Bluetooth® low energy Security
Bluetooth Smart (low energy) Security

The Bluetooth security model includes five distinct security features:

• Pairing: The process for creating one or more shared secret keys.
• Bonding: The act of storing the keys created during pairing for use in subsequent connections in order to form a trusted device pair.
• Device Authentication: Verification that the two devices have the same keys.
• Encryption: Provides message confidentiality.
• Message Integrity: Protects against message forgeries.
Bluetooth Smart (Low energy) Security

Association Models (AKA Pairing)
Bluetooth Smart (LE) uses 4 pairing methods:

- Just Works
- Out of Band
- Passkey Entry
- Numeric Comparison (Only Secure Connections)

Method determines computation of security keys:

- Legacy Encryption – Short Temporary Key (STK)
- Secure Connections - Long Term Key (LTK).
Security Modes and Levels

LE Security Mode 1
• Level 1 – No security. No authentication and no encryption.
• Level 2 – Unauthenticated pairing with encryption.
• Level 3 – Authenticated pairing with encryption.
• Level 4 – Authenticated Secure Connections pairing with encryption.

LE Security Mode 2
• Level 1 – Unauthenticated pairing with data signing.
• Level 2 – Authenticated pairing with data signing.
  (Mode 2 is only used for connection-based data signing.)

Authenticated Pairing
  Pairing is performed with MITM protection.

Unauthenticated Pairing
  Pairing is performed without MITM protection.
Encryption and Authentication

Encryption

The encryption in Bluetooth LE is based on 128-bit Advanced Encryption Standard – Counter with CBC-MAC (AES-CCM). LTK is used with this algorithm to create the 128-bit “shared secret” key.

Authentication

Authentication is provided in Bluetooth (LE) by digitally signing the data using the Connection Signature Resolving Key (CSRK). The sending device places a signature after the Data PDU. The receiver verifies the signature using the CSRK.
Initiation of pairing (Phase I) is the same for all LE connections. The differences come in Phase II (authentication). The I/O Capabilities, Out of Band flag, and Authorization requirements determine the method used for authentication in Phase II.
Pairing Request

- **NoInputNoOutput**
- **DisplayOnly**
- **KeyboardOnly**
- **DisplayYesNo**
- **KeyboardDisplay**

Determines key generation
Pairing Response

- Access Address: 0x50654d2e
- CRC: 0xabee62

LE DATA:
- LLID: Start
- NESN: 1
- SN: 0
- MD: 1
- Payload Length: 11

L2CAP:
- PDU Length: 7

SMP:
- Code: Pairing Response
- IO Capabilities: DisplayOnly
- OOB data flag: OOB Authentication data not present

AuthReq
- Bonding Flags: Bonding
- MITM: MITM Protection: No
- Maximum Encryption Key Size: 16 Octets

Initiator Key Distribution
- EncKey: Initiator shall distribute LTK followed by EDIV and Rand
- IdKey: Initiator shall distribute IRK followed by its address
- Sign: Initiator shall distribute CSRK

Responder Key Distribution
- EncKey: Responder shall distribute LTK followed by EDIV and Rand
- IdKey: Responder shall distribute IRK followed by its address
- Sign: Responder shall distribute CSRK
Secure Connections Pairing Request
(Just Works with Debug Keys)

Flags added in Version 4.2 of the spec with the addition of LE Secure Connections

Pairing Request

<table>
<thead>
<tr>
<th>Code: Pairing Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO Capabilities: NoInputNoOutput</td>
</tr>
<tr>
<td>OOB data flag: OOB Authentication data not present</td>
</tr>
</tbody>
</table>

AuthReq

<table>
<thead>
<tr>
<th>Bonding Flags: No Bonding</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITM: MITM Protection: No</td>
</tr>
<tr>
<td>Secure Connection Pairing: Yes</td>
</tr>
<tr>
<td>Keypress Notifications: No</td>
</tr>
<tr>
<td>Maximum Encryption Key Size: 16 Octets</td>
</tr>
<tr>
<td>Key Distribution: No keys shall be distributed</td>
</tr>
<tr>
<td>Key Distribution: No keys shall be distributed</td>
</tr>
</tbody>
</table>

Pairing Response

<table>
<thead>
<tr>
<th>Code: Pairing Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO Capabilities: NoInputNoOutput</td>
</tr>
<tr>
<td>OOB data flag: OOB Authentication data not present</td>
</tr>
</tbody>
</table>

AuthReq

<table>
<thead>
<tr>
<th>Bonding Flags: No Bonding</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITM: MITM Protection: No</td>
</tr>
<tr>
<td>Secure Connection Pairing: Yes</td>
</tr>
<tr>
<td>Keypress Notifications: No</td>
</tr>
<tr>
<td>Maximum Encryption Key Size: 16 Octets</td>
</tr>
<tr>
<td>Key Distribution: No keys shall be distributed</td>
</tr>
<tr>
<td>Key Distribution: No keys shall be distributed</td>
</tr>
</tbody>
</table>
## LE Legacy OOB and MITM flag rules

### Legacy Rules

<table>
<thead>
<tr>
<th>Responder</th>
<th>Initiator</th>
<th>OOB Set</th>
<th>OOB Not Set</th>
<th>MITM Set</th>
<th>MITM Not Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOB Set</td>
<td>Use OOB</td>
<td>Check MITM</td>
<td>Check MITM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OOB Not Set</td>
<td>Check MITM</td>
<td></td>
<td>Check MITM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MITM Set</td>
<td>Use IO Capabilities</td>
<td></td>
<td>Use IO Capabilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MITM Not Set</td>
<td>Use IO Capabilities</td>
<td></td>
<td>Use Just Works</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Secure Connection Rules

