Expedited Errata Correction 11838: Encryption Key Size Updates

***Bluetooth*® Expedited Errata Correction**

- **Revision:** v1.0
- **Revision Date:** 2019-08-13
- **Group Prepared By:** Core Specification Working Group
- **Feedback Email:** core-main@bluetooth.org

This Expedited Errata Correction is mandatory and applies to the following specifications (collectively, the "Source Specifications"):

- Core Specification v5.1 [1]
- Core Specification v5.0 [2]
- Core Specification v4.2 [3]

Abstract:

This document specifies the changes to be applied to the Core Specifications required to incorporate the various encryption key size related updates.

Revision History

Revision Number	Date	Comments
v1.0	2019-08-13	Adopted by the Bluetooth SIG Board of Directors.

Contributors

Name	Company
Mayank Batra	Qualcomm Technologies International, Ltd.
Clive D.W. Feather	Samsung Cambridge Solution Centre
Harish Balasubramaniam	Intel Corporation
Robert Hulvey	Cypress Semiconductor Corporation
Alain Michaud	Microsoft Corporation

Contents

1 Drafting conventions

1.1 Language

Please refer to and follow any terminology, language conventions, and interpretation sections of the Source Specifications.

1.2 Formatting and color

The formatting and color conventions described in Table 2.1 below are used in this Expedited Errata Correction to describe the specific changes and additions to the Source Specifications identified on the cover page.

Text Color	Description
black	Text that is unmodified from the Source Specification.
red	Text that is added to the Source Specification.
~~red strikethrough~~	Text that is deleted from the Source Specification.
[green bracketed text]	Comments that are intended to aid the reader.
blue	Default color used for section numbers and headings of this document.

Table 2.1: Color key for headings, captions, and body text

2 Changes to Core Specification v5.1

This Section sets forth the specific changes and additions, using the formatting and color conventions described in Section 1.2, to Core Specification v5.1.

2.1 Changes to Core Specification v5.1, Volume 2, Part C: Link Manager Protocol Specification

2.1.1 [Modified Section] *4.2.5.2 Encryption key size*

[Modify the second paragraph as shown below and add a new paragraph and a note after what is now the fourth paragraph.]

The master sends an LMP_encryption_key_size_req PDU including the suggested key size $L_{sug,\ m}$~~, m~~, that shall ~~is~~ initially be equal to $L_{max,\ m}$. If $L_{min,\ s} \leq L_{sug,\ m} \leq L_{max,\ s}$ ~~and the slave supports $L_{sug,\ m}$ it~~, the slave shall respond with an LMP_accepted PDU and $L_{sug,\ m}$ shall be used as the key size.

[Insert paragraph break]

If $L_{sug,m} > L_{max,s}$, the slave shall send back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug,s}$ set to $L_{max,s}$. If $L_{sug,m} < L_{min,s}$, the slave shall send back an LMP_not_accepted PDU with the error code *Unsupported LMP Parameter Value* (0x20) and the devices shall not communicate using encryption. ~~If both conditions are not fulfilled the slave sends back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug,\ s}$. This value shall be the slave's largest supported key size that is less than $L_{sug,\ m}$.~~

[Insert paragraph break]

If the slave sends back an LMP_encryption_key_size PDU, t~~T~~hen the master performs the corresponding test on the slave's suggestion. This procedure is repeated until a key size agreement is reached or it becomes clear that no such agreement can be reached. If an agreement is reached a device sends an LMP_accepted PDU and the key size in the last LMP_encryption_key_size_req PDU shall be used.

[Insert paragraph break]

If a key size is agreed, encryption is then started ~~After this, encryption is started~~; see Section 4.2.5.3. If an agreement is not reached a device sends an LMP_not_accepted PDU with the error code *Unsupported LMP Parameter Value (0x20)* and the devices shall not communicate using encryption.

