WHITEPAPER

Bluetooth Mesh Feature Enhancements

Six Bluetooth Mesh Feature Enhancements to Get Excited About

Introduction

Since the Bluetooth Special Interest Group (SIG) introduced the Bluetooth® mesh topology five years ago, Internet of Things (IoT) device connectivity and network communication have been completely transformed. Mesh topology enables multi hop and many-to-many communication and it is well suited for large-scale device network applications. Mesh networks simplify smart home network communication and make it easier for network engineers & IT to manage the large IoT networks used for building automation and industrial asset tracking.

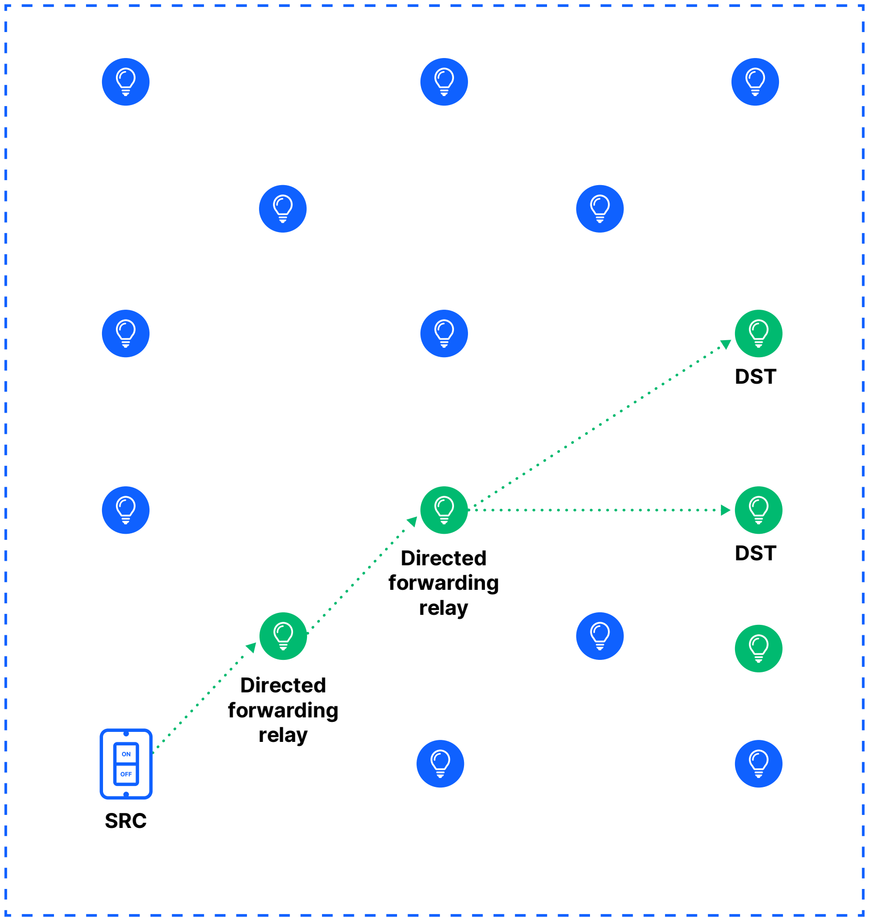
The new Bluetooth mesh feature enhancements announced by the Bluetooth SIG will increase security, reduce power consumption, simplify setup, improve mesh network efficiency and scalability. This white paper examines six of the key new features, Directed Forwarding, Subnet Bridging, Remote Provisioning, Certificate-based Provisioning, Device Firmware Updates, Private Beacons, along with the few minor enhancements added to the existing features. We are most excited about these new features at Silicon Labs and are preparing for this release so that device developers have access to these features as soon as possible.

Directed Forwarding

Bluetooth® mesh is message oriented. In a traditional mesh network, messages are flooded, which means messages are not routed in a direct path and there are multiple pathways a message can take to reach its intended destination.

To eliminate common issues with flooding a message such as message looping, scalability, and high-power consumption, mesh 1.0 introduced the concept of managed flooding. However, managed flooding still presents some issues with not efficiently targeting destinations as messages need to travel across multiple parts of the network.