<table>
<thead>
<tr>
<th>Responder</th>
<th>Initiator</th>
<th>OOB Set</th>
<th>OOB Not Set</th>
<th>MITM Set</th>
<th>MITM Not Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOB Set</td>
<td>Use OOB</td>
<td>Use OOB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OOB Not Set</td>
<td>Use OOB</td>
<td>Use IO Capabilities</td>
<td>Check MITM</td>
<td>Use IO Capabilities</td>
<td></td>
</tr>
<tr>
<td>MITM Set</td>
<td>Use IO Capabilities</td>
<td></td>
<td>Use IO Capabilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MITM Not Set</td>
<td>Use IO Capabilities</td>
<td></td>
<td>Use Just Works</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Mapping I/O capabilities to Key Generation Method

<table>
<thead>
<tr>
<th>Responder</th>
<th>Display Only</th>
<th>Display YesNo</th>
<th>Keyboard Only</th>
<th>NoInput NoOutput</th>
<th>Keyboard Display</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display Only</strong></td>
<td>Just Works Unauthenticated</td>
<td>Just Works Unauthenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
</tr>
<tr>
<td><strong>Display YesNo</strong></td>
<td>Just Works Unauthenticated</td>
<td>Just Works Unauthenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
</tr>
<tr>
<td><strong>Keyboard Only</strong></td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
</tr>
<tr>
<td><strong>NoInput NoOutput</strong></td>
<td>Just Works Unauthenticated</td>
<td>Just Works Unauthenticated</td>
<td>Passkey Entry (For LE Legacy Pairing): responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry (For LE Legacy Pairing): responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
</tr>
<tr>
<td><strong>Keyboard Display</strong></td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Passkey Entry: responder displays, initiator inputs Authenticated</td>
<td>Numeric Comparison (For LE Secure Connections) Authenticated</td>
<td>Numeric Comparison (For LE Secure Connections) Authenticated</td>
<td>Numeric Comparison (For LE Secure Connections) Authenticated</td>
</tr>
</tbody>
</table>

## Key Generation Method

- **Passkey Entry:** This method involves entering a passkey, which is a unique code used to authenticate the device. This can be done through a display, keyboard, or other input methods, depending on the device capabilities.
- **Responder:** The responder is the device that receives the passkey entry and verifies the identity of the initiator.
- **Display Only:** This method is used when only a display is available for input.
- **Display YesNo:** This method is used when a display and a yes/no input method are available.
- **Keyboard Only:** This method is used when only a keyboard is available for input.
- **NoInput NoOutput:** This method is used when there is no input or output available.
- **Keyboard Display:** This method is used when both a keyboard and a display are available.

(Source: Frontline)
LE Pairing with Legacy Encryption
Computation of Temporary Key (TK)  
(Legacy pairing)

Computation of TK is based on the pairing method.

• Just Works – TK is set to all zeros.

• Out of Band – Devices use a different technology, such as NFC or tethering, to pass the TK between the devices. If the OOB technology is not secure the TK can be discovered.

• Passkey Entry – User inputs 6 digit passkey. The value is used as TK. For instance:
  
  (Passkey of: “999999” becomes TK: 0x000000000000000000000000000F423F)

TK is used to calculate the Short-term Key (STK) which is used to initially encrypt the connection for the transmission of the security keys which will be used for the transmission of data across the connection.
Security keys used in Legacy encryption

Legacy LE encryption can exchange/generate multiple keys, each for a specific purpose:

- **Temporary Key (TK)**
  Used to generate the Short-Term Key (STK)

- **Short Term Key**
  Used to initially encrypt a connection in order to exchange additional keys

- **Long-Term Key (LTK)**
  Confidentiality of data (AES encryption) and device authentication.

- **Connection Signature Resolving Key (CSRK)**
  Authentication of unencrypted data (digital signing)

- **Identity Resolving Key (IRK)**
  Device Identity (random address resolution)
STK Generation – Just Works

STK is calculated by running $S_1$ with the TK value as the key input and initiator and responder random numbers.

$$STK = s_1(TK, Strand, Mrand)$$

Link is then encrypted using STK
Just Works Pairing

Example of Just Works pairing request/response

Pairing Request

- SMP:
  - Code: Pairing Request
    - IO Capabilities: KeyboardDisplay
      - OOB data flag: OOB Authentication data not present
    - AuthReq
      - Bonding Flags: Bonding
      - MITM: MITM Protection: Yes
        - Secure Connection Pairing: No
        - Keypress Notifications: No
        - Maximum Encryption Key Size: 16 Octets
    - Initiator Key Distribution
      - EncKey: Initiator shall distribute LTK followed by EDIV and Rand
      - IdKey: Initiator shall distribute IRK followed by its address
      - Sign: Initiator shall distribute CSRK
    - Responder Key Distribution
      - EncKey: Responder shall distribute LTK followed by EDIV and Rand
      - IdKey: Responder shall distribute IRK followed by its address
      - Sign: Responder shall distribute CSRK

