB/BV-04-C (legacy test case ID TP/OOB/BV-04-C) from SM/SLA/OOB/BI-02-C and SM/SLA/OOB/BV-02-C (legacy test case IDs TP/OOB/BI-02-C and TP/OOB/BV-02-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/SLA/OOB/BV-04-C (legacy test case ID TP/OOB/BV-04-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updated SM/SLA/OOB/BV-06-C (legacy test case ID TP/OOB/BV-06-C) mapping</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/SLA/OOB/BV-08-C (legacy test case ID TP/OOB/BV-08-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added SM/SLA/OOB/BV-10-C (legacy test case ID TP/OOB/BV-10-C) row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added TP/JW/BV-03-C row.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 5065: Updated reference, test procedure, and pass verdict for SM/SLA/EKS/BI-02-C (legacy test case ID TP/EKS/BI-02-C).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0.5r2</td>
<td>2013-06-05</td>
<td></td>
<td>BTI Review, Alicia’s Comments</td>
</tr>
<tr>
<td>5</td>
<td>4.0.5</td>
<td>2013-07-02</td>
<td>Prepare for Publication</td>
</tr>
<tr>
<td></td>
<td>4.0.6.rT</td>
<td>2013-10-07</td>
<td>Template Conversion</td>
</tr>
<tr>
<td>4.1.0r01</td>
<td>2013-10-07</td>
<td>2013-10-07</td>
<td>TSE 5260: Update to Test Procedure and Expected Outcome, for SM/MAS/EKS/BI-01-C (legacy test case ID TP/EKS/BI-01-C).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 5304: Renamed TP/JW/BV-03-C to SM/SLA/JW/BI-03-C (legacy test case ID TP/JW/BI-03-C). Renamed TP/JW/BV-06-C to SM/MAS/JW/BI-04-C (legacy test case ID TP/JW/BI-04-C), and updated 1c and 2c in the Test Procedure and updated TCMT with TC name changes.</td>
</tr>
<tr>
<td>6</td>
<td>4.1.0</td>
<td>2013-12-03</td>
<td>Prepare for Publication</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 4.1.1r00          |                  | 2014-01-23 | TSE 5438: Updated TCMT mapping for SM/MAS/KDU/BV-06-C (legacy test case ID TP/KDU/BV-06-C).  
| 4.1.1r01          |                  | 2014-04-08 | TSE 5402: Updated TC Description, Initial Condition, Test Procedure and Notes for SM/MAS/EKS/BV-01-C and SM/SLA/EKS/BV-02-C (legacy test case IDs TP/EKS/BV-01-C and TP/EKS/BV-02-C).  
Added SM IXIT as [7] in Reference section. |
| 4.1.1r02          |                  | 2014-06-12 | BTI Review, Alicia: Revised all instances of Tester to Lower Tester as appropriate.                                                      |
| 7                 | 4.1.1            | 2014-07-07 | TCRL 2014-1 Publication                                                                                                                                                                          |
| 4.2.0r00          |                  | 2014-11-17 | Integrated changes from Sections 1.4 – 1.5 of Core_Enhanced_Privacy_1_2.TS.CR.R05 and Section 8 of Core_LE_Secure_Connections.TS.CR.R16          |
| 4.2.0r01          |                  | 2014-11-24 | Additional minor editorial changes                                                                                                     |
| 4.2.0r02          |                  | 2014-11-25 | Review by Rajesh, added contributors from the LE Secure Connections CR and added missing IO capabilities in step 2 of SM/SLA/SCPK/BI-03-C (legacy test case ID TP/SCPK/BI-03-C). |
| 4.2.0r03          |                  | 2014-11-25 | Review by Miles; moved SM/SLA/SCJW/BI-02-C (legacy test case ID TP/SCJW/BI-02-C) to responder.                                           |
| 4.2.0r04          |                  | 2014-11-25 | Review by Alicia.  
TCMT: Moved SM/MAS/SCPK/BI-02-C (legacy test case ID TP/SCPK/BI-02-C) to Initiator, updated items for SM/MAS/SCCT/BV-01-C (legacy test case ID TP/SCCT/BV-01-C). |
<table>
<thead>
<tr>
<th>Publication Number</th>
<th>Revision History</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4.2.0</td>
<td>2014-12-05</td>
<td>Prepared for TCRL 2014-2 publication</td>
</tr>
<tr>
<td></td>
<td>4.2.1r00</td>
<td>2015-05-06</td>
<td>TSE 6267: Corrected numbering error in SM/SLA/SCPK/BI-03-C (legacy test case ID TP/SCPK/BI-03-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6337: Corrected steps 1 and 2 in SM/MAS/SCPK/BV-04-C (legacy test case ID TP/SCPK/BV-04-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6342: Corrected steps 1 and 2 in SM/SLA/SCPK/BV-02-C (legacy test case ID TP/SCPK/BV-02-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6346: Corrected Pass verdict in SM/MAS/SCOB/BI-01-C (legacy test case ID TP/SCOB/BI-01-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6351: Corrected roles in Initial Condition for SM/SLA/SCJW/BV-02-C (legacy test case ID TP/SCJW/BV-02-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6366: Corrected Initial Condition and Test Procedure for SM/MAS/SCCT/BV-01-C and SM/SLA/SCCT/BV-02-C (legacy test case IDs TP/SCCT/BV-01-C and TP/SCCT/BV-02-C)</td>