The new Bluetooth mesh enhancements will alleviate this issue and improve message delivery efficiency with a new concept for delivering messages known as directed forwarding. With directed forwarding, an additional multi-hop message delivery method is used on a collection of relay nodes to establish direct paths from the defined source to the destination. The direct path can consist of one or more lanes, with a lane being a set of nodes that can provide a directed forwarding service for a source address, destination address, or subnet. The nodes can propagate a message directly to the intended destination instead of more widely broadcasting the message throughout the network. This approach optimizes the number of hops needed to relay a message – saving power and improving network efficiency.

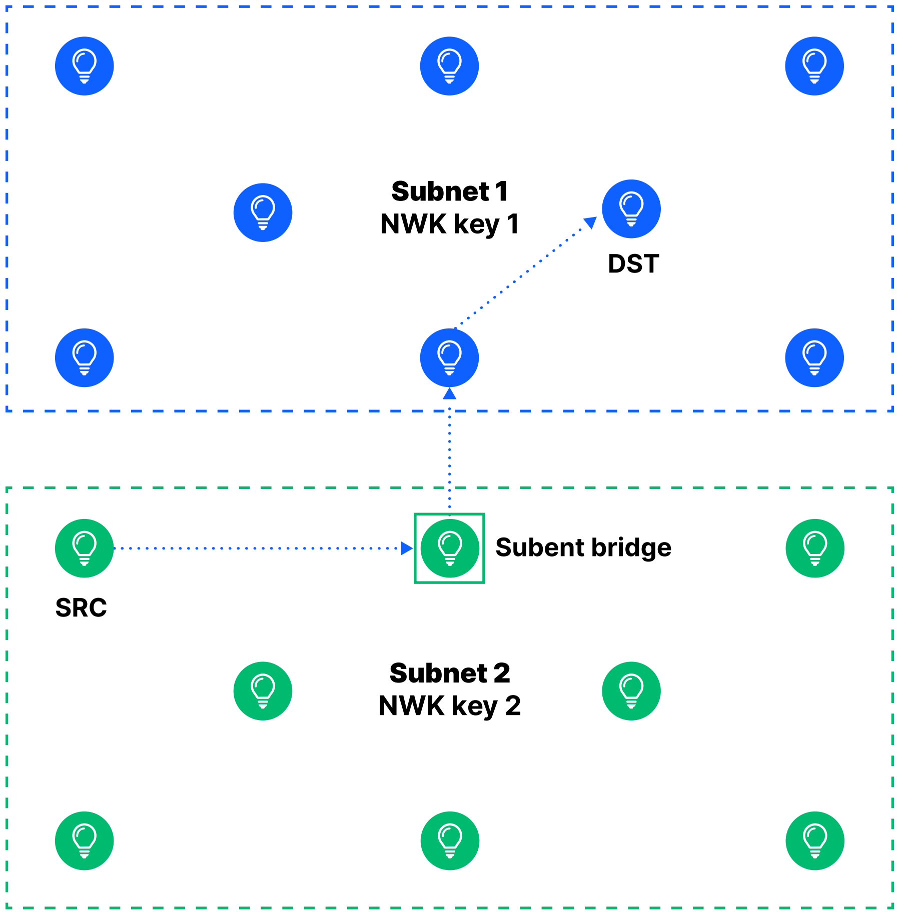


Subnet Bridging

In a Bluetooth® mesh network, you can securely isolate different parts of a building network using network security keys to create subnets. With mesh 1.0, once devices are isolated into different subnets, devices cannot communicate unless the devices are members of a common subnet and have access to the subnet's network key (NetKey). While this configuration is more secure, this setup can lead to communication inefficiencies in the network.

To continue to provide security in the network through isolation yet allow devices in different subnets the ability to communicate efficiently, these enhancements introduce a subnet bridging feature. Subnet bridging allows networks to include subnets for area isolation, but it also allows selective communication between specific devices in different, adjacent subnets without needing to know the NetKey.

To make a node a subnet bridge, a state called Subnet Bridge must be enabled on that node and a further state called the Bridging Table must be populated with entries that configure the subnet bridging functionality of the node. The Bridging Table contains entries defining source and destination addresses for the subnet bridging allowed in the network. With subnet bridging, the required communication in the network can be performed without compromising the security features provided by using subnets and isolation.



