Erratum 10395: Mandating ECDH Public Key Validation

***Bluetooth*® Erratum**

- **Revision:** v1.0
- **Revision Date:** 2018-07-16
- **Group Prepared By:** Mesh Working Group
- **Feedback Email:** mesh-main@bluetooth.org

This Erratum is mandatory and applies to the following specification:

- Bluetooth Mesh Profile Specification Version 1.0 [1] (“Source Specification”)

Abstract:

This erratum requires verification of a public key of the Provisioner and the unprovisioned device during the provisioning of the device on a mesh network. When the public key is invalid, the provisioning process is aborted.

Revision History

Revision Number	Date	Comments
v1.0	2018-07-16	Adopted by the Bluetooth SIG Board of Directors

Contributors

Name	Company
Piotr Winiarczyk	Silvair, Inc.
Robert Cragie	Arm Limited
Szymon Slupik	Silvair, Inc.

Contents

1 Language

1.1 Language conventions

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

2 Conventions used in this erratum

The formatting and color conventions described in Table 2.1 below are used in this erratum to describe the specific changes and additions to the Source Specification(s) 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

3 Changes to Bluetooth Mesh Profile Specification Version 1.0

This Section sets forth the specific changes and additions, using the formatting and color conventions described in Section 2, to Bluetooth Mesh Profile Specification Version 1.0.

3.1 Changes to Bluetooth Mesh Profile Specification Version 1.0

3.1.1 [Modified Section] 5.4.2.3 Exchanging public keys

[Add new text in Section 5.4.2.3.]

If the public key was not available using an OOB technology, then the public keys are exchanged between the Provisioner and the unprovisioned ~~both~~ device~~s~~. For each exchange, a new key pair shall be generated by the Provisioner and the unprovisioned device.

The device shall send its public key if the key is not delivered OOB.

The message sequence for public key exchange when the unprovisioned device public key is unknown is illustrated by Figure 5.14 below.

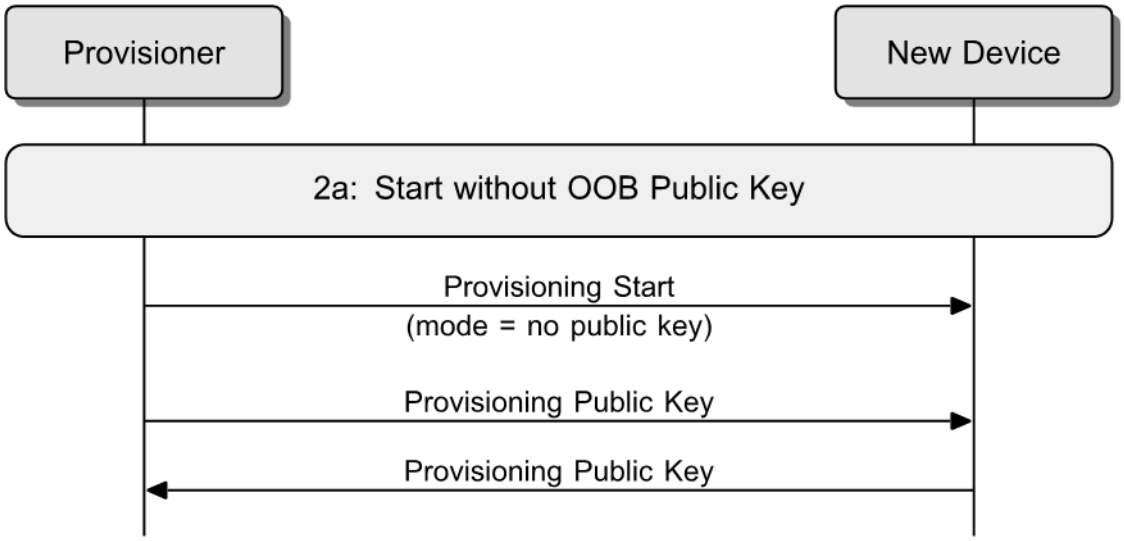

Figure 3.1: *Public key exchange when unprovisioned device public key is unknown*

Otherwise, if the public key is available via an OOB mechanism, then a new key pair shall be generated by the Provisioner, and the ~~an ephemeral~~ public key of the generated key pair shall be transmitted from the Provisioner to the device, and a static public key shall be read from the device using the appropriate OOB technology.

The message sequence for public key exchange when the unprovisioned device public key is delivered OOB is illustrated by Figure 5.15 below.