$L_{max,\ m}$ and $L_{max,\ s}$ shall be set to at least 7 octets. $L_{min,\ m}$ and $L_{min,\ s}$ should be set to at least 7 octets. The values of $L_{max,\ m}$, $L_{min,\ m}$, $L_{max,\ s}$ and $L_{min,\ s}$ shall not change during an ACL connection between the master and the slave.

Note: If the Host of either the master or the slave uses services that require security mode 4 (see [Vol 3] Part C, Section 5.2.2.8), a key size higher than the key size negotiated by the two Link Managers can be enforced.

2.2 Changes to Core Specification v5.1, Volume 2, Part E: Host Controller Interface Functional Specification

2.2.1 [Modified Section] 7.5.7 Read Encryption Key Size command

[Add a new paragraph at the end of the Description as shown below.]

Description:

This command reads the current encryption key size associated with the Connection_Handle. The Connection_Handle shall be a Connection_Handle for an active ACL connection.

All BR/EDR Controllers shall implement this command.

2.3 Changes to Core Specification v5.1, Volume 3, Part C: Generic Access Profile

2.3.1 [Modified Section] *5.2.2.8 Security database*

[Modify the text as shown below.]

A Bluetooth device in security mode 4 shall classify and enforce the security requirements of its services using ~~at least~~ the following levels attributes (in order of decreasing security) for use when pairing with remote devices supporting Secure Simple Pairing:

• Level 4, for services with the following attributes ~~or devices in Secure Connections Only Mode~~:

  MITM protection required

  128-bit equivalent strength for link and encryption keys required using FIPS approved algorithms (E0 not allowed, SAFER+ not allowed, and P-192 not allowed; encryption key not shortened)

  User interaction acceptable

• Level 3, for services with the following attributes:

  MITM protection required

  Encryption required

  At least 56-bit equivalent strength for encryption key should be used

  User interaction acceptable

• Level 2, for services with the following attributes:

  MITM protection not required

  Encryption required

  At least 56-bit equivalent strength for encryption key should be used

• Level 1, for services with the following attributes:

MITM protection not required

At least 56-bit equivalent strength for encryption key when encryption is enabled should be used

Minimal user interaction desired

• Level 0: Service requires the following:

MITM protection not required

No encryption required

No user interaction required

Security Mode 4 Level 0 shall only be used for:

a) L2CAP fixed signaling channels with CIDs 0x0001, 0x0003, and 0x003F

b) SDP

c) broadcast data sent on the connectionless L2CAP channel (CID 0x0002)

d) services with the combinations of Service Class UUIDs and L2CAP traffic types listed in [Core Specification Supplement], Part C, Section 1.

The security level required for each service offered should be stored in a security database that is accessed to determine the type of link key and the encryption key size that is required for access to the respective service. The security level required for service data transmitted on an L2CAP connection-oriented channel may differ from the security level required for service data transmitted on another L2CAP connection-oriented channel or on the connectionless L2CAP channel. Table 5.8 shows the type of link key required for each security level for both remote devices that support Secure Simple Pairing (v2.1 + EDR remote devices) and for those that do not (pre-v2.1 + EDR remote devices).

[...]

A previously generated link key is considered “sufficient” if the link key type is of the type required for the service, or of a higher strength. Authenticated link keys are considered higher strength than Unauthenticated or Combination keys. Unauthenticated link keys are considered higher strength than Combination keys.

A device shall enforce an encryption key with at least 128-bit equivalent strength for all services that require Security Mode 4, Level 4. For all other services that require encryption, a device should enforce an encryption key with at least 56-bit equivalent strength, irrespective of whether the remote device supports Secure Simple Pairing.

After encryption has been enabled, the Host should check the encryption key size using either the HCI_Read_Encryption_Key_Size command (see [Vol 2] Part E, Section 7.5.7) or a vendor-specific method.

3 Changes to Core Specification v5.0

This Section sets forth the specific changes and additions, using the formatting and color conventions described in Section 1.2, to Core Specification v5.0.

3.1 Changes to Core Specification v5.0, Volume 2, Part C: Link Manager Protocol Specification

3.1.1 [Modified Section] *4.2.5.2 Encryption key size*

[Insert a paragraph break after the Note as shown below and add a new paragraph and a note after what is now the fourth paragraph.]