Remote Provisioning

With mesh 1.0, provisioning to add a new device to the network can only be done over a single hop. This means the provisioner needs to be within direct range of the new device, which can be challenging in large buildings with complex networks. The new Remote Provisioning feature removes the requirement to be within radio range of the provisioner by adding remote provisioning capability. Therefore, the provisioner and un-provisioned device can be in any location within the building, and the provisioning can be carried out over the mesh network since provisioning messages can now take one or more hops to the remote, un-provisioned device.

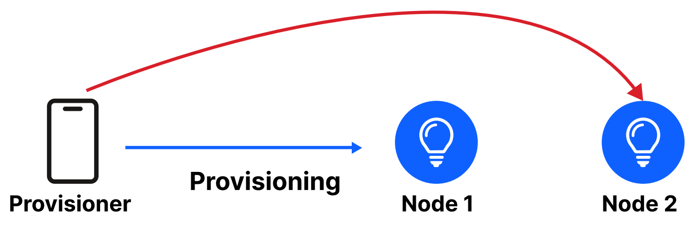


Fig 1: Without Remote Provisioning

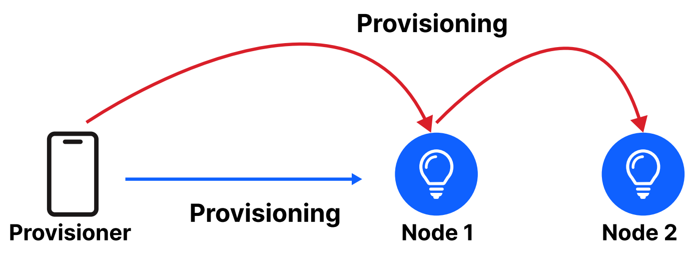
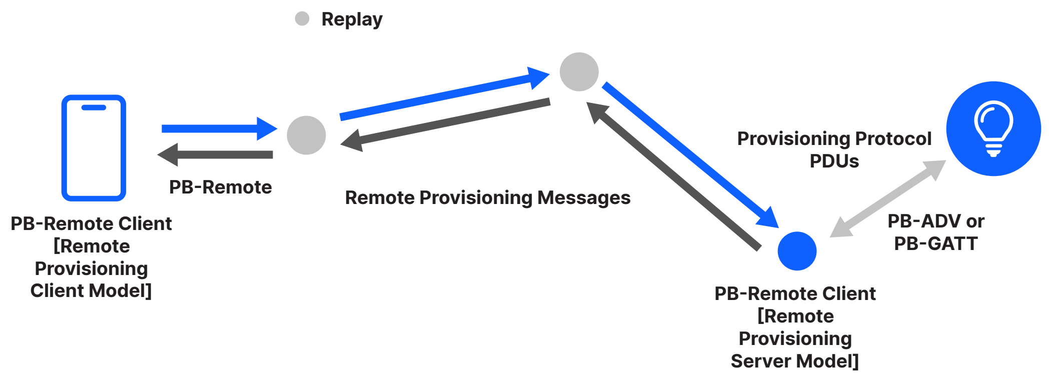


Fig 2: With Remote Provisioning

More specifically, to start the remote provisioning process, the Provisioning Bearer PB-Remote Server needs to establish a link to a selected, un-provisioned device at the request of the PB-Remote Client. It then receives provisioning protocol data units (PDUs) encapsulated within remote provisioning messages, extracts those PDUs, and sends the PDUs over the link. Provisioning protocol PDUs are received in response that are encapsulated within other remote provisioning messages to send back to the PB-Remote Client. The PB-Remote Server executes the usual provisioning process with a selected, un-provisioned device on behalf of a PB-Remote Client, which is located somewhere else in the network. As shown in the example in the diagram below, a mobile phone can be used as the PB-Remote Client for remote provisioning, even though the mobile phone is a Bluetooth LE device, not mesh.


Replay
PB-Remote
PB-Remote Client [Remote Provisioning Client Model]
Remote Provisioning Messages
Provisioning Protocol PDUs
PB-ADV or PB-GATT
PB-Remote Client [Remote Provisioning Server Model]

This diagram shows how remote provisioning is performed (diagram courtesy of the Bluetooth SIG)

Remote provisioning also provides plug-and-play configuration capabilities for complex devices such as those used in large office networks, university campuses, or multi-story hotels. This means you no longer need to reset, re-provision, and reconfigure a complex device if it detects a change to its physical composition. Overall, remote provisioning simplifies the provisioning process, and makes it more efficient, especially if you are working in a large building with a widely distributed mesh network.