Pairing Response

- SMP:
  - Code: Pairing Response
    - IO Capabilities: NoInputNoOutput
      - OOB data flag: OOB Authentication data not present
    - AuthReq
      - Bonding Flags: Bonding
      - MITM: MITM Protection: No
        - Secure Connection Pairing: No
        - Keypress Notifications: No
        - Maximum Encryption Key Size: 16 Octets
    - Initiator Key Distribution
      - EncKey: Initiator shall distribute LTK followed by EDIV and Rand
      - IdKey: Initiator shall distribute IRK followed by its address
      - Sign: Initiator shall distribute CSRK
    - Responder Key Distribution
      - EncKey: Responder shall distribute LTK followed by EDIV and Rand
      - IdKey: Responder shall distribute IRK followed by its address
      - Sign: Responder shall distribute CSRK

Must use Just Works
### Just Works Pairing and Key Distribution

<table>
<thead>
<tr>
<th>B...</th>
<th>Frame#</th>
<th>Side</th>
<th>Code</th>
<th>Frame Size</th>
<th>Delta</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>835</td>
<td>1</td>
<td>P</td>
<td>Pairing Request</td>
<td>29</td>
<td>00:00:00.060234</td>
<td>4/29/2014 5:43:30.130777...</td>
</tr>
<tr>
<td>839</td>
<td>2</td>
<td>P</td>
<td>Pairing Response</td>
<td>29</td>
<td>00:00:01.319772</td>
<td>4/29/2014 5:43:30.191013...</td>
</tr>
<tr>
<td>962</td>
<td>1</td>
<td>P</td>
<td>Pairing Confirm</td>
<td>39</td>
<td>00:00:00.120233</td>
<td>4/29/2014 5:43:31.510785...</td>
</tr>
<tr>
<td>972</td>
<td>2</td>
<td>P</td>
<td>Pairing Confirm</td>
<td>39</td>
<td>00:00:00.059771</td>
<td>4/29/2014 5:43:31.631014...</td>
</tr>
<tr>
<td>973</td>
<td>1</td>
<td>P</td>
<td>Pairing Random</td>
<td>39</td>
<td>00:00:00.060230</td>
<td>4/29/2014 5:43:31.751012...</td>
</tr>
<tr>
<td>979</td>
<td>2</td>
<td>P</td>
<td>Pairing Random</td>
<td>39</td>
<td>00:00:00.060007</td>
<td>4/29/2014 5:43:32.411020...</td>
</tr>
<tr>
<td>1,023</td>
<td>S</td>
<td>S</td>
<td>Encryption Information</td>
<td>43</td>
<td>00:00:00.060000</td>
<td>4/29/2014 5:43:32.471023...</td>
</tr>
<tr>
<td>1,028</td>
<td>S</td>
<td>S</td>
<td>Master Identification</td>
<td>37</td>
<td>00:00:00.120003</td>
<td>4/29/2014 5:43:32.591022...</td>
</tr>
<tr>
<td>1,034</td>
<td>S</td>
<td>S</td>
<td>Identity Information</td>
<td>43</td>
<td>00:00:00.060001</td>
<td>4/29/2014 5:43:32.651025...</td>
</tr>
<tr>
<td>1,039</td>
<td>S</td>
<td>S</td>
<td>Identity Address Information</td>
<td>34</td>
<td>00:00:00.120001</td>
<td>4/29/2014 5:43:32.771032...</td>
</tr>
<tr>
<td>1,045</td>
<td>S</td>
<td>S</td>
<td>Signing Information</td>
<td>43</td>
<td>00:00:00.059772</td>
<td>4/29/2014 5:43:32.830800...</td>
</tr>
<tr>
<td>1,049</td>
<td>M</td>
<td>M</td>
<td>Encryption Information</td>
<td>43</td>
<td>00:00:00.060062</td>
<td>4/29/2014 5:43:32.891465...</td>
</tr>
<tr>
<td>1,055</td>
<td>M</td>
<td>M</td>
<td>Master Identification</td>
<td>37</td>
<td>00:00:00.059955</td>
<td>4/29/2014 5:43:32.951412...</td>
</tr>
<tr>
<td>1,058</td>
<td>M</td>
<td>M</td>
<td>Identity Information</td>
<td>43</td>
<td>00:00:00.120051</td>
<td>4/29/2014 5:43:33.071471...</td>
</tr>
<tr>
<td>1,069</td>
<td>M</td>
<td>M</td>
<td>Identity Address Information</td>
<td>34</td>
<td>00:00:00.120001</td>
<td>4/29/2014 5:43:33.131402...</td>
</tr>
<tr>
<td>1,072</td>
<td>M</td>
<td>M</td>
<td>Signing Information</td>
<td>43</td>
<td>00:00:00.059931</td>
<td>4/29/2014 5:43:33.131402...</td>
</tr>
</tbody>
</table>
Encryption Information (LTK)

- Header Version: 3
- CP #: 1
- Channel Index: 25 - 2456 MHz
- Meets Predefined Filter Criteria for BT low energy devices: No
- Receive Status: Received without errors
- Decryption Initiated: Yes
- Decryption Status: Decrypted successfully
- Signal Strength: 5 [medium]
- PDU Length: 27