Note: This section uses the same terms as in [Vol 2] Part H, Section 4.1.

The master sends an LMP_encryption_key_size_req PDU including the suggested key size $L_{sug,\ m}$~~, m~~, that shall ~~is~~ initially be equal to $L_{max,\ m}$. If $L_{min,\ s} \leq L_{sug,\ m} \leq L_{max,\ s}$ ~~and the slave supports $L_{sug,\ m}$ it~~, the slave shall respond with an LMP_accepted PDU and $L_{sug,\ m}$ shall be used as the key size.

[Insert paragraph break]

If $L_{sug,m} > L_{max,s}$, the slave shall send back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug,s}$ set to $L_{max,s}$. If $L_{sug,m} < L_{min,s}$, the slave shall send back an LMP_not_accepted PDU with the error code *Unsupported LMP Parameter Value* (0x20) and the devices shall not communicate using encryption. ~~If both conditions are not fulfilled the slave sends back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug,\ s}$. This value shall be the slave's largest supported key size that is less than $L_{sug,\ m}$.~~

[Insert paragraph break]

If the slave sends back an LMP_encryption_key_size PDU, t~~T~~hen the master performs the corresponding test on the slave's suggestion. This procedure is repeated until a key size agreement is reached or it becomes clear that no such agreement can be reached. If an agreement is reached a device sends an LMP_accepted PDU and the key size in the last LMP_encryption_key_size_req PDU shall be used.

[Insert paragraph break]

If a key size is agreed, encryption is then started ~~After this, encryption is started~~; see Section 4.2.5.3. If an agreement is not reached a device sends an LMP_not_accepted PDU with the error code *Unsupported LMP Parameter Value (0x20)* and the devices shall not communicate using encryption.

$L_{max,\ m}$ and $L_{max,\ s}$ shall be set to at least 7 octets. $L_{min,\ m}$ and $L_{min,\ s}$ should be set to at least 7 octets. The values of $L_{max,\ m}$, $L_{min,\ m}$, $L_{max,\ s}$ and $L_{min,\ s}$ shall not change during an ACL connection between the master and the slave.

Note: If the Host of either the master or the slave uses services that require security mode 4 (see [Vol 3] Part C, Section 5.2.2.8), a key size higher than the key size negotiated by the two Link Managers can be enforced.

3.2 Changes to Core Specification v5.0, Volume 2, Part E: Host Controller Interface Functional Specification

3.2.1 [Modified Section] 7.5.7 Read Encryption Key Size command

[Add a new paragraph at the end of the Description as shown below.]

Description:

This command reads the current encryption key size associated with the Connection_Handle. The Connection_Handle shall be a Connection_Handle for an active ACL connection.

All BR/EDR Controllers shall implement this command.

3.3 Changes to Core Specification v5.0, Volume 3, Part C: Generic Access Profile

3.3.1 [Modified Section] *5.2.2.8 Security database*

[Modify the text as shown below.]

A Bluetooth device in security mode 4 shall classify and enforce the security requirements of its services using ~~at least~~ the following levels attributes (in order of decreasing security) for use when pairing with remote devices supporting Secure Simple Pairing:

• Level 4, for services with the following attributes ~~or devices in Secure Connections Only Mode~~:

  MITM protection required

  128-bit equivalent strength for link and encryption keys required using FIPS approved algorithms (E0 not allowed, SAFER+ not allowed, and P-192 not allowed; encryption key not shortened)

  User interaction acceptable

• Level 3, for services with the following attributes:

  MITM protection required

  Encryption required

  At least 56-bit equivalent strength for encryption key should be used

  User interaction acceptable

• Level 2, for services with the following attributes:

  MITM protection not required

  Encryption required

  At least 56-bit equivalent strength for encryption key should be used

• Level 1, for services with the following attributes:

MITM protection not required

At least 56-bit equivalent strength for encryption key when encryption is enabled should be used

Minimal user interaction desired

• Level 0: Service requires the following:

MITM protection not required

No encryption required

No user interaction required

Security Mode 4 Level 0 shall only be used for:

a) L2CAP fixed signaling channels with CIDs 0x0001, 0x0003, and 0x003F

b) SDP

c) broadcast data sent on the connectionless L2CAP channel (CID 0x0002)

d) services with the combinations of Service Class UUIDs and L2CAP traffic types listed in [Core Specification Supplement], Part C, Section 1.