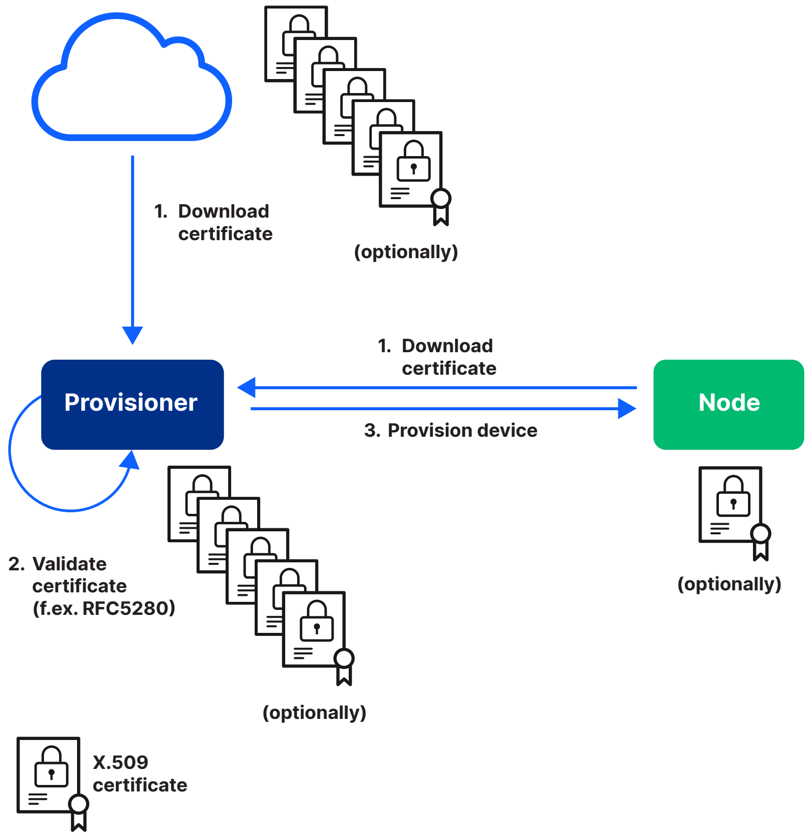
Certificate-Based Provisioning

Another improvement to the provisioning process is the addition of certificate-based provisioning, which should be a familiar approach to most network engineers as it is the same approach used with web server provisioning. With certificate-based provisioning, an X.509 format digital certificate may be provided by device manufacturers or vendors and used in the provisioning process. This digital certificate authenticates the identity of the device and only authenticated devices are allowed to connect to the network.

The new enhancements enable the use of X.509 digital certificates during the provisioning process to verify the authenticity of devices being added to the network. This not only makes device provisioning more secure but also supports bulk commissioning as the provisioner does not need to see or hear the device during the provisioning process. This is especially useful when combined with the new Remote Provisioning feature.

With certificate-based provisioning, an X.509 format digital certificate must be provided by device manufacturers or vendors for each device provisioned to the network. The device certificate contains the device's UUID and its public key. The mesh nodes must contain and securely store a secret private key that matches to the public key in the certificate. This information is then used to authenticate the identity of the device and only authenticated devices are allowed to connect to the network, which prevents unauthorized and counterfeit devices from being provisioned into the network.

The X.509 certificates can be stored in the nodes, in the provisioner or most typically in a cloud where the provisioner has access to.



Device Firmware Updates

Like most hardware, Bluetooth mesh devices run firmware to control the hardware, and this firmware must be kept up to date to minimize security threats to the network. But this is not simple to do in Mesh 1.0 as there is no standardized method to query the firmware on mesh network nodes, nor to roll out new firmware; this a manual process that must be managed by a person. Therefore, performing firmware updates manually is a labor intensive process and security risk because important updates can be delayed, missed, or not deployed to all devices.

Automatic monitoring for new device firmware updates is another mesh feature enhancement. With this functionality, a device is designated to monitor for updates, acquire the binary images for the update and distribute the images throughout the network to update the appropriate nodes and to verify that the update was correctly applied. This is done using multicast firmware distribution capabilities to minimize the number of radio transmissions needed to perform the updates, which makes the process highly efficient and does not consume a lot of power.