**LE PKT:**
- Preamble: 0xa9
- Access Address: 0x50654d2e
- CRC: 0x88e744

**LE DATA:**
- LLID: Start
- NESN: 1
- SN: 0
- MD: 1
- Payload Length: 25
- MIC Present: Yes
- Encrypted MIC: 0x94cd05ec

**L2CAP:**
- PDU Length: 17
- Channel ID: 0x0006 (LE Security Manager Protocol)

**SMP:**
- Code: Encryption Information
- LTK: 0x201bf82ce563414b16ead9c6a35d7

<table>
<thead>
<tr>
<th>B...</th>
<th>Frame#</th>
<th>Side</th>
<th>Access Address</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2</td>
<td></td>
<td>0x50654d2e</td>
<td>Security Request</td>
</tr>
<tr>
<td>105</td>
<td>1</td>
<td></td>
<td>0x50654d2e</td>
<td>Pairing Request</td>
</tr>
<tr>
<td>110</td>
<td>2</td>
<td></td>
<td>0x50654d2e</td>
<td>Pairing Response</td>
</tr>
<tr>
<td>152</td>
<td>1</td>
<td></td>
<td>0x50654d2e</td>
<td>Pairing Confirm</td>
</tr>
<tr>
<td>155</td>
<td>2</td>
<td></td>
<td>0x50654d2e</td>
<td>Pairing Confirm</td>
</tr>
<tr>
<td>156</td>
<td>1</td>
<td></td>
<td>0x50654d2e</td>
<td>Pairing Random</td>
</tr>
<tr>
<td>159</td>
<td>2</td>
<td></td>
<td>0x50654d2e</td>
<td>Pairing Random</td>
</tr>
<tr>
<td>176</td>
<td>S</td>
<td></td>
<td>0x50654d2e</td>
<td>Encryption Information</td>
</tr>
<tr>
<td>178</td>
<td>S</td>
<td></td>
<td>0x50654d2e</td>
<td>Master Identification</td>
</tr>
<tr>
<td>180</td>
<td>S</td>
<td></td>
<td>0x50654d2e</td>
<td>Identity Information</td>
</tr>
</tbody>
</table>

BINARY:

```
0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1
0 0 0 1 1 0 1 1 1 0 1 0 1 0 1 0 0 0 1 0 1 1 0 0 1 0
0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 1 0
0 0 0 1 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 0 1 1 0 0 1 0
0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

RADIX:

```
9 c a d 6 e b 1 b d 1 4 3 4 5 6 c c 8 2
```
Master Identification (EDIV/Rand)

-- Header Length: 13
-- Header Version: 3
-- CP #: 0
-- Channel Index: 16 - 2438 MHz
-- Meets Predefined Filter Criteria for BT low energy devices: No
-- Receive Status: Received without errors
-- Decryption Initiated: Yes
-- Decryption Status: Decrypted successfully
-- Signal Strength: 3 (weak)
-- PDU Length: 21

LE_PKT:
-- Preamble: 0x05
-- Access Address: 0xf89abce7
-- CRC: 0x31a41

LE_DATA:
-- LLID: Start
-- NESN: 1
-- SN: 0
-- SN+NESN: 2
-- MD: 0
-- FMT: Reserved [0]
-- Payload Length: 19
-- MIC Present: Yes
-- Encrypted MIC: 0x7a68731b

L2CAP:
-- PDU Length: 11
-- Channel ID: 0x0006 (LE Security Manager Protocol)

SMP:
-- Code: Master Identification
-- EDIV: 0x015f
-- Rand: 0x4133f7c499b23d4e
Identity Address Information

LE_PKT:
- Preamble: 0x55
- Access Address: 0x0f9abce7
- CRC: 0x035636

LE_DATA:
- LLID: Start
- NESN: 1
- SN: 0
- SN+NESN: 2
- MD: 0
- FMT: Reserved [0]
- Payload Length: 16
- MIC Present: Yes
- Encrypted MIC: 0x01591022

L2CAP:
- PDU Length: 3
- Channel ID: 0x0006 (LE Security Manager Protocol)
- Code: Identity Address Information
- Address Type: Public device address
- BD_ADDR: 0x7ca15d08303d
Identity Information (IRK)

- Header Version: 3
- CP #: 1
- Channel Index: 25 - 2456 MHz
- Meets Predefined Filter Criteria for BT low energy devices: No
- Receive Status: Received without errors
- Decryption Initiated: Yes
- Decryption Status: Decrypted successfully
- Signal Strength: 5 (medium)
- PDU Length: 27

- LE PKT:
  - Preamble: 0x0aa
  - Access Address: 0x50654d2e
  - CRC: 0x77593c

- LE DATA:
  - LLID: Start
  - NESN: 1
  - SN: 0
  - MD: 0
  - Payload Length: 25
  - MIC Present: Yes
  - Encrypted MIC: 0x7bd3812

- L2CAP:
  - PDU Length: 17
  - Channel ID: 0x0006 (LE Security Manager Protocol)

- SMP:
  - Code: Identity Information
  - IRK: 0x8223a9fdd79fb331d05c05c01d0a972f
Signing Information (CSRK)

- Header Version: 3
- CP #: 1
- Channel Index: 12 - 2430 MHz
- Meets Predefined Filter Criteria for BT low energy devices: No
- Receive Status: Received without errors
- Decryption Initiated: Yes
- Decryption Status: Decrypted successfully
- Signal Strength: 9 (medium)
- PDU Length: 27

LE PKT:
- Preamble: 0xaa
- Access Address: 0x50654d2e
- CRC: 0x515a08

LE DATA:
- LLID: Start
- NESN: 1
- SN: 1
- MD: 0
- Payload Length: 25
- MIC Present: Yes
- Encrypted MIC: 0x76621c14

L2CAP:
- PDU Length: 17
- Channel ID: 0x0006 (LE Security Manager Protocol)

SMP:
- Code: Signing Information
- Signature Key: 0xfa3e9eaf51c7a23008a02972d121c8311
STK Generation – Out of Band

Out of band pairing algorithm selected from parameters of Pairing Request and Pairing Response on Phase 1: Pairing Feature Exchange

Create Mrand
Set TK = User input passkey

Create Srand
Set TK = User input passkey

Mrand = c1 TK, Mrand, Pairing Request command, Pairing Response command, initiating device address, responding device address, type, initiating device address, responding device address, type, responding device address

Srand = c1 TK, Srand, Pairing Request command, Pairing Response command, initiating device address, responding device address, type, initiating device address, responding device address, type, responding device address

PairingConfirm (Mrand)
PairingConfirm (Srand)

PairingRandom (Mrand)
PairingRandom (Srand)

Check for confirm value match

STK is calculated by running S1 with the TK value as the key input and initiator and responder random numbers.

\[ \text{STK} = s1(TK, \text{Srand}, \text{Mrand}) \]

Link is then encrypted using STK
**STK Generation – Passkey Entry**

1. **Step 1: Passkey Entry Pairing After Phase 1: Pairing Feature Change**

   Passkey Entry pairing algorithm selected from parameters of Pairing Request and Pairing Response on Phase 1: Pairing Feature Exchange.

   - Create Mrand
   - Set TK = User input passkey

   - Create Srand
   - Set TK = User input passkey

   **Macfor** = c1(TK, Mrand, Pairing Request command, Pairing Response command, initiating device address type, initiating device address, responding device address type, responding device address)

   **Macfor** = c1(TK, Srand, Pairing Request command, Pairing Response command, initiating device address type, initiating device address, responding device address type, responding device address)

   - Check for confirm value match

   STK is calculated by running S1 with the TK value as the key input and initiator and responder random numbers.
   
   \[ STK = S1(TK, Srand, Mrand) \]

   - Link is then encrypted using STK
Example of Passkey pairing request/response

**Pairing Request**

- **Initiator can input a value**

**Pairing Response**

- **Advertiser can display a value**

Can implement MITM
# Passkey Pairing and Key Distribution

<table>
<thead>
<tr>
<th>Unfiltered</th>
<th>Info</th>