The security level required for each service offered should be stored in a security database that is accessed to determine the type of link key and the encryption key size that is required for access to the respective service. The security level required for service data transmitted on an L2CAP connection-oriented channel may differ from the security level required for service data transmitted on another L2CAP connection-oriented channel or on the connectionless L2CAP channel. Table 5.8 shows the type of link key required for each security level for both remote devices that support Secure Simple Pairing (v2.1 + EDR remote devices) and for those that do not (pre-v2.1 + EDR remote devices).

[...]

A previously generated link key is considered “sufficient” if the link key type is of the type required for the service, or of a higher strength. Authenticated link keys are considered higher strength than Unauthenticated or Combination keys. Unauthenticated link keys are considered higher strength than Combination keys.

A device shall enforce an encryption key with at least 128-bit equivalent strength for all services that require Security Mode 4, Level 4. For all other services that require encryption, a device should enforce an encryption key with at least 56-bit equivalent strength, irrespective of whether the remote device supports Secure Simple Pairing.

After encryption has been enabled, the Host should check the encryption key size using either the HCI_Read_Encryption_Key_Size command (see [Vol 2] Part E, Section 7.5.7) or a vendor-specific method.

4 Changes to Core Specification v4.2

This Section sets forth the specific changes and additions, using the formatting and color conventions described in Section 1.2, to Core Specification v4.2.

4.1 Changes to Core Specification v4.2, Volume 2, Part C: Link Manager Protocol Specification

4.1.1 [Modified Section] *4.2.5.2 Encryption key size*

[Modify the Note as shown below, insert a paragraph break after the Note, and add a new paragraph and a note after what is now the fourth paragraph.]

Note: T~~t~~his section uses the same terms as in [Vol 2] Part H, Section 4.1.

The master sends an LMP_encryption_key_size_req PDU including the suggested key size $L_{sug,\ m}$~~, m~~, that shall ~~is~~ initially be equal to $L_{max,\ m}$. If $L_{min,\ s} \leq L_{sug,\ m} \leq L_{max,\ s}$ ~~and the slave supports $L_{sug,\ m}$ it~~, the slave shall respond with an LMP_accepted PDU and $L_{sug,\ m}$ shall be used as the key size.

[Insert paragraph break]

If $L_{sug,m} > L_{max,s}$, the slave shall send back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug,s}$ set to $L_{max,s}$. If $L_{sug,m} < L_{min,s}$, the slave shall send back an LMP_not_accepted PDU with the error code *Unsupported LMP Parameter Value* (0x20) and the devices shall not communicate using encryption. ~~If both conditions are not fulfilled the slave sends back an LMP_encryption_key_size_req PDU including the slave's suggested key size $L_{sug,\ s}$. This value shall be the slave's largest supported key size that is less than $L_{sug,\ m}$.~~

[Insert paragraph break]

If the slave sends back an LMP_encryption_key_size PDU, t~~T~~hen the master performs the corresponding test on the slave's suggestion. This procedure is repeated until a key size agreement is reached or it becomes clear that no such agreement can be reached. If an agreement is reached a device sends an LMP_accepted PDU and the key size in the last LMP_encryption_key_size_req PDU shall be used.

[Insert paragraph break]

If a key size is agreed, encryption is then started ~~After this, encryption is started~~; see Section 4.2.5.3. If an agreement is not reached a device sends an LMP_not_accepted PDU with the error code *Unsupported LMP Parameter Value (0x20)* and the devices shall not communicate using encryption.