The updated specification defines the following roles for automated device firmware updates:

Initiator node – The initiator node identifies the available firmware updates for the nodes in the network. The initiator node is usually a device that supports both Bluetooth mesh and TCP/IP (such as a smartphone or gateway) so that it can periodically scan manufacturers' websites for new firmware. Updates discovered by the initiator node are sent to distributor nodes.

Distributor node - The distributor node receives the binary firmware image from the initiator as well instructions which nodes are to be updated and also distributes the new firmware images from the initiator to the appropriate nodes that need to be updated.

Updating nodes – These are the nodes that receive the firmware images from the distributors

Stand-alone Updater – The Stand-alone Updater acts as both the Initiator node as well the Distributor node. And typically, it is an Internet connected gateway device that is permanently part of a Bluetooth mesh network. Additionally, several new mesh models are defined to carry out a series of procedures in the firmware update process.

Firmware Distribution Client is the model used by the Initiator to send the firmware image and the firmware distribution parameters to the Distributor, and to start the firmware image transfer.

Firmware Distribution Server is the model used by the Distributor to receive from the Initiator the firmware update parameters, the set of Updating nodes to update, and the firmware image to transfer. This model can transfer one firmware image at a time.

Firmware Update Client is the model used by the Distributor and Initiator to manage firmware updates. The Initiator uses this model to retrieve the information about the firmware subsystems installed on the Updating node, and to get the download URIs of the new firmware images. The Distributor uses this model to start a firmware image transfer to the Updating nodes.

Firmware Update Server is the model used by the Updating node to report the firmware images installed on the node and the download URI of new firmware images, and to initiate a firmware update to receive a new firmware image.

BLOB Transfer Client is the model used to transfer binary large objects (BLOBs) over a Bluetooth mesh network. An MBT client can transfer a BLOB to one or more MBT servers, either unicasting or multicasting depending on the situation.

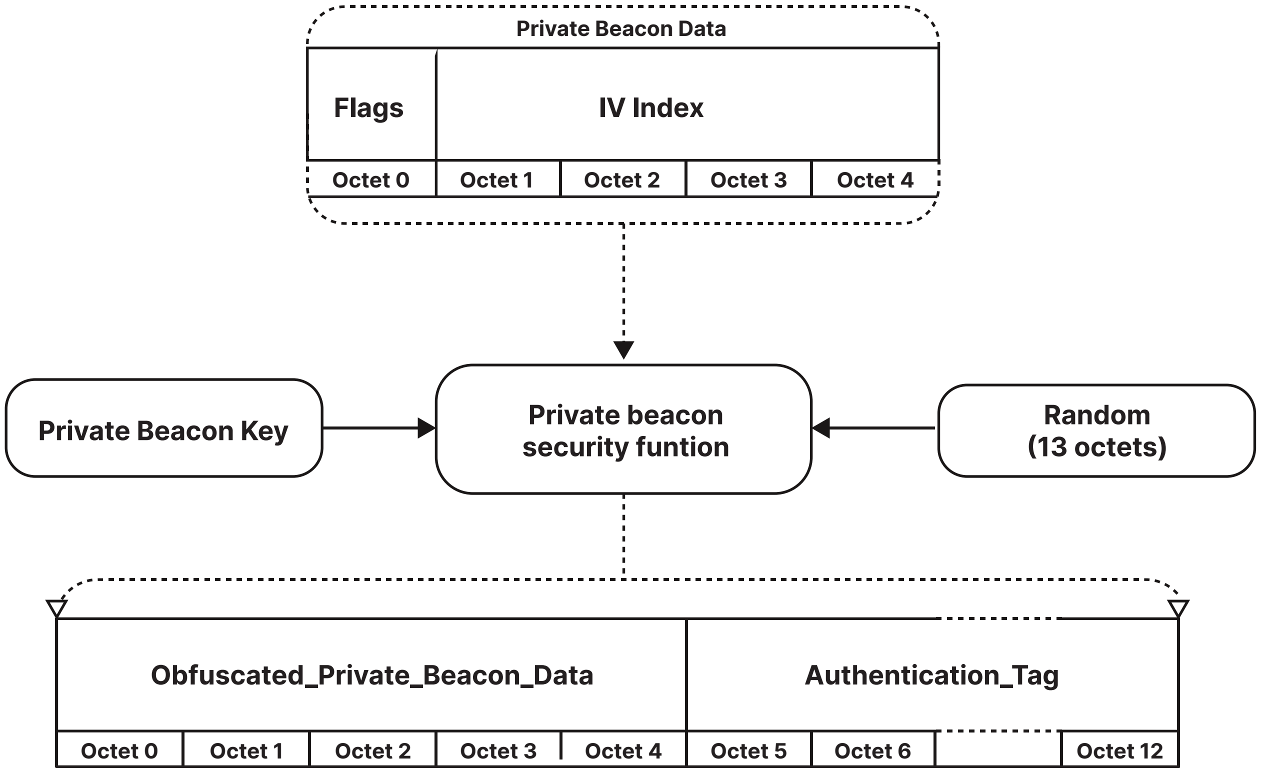
BLOB Transfer Server is the model used to receive BLOBs over a Bluetooth mesh network from an MBT client.