<th>Configured BT low energy devices</th>
<th>Errors</th>
<th>Undecoded L2CAP Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>LE BB</td>
<td>LE PKT</td>
<td>LE ADV</td>
<td>LE DATA</td>
<td>LE LL</td>
</tr>
<tr>
<td>2,859</td>
<td>1</td>
<td>Pairing Request</td>
<td>29</td>
<td>00:00:00.030229</td>
</tr>
<tr>
<td>2,867</td>
<td>2</td>
<td>Pairing Response</td>
<td>29</td>
<td>00:00:00.030229</td>
</tr>
<tr>
<td>4,186</td>
<td>1</td>
<td>Pairing Confirm</td>
<td>39</td>
<td>00:00:00.11669713</td>
</tr>
<tr>
<td>4,189</td>
<td>2</td>
<td>Pairing Confirm</td>
<td>39</td>
<td>00:00:00.030229</td>
</tr>
<tr>
<td>4,192</td>
<td>1</td>
<td>Pairing Random</td>
<td>39</td>
<td>00:00:00.029771</td>
</tr>
<tr>
<td>4,199</td>
<td>2</td>
<td>Pairing Random</td>
<td>39</td>
<td>00:00:00.030229</td>
</tr>
<tr>
<td>4,219</td>
<td>S</td>
<td>Encryption Information</td>
<td>43</td>
<td>00:00:00.239999</td>
</tr>
<tr>
<td>4,221</td>
<td>S</td>
<td>Master Identification</td>
<td>37</td>
<td>00:00:00.000660</td>
</tr>
<tr>
<td>4,223</td>
<td>S</td>
<td>Identity Information</td>
<td>43</td>
<td>00:00:00.000612</td>
</tr>
<tr>
<td>4,225</td>
<td>S</td>
<td>Identity Address Information</td>
<td>34</td>
<td>00:00:00.000660</td>
</tr>
<tr>
<td>4,229</td>
<td>M</td>
<td>Encryption Information</td>
<td>43</td>
<td>00:00:00.027838</td>
</tr>
<tr>
<td>4,231</td>
<td>M</td>
<td>Master Identification</td>
<td>37</td>
<td>00:00:00.000660</td>
</tr>
<tr>
<td>4,233</td>
<td>M</td>
<td>Identity Information</td>
<td>43</td>
<td>00:00:00.000612</td>
</tr>
<tr>
<td>4,235</td>
<td>M</td>
<td>Identity Address Information</td>
<td>34</td>
<td>00:00:00.000660</td>
</tr>
</tbody>
</table>
Initiating Legacy Encryption Between Paired Devices

• The Link Layer controls encryption of packets once devices have entered the Connection State.

• The Host initiates encryption. The Slave can send a Security Request command and ask the Host to initiate encryption, but only the Host can initiate encryption.

• To initiate encryption, the Host sends an encryption request (LL_ENC_REQ) to the Slave.

LL_ENC_REQ contains:
• EDIV (Encrypted Diversifier) – 16-bit value used to identify the LTK distributed during LE Legacy pairing.
• Rand – 64-bit value used to identify the LTK distributed during LE Legacy pairing.
• SKDm – Master’s portion of the session key diversifier.
• IVm – Master’s portion of the initialization vector.
Initiating Legacy Encryption Between Paired Devices
LE Legacy Encryption: Role of EDIV and Rand

- EDIV and Rand are used to identify the LTK that should be used for encrypting the connection.
- Distributed using the SMP Master Identification command during the connection setup.
- An EDIV and Rand with values of zero indicate that the STK should be used to encrypt the link. (The EDIV and Rand are zero on an initial pairing.)
- An EDIV and Rand with non-zero values indicate that the LTK associated with these values should be used to encrypt the link. (On a reconnection, the EDIV and Rand should not be zero.)
# Link Layer Initiation of Encryption

<table>
<thead>
<tr>
<th>B...</th>
<th>Frame#</th>
<th>Access Address</th>
<th>Event Counter</th>
<th>Side</th>
<th>Control Pkt</th>
<th>Instant</th>
<th>Frame...</th>
<th>Delta</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.969</td>
<td>0x50656b6d</td>
<td>0x0001</td>
<td>1</td>
<td>LL_VERSION_IND</td>
<td>24</td>
<td>00:00:00.029825</td>
<td>9/18/2015 9:14:44.5117</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.981</td>
<td>0x50656b6d</td>
<td>0x0002</td>
<td>2</td>
<td>LL_VERSION_IND</td>
<td>24</td>
<td>00:00:00.029714</td>
<td>9/18/2015 9:14:44.5415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.990</td>
<td>0x50656b6d</td>
<td>0x0003</td>
<td>1</td>
<td>Reserved [0xff]</td>
<td>24</td>
<td>00:00:00.030031</td>
<td>9/18/2015 9:14:44.5712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.993</td>
<td>0x50656b6d</td>
<td>0x0004</td>
<td>2</td>
<td>LL_UNKNOWN_RSP</td>
<td>20</td>
<td>00:00:00.030301</td>
<td>9/18/2015 9:14:44.6015</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.109</td>
<td>0x50656b6d</td>
<td>0x0027</td>
<td>1</td>
<td>LL_CHANNEL_MAP_REQ</td>
<td>0x0034</td>
<td>00:00:00.046989</td>
<td>9/18/2015 9:14:45.6512</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.425</td>
<td>0x50656b6d</td>
<td>0x0084</td>
<td>1</td>
<td>LL_CHANNEL_MAP_REQ</td>
<td>0x0091</td>
<td>00:00:00.079001</td>
<td>9/18/2015 9:14:48.4412</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.636</td>
<td>0x50656b6d</td>
<td>0x00c4</td>
<td>1</td>
<td>LL_ENC_REQ</td>
<td>41</td>
<td>00:00:00.132000</td>
<td>9/18/2015 9:14:50.3161</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.639</td>
<td>0x50656b6d</td>
<td>0x00c5</td>
<td>2</td>
<td>LL_ENC_RSP</td>
<td>31</td>
<td>00:00:00.030229</td>
<td>9/18/2015 9:14:50.3915</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.646</td>