</tr>
<tr>
<td>9</td>
<td>4.2.1</td>
<td>2015-07-14</td>
<td>Prepared for TCRL 2015-1 publication</td>
</tr>
<tr>
<td></td>
<td>4.2.2r00</td>
<td>2015-10-14</td>
<td>TSE 6707: Updated SM/MAS/OOB/BI-01-C and SM/SLA/OOB/BI-02-C (legacy test case IDs TP/OOB/BI-01-C and TP/OOB/BI-02-C) test titles for clarity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6614: Corrected test case mapping for SM/SLA/SIE/BV-01-C and SM/SLA/SIP/BV-01-C (legacy test case IDs TP/SIE/BV-01-C &amp; TP/SIP/BV-01-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6602: deleted tests TP/KDU/BV-08-C and TP/KDU/BV-09-C.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6580: Clarified test description for SM/SLA/PKE/BV-05-C (legacy test case ID TP/PKE/BV-05-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 6463: Clarified requirements regarding Secure Connections and Keypress bits in SM/MAS/JW/BI-04-C (legacy test case ID TP/JW/BI-04-C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TSE 5744: Revised bonding flag value requirement in step 2c of SM/MAS/PKE/BV-04-C (legacy test case ID TP/PKE/BV-04-C).</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5.0.2r00</td>
<td>2017-08-23</td>
<td></td>
<td>TSE 9698: For TCMT test cases SM/SLA/SCCT/BV-02-C, 04-C, 06-C, 08-C, and 10-C, changed 25/9 to SM 2/5 and moved GAP items to the end of the logical statements. TSE 9389: For test cases SM/SLA/KDU/BV-01-C - ...03-C and SM/MAS/KDU/BV-04-C - ...06-C: added &quot;LE Legacy Pairing&quot; to titles and updated test procedure steps. Added new test cases SM/SLA/KDU/BV-08-C - ...09-C and SM/MAS/KDU/BV-10-C - ...11-C. In the TCMT, revised SM/MAS/KDU/BV-06-C and added new test cases SM/MAS/KDU/BV-10-C - ...11-C and SM/SLA/KDU/BV-08-C - ...09-C.</td>
</tr>
<tr>
<td>14</td>
<td>5.0.2</td>
<td>2017-12-07</td>
<td>Approved by BTI. Prepared for TCRL 2017-2 publication.</td>
</tr>
<tr>
<td>5.0.3r00</td>
<td>2018-06-14</td>
<td></td>
<td>Incorporated E10734 Pairing Updates Test CR: Added new test cases SM/SLA/KDU/BI-01-C and SM/MAS/KDU/BI-01-C. Added SM 1/1 AND SM 2/5 AND SUM 34/13 to TCMT.</td>
</tr>
<tr>
<td>15</td>
<td>5.0.3</td>
<td>2018-07-02</td>
<td>Approved by BTI. Prepared for TCRL 2018-1 publication.</td>
</tr>
<tr>
<td>5.1.0</td>
<td>2018-11-13</td>
<td></td>
<td>Updated revision number from 5.0.4 to 5.1.0 to align with the adoption of Core Specification version 5.1</td>
</tr>
<tr>
<td>16</td>
<td>5.1.0</td>
<td>2018-12-07</td>
<td>Approved by BTI. Prepared for TCRL 2018-2 publication.</td>
</tr>
<tr>
<td>5.1.1r00–r02</td>
<td>2019-03-27–2019-06-25</td>
<td></td>
<td>TSE 11721 (rating 1): Replaced MSCs for test cases SM/SLA/KDU/BI-01-C and SM/MAS/KDU/BI-01-C with the revised Visio diagrams included in the CR.</td>
</tr>
<tr>
<td>17</td>
<td>5.1.1</td>
<td>2019-08-01</td>
<td>Approved by BTI. Prepared for TCRL 2019-1 publication.</td>
</tr>
<tr>
<td>p18r00–r01</td>
<td>2019-10-08 – 2019-11-22</td>
<td></td>
<td>TSE 12787 (rating 2): Updated TCMT to remove references to SUM ICS 34/13.</td>
</tr>
<tr>
<td>Publication Number</td>
<td>Revision History</td>
<td>Date</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

**Contributors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike Tsai</td>
<td>Atheros</td>
</tr>
<tr>
<td>Alicia Courtney</td>
<td>Broadcom</td>
</tr>
<tr>
<td>Angel Polo</td>
<td>Broadcom</td>
</tr>
<tr>
<td>Mayank Batra</td>
<td>CSR</td>
</tr>
<tr>
<td>Joe Decuir</td>
<td>CSR</td>
</tr>
<tr>
<td>Rajesh Garai</td>
<td>CSR</td>
</tr>
<tr>
<td>Giriraj Goyal</td>
<td>CSR</td>
</tr>
<tr>
<td>Robin Heydon</td>
<td>CSR</td>
</tr>
<tr>
<td>Magnus Sommansson</td>
<td>CSR</td>
</tr>
<tr>
<td>Patrick Reinelt</td>
<td>FTE</td>
</tr>
<tr>
<td>Harish Balasubramaniam</td>
<td>Intel</td>
</tr>
<tr>
<td>Marcel Holtmann</td>
<td>Intel</td>
</tr>
<tr>
<td>Yao Wang</td>
<td>IVT Corporation</td>
</tr>
<tr>
<td>David Engelien-Lopes</td>
<td>Nordic</td>
</tr>
<tr>
<td>David Lopes</td>
<td>Nordic Semiconductor</td>
</tr>
<tr>
<td>Miles Smith</td>
<td>Nordic Semiconductor ASA</td>
</tr>
<tr>
<td>Joel Linsky</td>
<td>Qualcomm Atheros</td>
</tr>
<tr>
<td>Rasmus Abildgren</td>
<td>Samsung Electronics Co., Ltd</td>
</tr>
<tr>
<td>Jason Hillyard</td>
<td>Wicentric</td>
</tr>
</tbody>
</table>