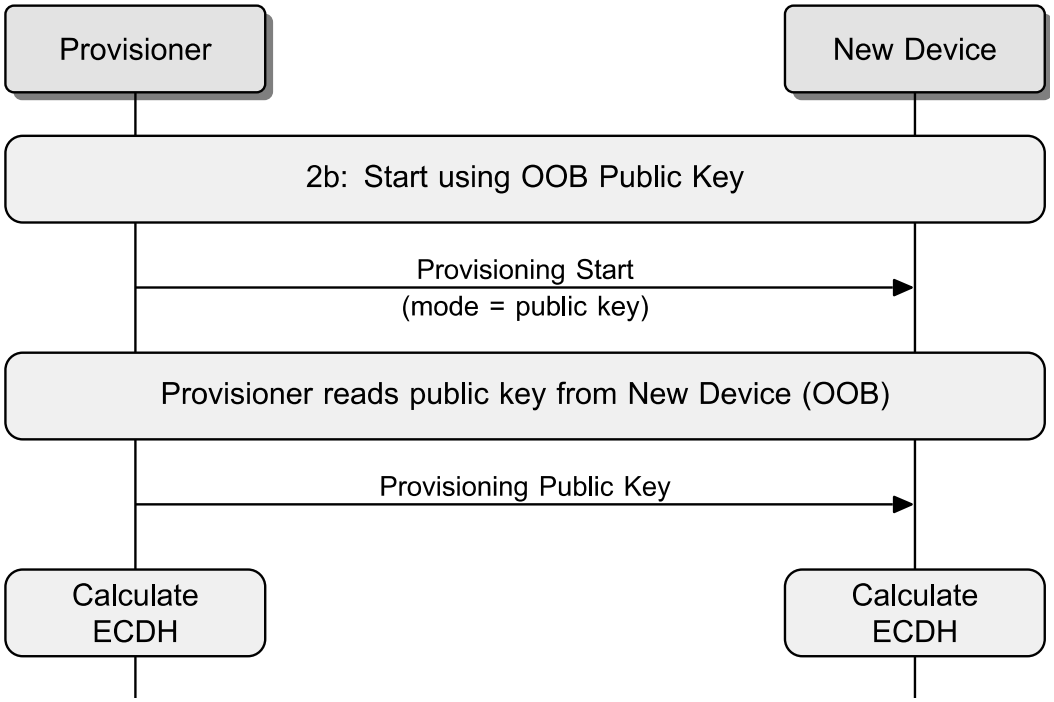

Figure 3.2: *Public key exchange when unprovisioned device public key is out-of-band*

The Provisioner and the device shall check whether the public key provided by the peer device or obtained OOB is valid (see Section 5.4.3.1).

When the Provisioner receives an invalid public key, then provisioning fails, and the Provisioner shall act as described in Section 5.4.4. When the device receives an invalid public key, then provisioning fails, and the device shall act as described in Section 5.4.4.

~~Once~~After the ~~P~~public ~~K~~key ~~of the peer device~~ is known and has been validated, the ECDHSecret shall be computed using the following formula:

ECDHSecret = P-256(private key, peer public key)

After the ECDHSecret is computed, the Provisioner and the unprovisioned device shall delete its private-public key pair that was generated in this step.

3.1.2 [Modified Section] 5.4.3.1 FIPS P-256 Elliptic Curve definition

[Change the paragraph and add new text at the bottom of Section 5.4.3.1.]

The private keys shall be between 1 and r/2, where r is the Order of the Abelian Group on the elliptic curve (~~e.g.~~i.e., between 1 and $2^{256}/2$).

A valid public key Q = (X_Q, Y_Q) is one where X_Q and Y_Q are both in the range 0 to p - 1 and satisfy the equation $(Y_Q)^2 = (X_Q)^3 + aX_Q + b \pmod p$ in the relevant curve's finite field.

Note: For additional information about public key validation, see NIST Special Publication 800-56A, Revision 3 [13].

3.1.3 [Modified Section] 9 References

[Add new text at the bottom of Section 9.]

[12] FIPS PUB 186-4 (http://dx.doi.org/10.6028/NIST.FIPS.186-4)

[13] NIST Special Publication 800-56A, Revision 3 (http://dx.doi.org/10.6028/NIST.SP.800-56Ar3)

4 References

[1] Bluetooth Mesh Profile Specification Version 1.0