<td>0x50656b6d</td>
<td>0x00c7</td>
<td>2</td>
<td>LL_START_ENC_REQ</td>
<td>19</td>
<td>00:00:00.060000</td>
<td>9/18/2015 9:14:50.4515</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.647</td>
<td>0x50656b6d</td>
<td>0x00c8</td>
<td>M</td>
<td>LL_START_ENC_RSP</td>
<td>23</td>
<td>00:00:00.029770</td>
<td>9/18/2015 9:14:50.4812</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.650</td>
<td>0x50656b6d</td>
<td>0x00c9</td>
<td>S</td>
<td>LL_START_ENC_RSP</td>
<td>23</td>
<td>00:00:00.030229</td>
<td>9/18/2015 9:14:50.5115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.688</td>
<td>0x50656b6d</td>
<td>0x00d0</td>
<td>M</td>
<td>LL_CONNECTION_UPDATE_REQ</td>
<td>0x00dd</td>
<td>00:00:00.209769</td>
<td>9/18/2015 9:14:50.7212</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.443</td>
<td>0x50656b6d</td>
<td>0x01a1</td>
<td>M</td>
<td>LL_CHANNEL_MAP_REQ</td>
<td>0x01ae</td>
<td>00:00:00.749586</td>
<td>9/18/2015 9:14:58.4712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.707</td>
<td>0x50656b6d</td>
<td>0x01ee</td>
<td>M</td>
<td>LL_CHANNEL_MAP_REQ</td>
<td>0x01eb</td>
<td>00:00:00.887519</td>
<td>9/18/2015 9:15:01.3587</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.126</td>
<td>0x50656b6d</td>
<td>0x037f</td>
<td>M</td>
<td>LL_CHANNEL_MAP_REQ</td>
<td>0x038c</td>
<td>00:00:00.153745</td>
<td>9/18/2015 9:15:16.3962</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.377</td>
<td>0x50656b6d</td>
<td>0x03c9</td>
<td>M</td>
<td>LL_CHANNEL_MAP_REQ</td>
<td>0x03d6</td>
<td>00:00:00.774997</td>
<td>9/18/2015 9:15:19.1712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.135</td>
<td>0x50656b6d</td>
<td>0x049b</td>
<td>S</td>
<td>LL_TERMINATE_IND</td>
<td>24</td>
<td>00:00:07.875218</td>
<td>9/18/2015 9:15:27.0464</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LE Pairing with Secure Connections Encryption

Phase I

- Pairing Request
- Pairing Response

Phase II

- Public Key Exchange (Diffie-Hellman)
- Authentication Stage 1 (Just works, # comparison, passkey, or OOB)
- Authentication Stage 2 (LTK computation)

Establish LTK-based Encryption

Phase III

- Key Distribution (IRK and CSRK)
- Key Distribution (IRK and CSRK)
Secure Connections – Just Works/Numeric Comparison

Step 2: Secure Connections Key (LTK) Generation - Just Works or Numeric Comparison

Master

- Pick Na, Set ra & rb = 0
- Calculate User Confirm Value
- User Confirms Value (Success)
- Check Confirmation - success
- Pairing Random (Ra)
- Pairing Random (Rb)
- Pairing Confirm (Cb)

Slave

- Pick Nb, Set ra & rb = 0
- Calculate confirmation
- Calculate User Confirm Value
- User Confirms Value (Success)

These values are presented to the user

For Just Works, the values are not presented to the user
Secure Connections – Just Works/Numeric Comparison (More details)

Initiating Device A

1. Authentication Stage 1: Just Works
   2a. Select Random Na
   3a. Set ra and rb to 0

Non-initiating Device B

4. Cb
5. Na
6. Nb

6a. Check if Cb = f4(PKb, PKa, Nb, 0)
   If check fails, abort

7a. Vb = g2(PKa, PKb, Na, Nb)

Proceed if user Confirms "OK"

USER checks if Va = Vb
Proceed if each USER confirms "OK"

7b. Vb = g2(PKa, PKb, Na, Nb)

Proceed if user Confirms "OK"
LE Secure Connections Pairing
(Just Works with Debug Keys)

<table>
<thead>
<tr>
<th>B...</th>
<th>Frame#</th>
<th>Side</th>
<th>Code</th>
<th>Fram...</th>
<th>Delta</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>Pairing Request</td>
<td>29</td>
<td></td>
<td>9/17/2015 11:13:12.421036 AM</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1</td>
<td>Pairing Response</td>
<td>29</td>
<td>00:00:00:250230</td>
<td>9/17/2015 11:13:12.671266 AM</td>
</tr>
<tr>
<td>15</td>
<td>1*</td>
<td>1</td>
<td>* Pairing Public Key</td>
<td>33</td>
<td>00:00:00:251123</td>
<td>9/17/2015 11:13:12.922390 AM</td>
</tr>
<tr>
<td>22</td>
<td>2*</td>
<td>2</td>
<td>* Pairing Public Key</td>
<td>33</td>
<td>00:00:00:250231</td>
<td>9/17/2015 11:13:13.172620 AM</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>2</td>
<td>Pairing Confirm</td>
<td>39</td>
<td>00:00:00:498645</td>
<td>9/17/2015 11:13:13.671265 AM</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>1</td>
<td>Pairing Random</td>
<td>39</td>
<td>00:00:00:249769</td>
<td>9/17/2015 11:13:13.921035 AM</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>2</td>
<td>Pairing Random</td>
<td>39</td>
<td>00:00:00:250230</td>
<td>9/17/2015 11:13:14.171265 AM</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>1</td>
<td>Pairing DHKey Check</td>
<td>39</td>
<td>00:00:01.249769</td>
<td>9/17/2015 11:13:15.421033 AM</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>2</td>
<td>Pairing DHKey Check</td>
<td>39</td>
<td>00:00:01.250229</td>
<td>9/17/2015 11:13:16.671263 AM</td>
</tr>
</tbody>
</table>

SMP Pairing Public Key transfer

SMP:

```plaintext
... * Code: Pairing Public Key
... * Debug Public Key X: 0xe6 9d 35 0e 48 01 03 cc db fd f4 ac 11 91 f4 e1 b9 a5 f9 e9 ...
... * Debug Public Key Y: 0x 8b d2 89 15 d0 8e 1c 74 24 30 ed 8f c2 45 83 76 5c 15 52 5a ...
```

SMP Diffie-Hellman Key check

SMP:

```plaintext
... Code: Pairing DHKey Check
... DHKey: 0x f7 db 50 a6 db 0e 0f fc 2b 1f 1c 0d 84 69 55 e1
```
Secure Connections – Passkey Entry

Example Notifications
- Pairing Keypress Notification
- Pairing Keypress Notification

Repeat 20 times
- Pick Na, ra, rb
- Calculate
- Confirm

Pick Na, ra, rb
- Calculate
- Confirm

Pairing Confirm (Ca)
- Pairing Confirm (Cb)
- Pairing Random (Nc)

Check Confirm (Confirm matches)
- Pairing Random (Nd)
Secure Connections – Passkey Entry (More details)

- **Initiating Device A**
  - 2a. Inject secret $\text{ra}$, set $\text{rb} = \text{ra}$
  - 3a. Select random $\text{Na}_i$
  - 4a. Compute confirm: $\text{Ca}_i = f(\text{PKa}, \text{PKb}, \text{Na}_i, \text{ra})$

- **Non-initiating Device B**
  - 2b. Inject secret $\text{rb}$, set $\text{ra} = \text{rb}$
  - 3b. Select random $\text{Nb}_i$
  - 4b. Compute confirm: $\text{Cb}_i = f(\text{PKb}, \text{PKa}, \text{Nb}_i, \text{rb})$

- **Authentication Stage 1:** Passkey Entry
  - Execute 20 times:
    - $\text{ra} = \text{ra}_1 \mid \text{ra}_2 \mid \ldots \mid \text{ra}_{20}$
    - $\text{rb} = \text{rb}_1 \mid \text{rb}_2 \mid \ldots \mid \text{rb}_{20}$
    - New random numbers are selected in each round
  - 5. $\text{Ca}_i$
  - 6. $\text{Cb}_i$
  - 7. $\text{Na}_i$
  - 7a. Check if $\text{Ca}_i = f(\text{PKa}, \text{PKb}, \text{Na}_i, \text{ri})$
    - If check fails, abort
  - 8. $\text{Nb}_i$
  - 8a. Check if $\text{Cb}_i = f(\text{PKb}, \text{PKa}, \text{Nb}_i, \text{ri})$
    - If check fails, abort
Secure Connections Pairing with Passkey
(Not Debug keys)
Secure Connections – Out of Band

Step 2: Secure Connection Key (LTK) Generation - Out of Band

Master

Check Confirm received from OOB (Confirm matches)

Pick Na

Slave

Check Confirm received from OOB (Confirm matches)

Pick Nb

Pairing Random (Na)

Pairing Random (Nb)
LE Secure Connections LTK Calculation

9. Compute the LTK and MacKey:
   MacKey || LTK = f5(DHKey,Na,Nb,A,B)

10a. Compute: 
   Ea = f6(MacKey,Na,Nb,rb,IOcapA,A,B)

10. Ea

11. Check if 
   Ea = f6(MacKey,Na,Nb,rb,IOcapA,A,B)
   If check fails, abort.

12a. Check if 
   Eb = f6(MacKey,Nb,Na,ra,IOcapB,B,A)
   If check fails, abort.
The Bluetooth Smart (LE) Privacy feature reduces the ability to track a device over a period of time by changing the device address on a frequent basis.

The address of a device using Privacy mode can be “resolved” using the Identity Resolving Key (IRK) which is one of the encryption keys exchanged during the pairing process.

Resolvable Address Format

<table>
<thead>
<tr>
<th>Hash</th>
<th>prand</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 bits</td>
<td>24 bits</td>
</tr>
</tbody>
</table>

\[ hash = ah(\text{IRK}, \text{prand}) \]
Cross-transport Key Derivation

When a pair of BR/EDR/LE devices support Secure Connections on a transport, the devices may optionally generate a key of identical strength for the other transport.

The LTK from the LE physical transport can be converted to the BR/EDR link key for the BR/EDR transport as follows, using intermediate link key (ILK) as an intermediate value:
1. ILK = h6(LTK, “tmp1”)
2. BR/EDR link key = h6(ILK, “lebr”)

The string “lebr” is mapped into keyID using extended ASCII as follows:

keyID[0] = 0111 0010
keyID[1] = 0110 0010
keyID[2] = 0110 0101
keyID[3] = 0110 1100
keyID = 0x6c666272

The string “tmp1” is mapped into keyID using extended ASCII as follows:

keyID[0] = 0011 0001
keyID[1] = 0111 0000
keyID[2] = 0110 1101
keyID[3] = 0111 0100
keyID = 0x746d7031

The BR/EDR Link Key from the BR/EDR physical transport can be converted to the LTK for the LE transport as follows, using intermediate long term key (ILTK) as an intermediate value:
1. ILTK = h6(Link Key, “tmp2”)
2. LTK = h6(ILTK, “brle”)

The string “brle” is mapped into keyID using extended ASCII as follows:

keyID[0] = 0110 0101
keyID[1] = 0110 1100
keyID[2] = 0111 0010
keyID[3] = 0110 0010
keyID = 0x62726ec65

The string “tmp2” is mapped into keyID using extended ASCII as follows:

keyID[0] = 0011 0010
keyID[1] = 0111 0000
keyID[2] = 0110 1101
keyID[3] = 0111 0100
keyID = 0x746d7032
## Service Request Behavior Based on Authentication Requirements

<table>
<thead>
<tr>
<th>Link Encryption State</th>
<th>Link's Access Requirement for Service</th>
<th>Local Device's Pairing Status</th>
<th>Local Device Pairing Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local Device's Pairing Status</td>
<td>Unauthenticated LTK</td>
<td>Authenticated LTK</td>
</tr>
<tr>
<td></td>
<td>No LTK</td>
<td>Request succeeds</td>
<td>Request succeeds</td>
</tr>
<tr>
<td>Unencrypted</td>
<td>Encryption, No MITM Protection</td>
<td>Error Resp.: Insufficient Authentication</td>
<td>Error Resp.: Insufficient Encryption</td>
</tr>
<tr>
<td></td>
<td>Encryption, MITM Protection</td>
<td>Error Resp.: Insufficient Authentication</td>
<td>Error Resp.: Insufficient Encryption</td>
</tr>
<tr>
<td>Encrypted</td>
<td>None</td>
<td>Request succeeds</td>
<td>Request succeeds</td>
</tr>
<tr>
<td></td>
<td>Encryption, No MITM Protection</td>
<td>Request succeeds</td>
<td>Request succeeds</td>
</tr>
<tr>
<td></td>
<td>Encryption, MITM Protection</td>
<td>Error Resp.: Insufficient Authentication</td>
<td>Request succeeds</td>
</tr>
<tr>
<td></td>
<td>Encryption, MITM Protection, Secure Connections</td>
<td>Error Resp.: Insufficient Authentication</td>
<td>Request succeeds</td>
</tr>
</tbody>
</table>