$L_{max,\ m}$ and $L_{max,\ s}$ shall be set to at least 7 octets. $L_{min,\ m}$ and $L_{min,\ s}$ should be set to at least 7 octets. The values of $L_{max,\ m}$, $L_{min,\ m}$, $L_{max,\ s}$ and $L_{min,\ s}$ shall not change during an ACL connection between the master and the slave.

Note: It the Host of either the master or the slave uses services that require security mode 4 (see [Vol 3] Part C, Section 5.2.2.8), a key size higher than the key size negotiated by the two Link Managers can be enforced.

4.2 Changes to Core Specification v4.2, Volume 2, Part E: Host Controller Interface Functional Specification

4.2.1 [Modified Section] 7.5.7 Read Encryption Key Size command

[Add a new paragraph at the end of the Description as shown below.]

Description:

This command reads the current encryption key size associated with the Connection_Handle. The Connection_Handle shall be a Connection_Handle for an active ACL connection.

All BR/EDR Controllers shall implement this command.

4.3 Changes to Core Specification v4.2, Volume 3, Part C: Generic Access Profile

4.3.1 [Modified Section] *5.2.2.8 Security database*

[Modify the text as shown below.]

A Bluetooth device in security mode 4 shall classify and enforce the security requirements of its services using ~~at least~~ the following levels attributes (in order of decreasing security) for use when pairing with remote devices supporting Secure Simple Pairing:

• Level 4, for services with the following attributes ~~or devices in Secure Connections Only Mode~~:

MITM protection required

128-bit equivalent strength for link and encryption keys required using FIPS approved algorithms (E0 not allowed, SAFER+ not allowed, and P-192 not allowed; encryption key not shortened)

User interaction acceptable

• Level 3, for services with the following attributes:

MITM protection required

Encryption required

At least 56-bit equivalent strength for encryption key should be used

User interaction acceptable

• Level 2, for services with the following attributes:

MITM protection not required

Encryption required

At least 56-bit equivalent strength for encryption key should be used

• Level 1, for services with the following attributes:

MITM protection not required

At least 56-bit equivalent strength for encryption key when encryption is enabled should be used

Minimal user interaction desired

• Level 0: Service requires the following:

MITM protection not required

No encryption required

No user interaction required

Security Mode 4 Level 0 shall only be used for:

a) L2CAP fixed signaling channels with CIDs 0x0001, 0x0003, and 0x003F

b) SDP

c) broadcast data sent on the connectionless L2CAP channel (CID 0x0002)

d) services with the combinations of Service Class UUIDs and L2CAP traffic types listed in [Core Specification Supplement], Part C, Section 1.

The security level required for each service offered should be stored in a security database that is accessed to determine the type of link key and the encryption key size that is required for access to the respective service. The security level required for service data transmitted on an L2CAP connection-oriented channel may differ from the security level required for service data transmitted on another L2CAP connection-oriented channel or on the connectionless L2CAP channel. Table 5.8 shows the type of link key required for each security level for both remote devices that support Secure Simple Pairing (v2.1 + EDR remote devices) and for those that do not (pre-v2.1 + EDR remote devices).

[...]

A previously generated link key is considered “sufficient” if the link key type is of the type required for the service, or of a higher strength. Authenticated link keys are considered higher strength than Unauthenticated or Combination keys. Unauthenticated link keys are considered higher strength than Combination keys.

A device shall enforce an encryption key with at least 128-bit equivalent strength for all services that require Security Mode 4, Level 4. For all other services that require encryption, a device should enforce an encryption key with at least 56-bit equivalent strength, irrespective of whether the remote device supports Secure Simple Pairing.

After encryption has been enabled, the Host should check the encryption key size using either the HCI_Read_Encryption_Key_Size command (see [Vol 2] Part E, Section 7.5.7) or a vendor-specific method.

5 References

[1] Core Specification version 5.1, dated 2019-Jan-21, location
https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=457080

[2] Core Specification version 5.0, dated 2016-Dec-06, location
https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043

[3] Core Specification version 4.2, dated 2014-Dec-02, location
https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=286439