Not part of these mesh enhancements, but a key feature nonetheless, is the mesh Binary Large Object (BLOB) transfer protocol. The BLOB transfer method uses Bluetooth mesh multicast messages to transfer the binary firmware image to Updating nodes to make the firmware transfer as fast as possible. Once the firmware is completely transferred the Distributor will query every node individually for any missing packets and use reliable uncast messaging to transfer the missing packets to each node one-by-one.

Private Beacons

Mesh networks use beaconing, which is when an event triggers a node to transmit information such as sensor data or location or point of interest information. With these enhancements, private beacons improve security and user privacy by eliminating the possibility of static information becoming visible in beacon messages. This improves network security for devices such as wearables because it eliminates the possibility of tracking devices using the information contained in a beacon message.

Private beaconing is achieved by encrypting data using a key called the PrivateBeaconKey and a random 13 octet number. The PrivateBeaconKey is derived from the main network key, which means it can only be decrypted by nodes that are a part of the network. Also, both the random number and the Bluetooth device address (BD_ADDR) in the Bluetooth LE packet now change periodically (by default every 10 minutes).



Additional Minor Enhancements

In addition to the key new features introduced, there are a few notable minor enhancements to the existing features that improve performance, efficiency, and scalability.

On-Demand Private GATT Proxy

Proxy clients can broadcast Solicitation PDUs containing the network identity of the proxy server to signal the proxy server that it should start advertising with Private Network Identity type. The proxy client can then connect to the proxy node. Unlike the behavior of the proxy node in Bluetooth mesh 1.0 where it is continually broadcasting, the proxy node can now stay in passive scanning mode until it receives a signal from the proxy client to advertise.

Segmentation and Reassembly (SAR) Configuration

A new mesh model is added which can be used to configure the SAR behavior to optimize the network operation during its lifetime.

Message Opcode Aggregation

This feature can be used to aggregate multiple Bluetooth mesh messages into a single message to optimize network traffic. This is useful for example during the configuration of mesh nodes which typically includes a lot of configuration messages and acknowledgements being sent over the network.

Large Composition Data Values

Composition Data describes a Bluetooth mesh node, and it’s features such as how many elements it has. Large DCD value enables more complex mesh devices that will have more data in the DCD field.

Health Fault Codes

Bluetooth mesh Health Model fault codes are now described in the assigned numbers document making it easier to request new fault code values.

Models Metadata

Some mesh models have been enhanced to enable the indication of one or more metadata types supported. A model implementation may then retrieve metadata values from the Large Composition Data Server Models Metadata Page 0 state.

Silicon Labs Will Support New Design Considerations for Device Development

When designing Bluetooth mesh devices, the software stack selected for development is critical. In general Silicon Labs makes it easy for device developers to access the software they need through our Simplicity Studio. This is where all the software development kits (SDKs) are available for you to download to your development board. Our latest GSDK not only supports the new Bluetooth mesh features but also supports further security enhancements. These SDKs are now integrated with our Secure Vault Key Management, which means that when deployed to Secure Vault, mesh encryption keys are protected using the Secure Vault Key Management functionality.

Improve Mesh Network Security and Power Efficiency with New Bluetooth Mesh Features

These new features will help device developers produce more secure Bluetooth mesh devices while also making it easier for end users to configure devices and make their networks more efficient and secure. Since we've been following this release for quite some time, we are be prepared to help you simplify development and get your new highly efficient and secure devices to market faster.

Visit silabs.com/bluetooth to learn more about Silicon Labs' product portfolio.