Secure OTA Update with SiLabs Bluetooth SDK 2.7

Create bootloader with secure OTA DFU capability:

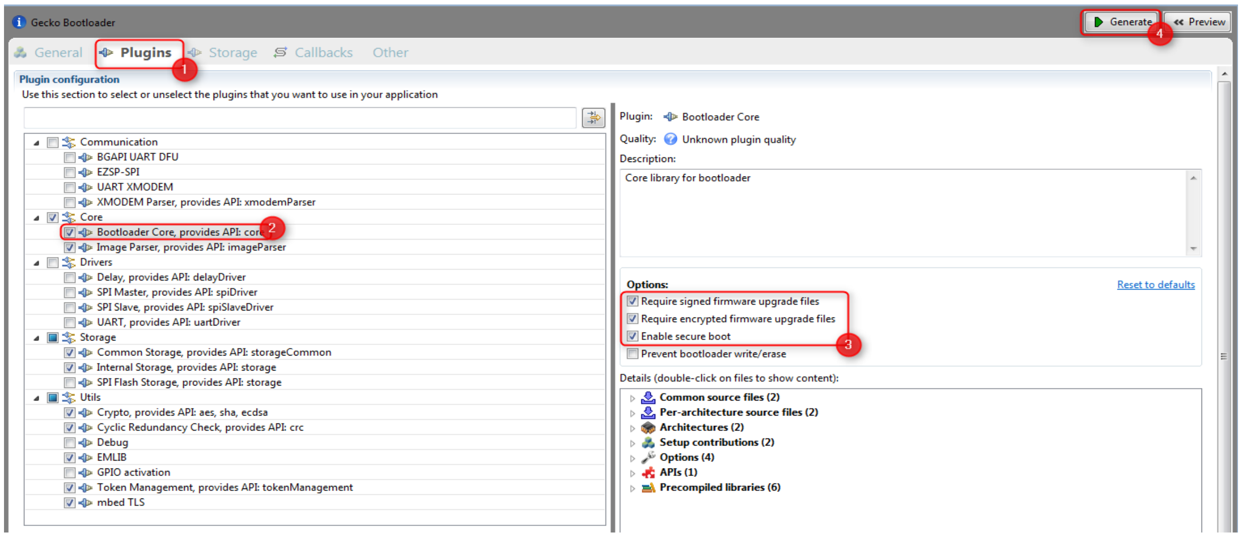
Since Gecko SDK Suite 1.0 / Bluetooth SDK 2.3.0 the Gecko Bootloader is included in the SDK. There are a number of predefined configurations available to help customers easily create new Bootloader projects for different purposes.

For the OTA DFU we need the Bluetooth in-place OTA DFU Bootloader (EFR32BG1 parts) configuration.

1. Open Simplicity Studio, and select your device in the Devices tab
2. Check the Preferred SDK at the top of the main window
3. Click New Project button
4. Select Gecko Bootloader application type, click Next
5. Select the latest installed SDK, click Next
6. Select **Bluetooth in-place OTA DFU Bootloader** sample application, click Next
7. Name your project, click Next
8. Check your device and choose toolchain. Click Finish

After the bootloader project is created, the Application Builder automatically opens up. To add security features

1. Open the Plugins tab
2. Click on Bootloader Core
3. On the right side tick the checkboxes
4. Require singed firmware upgrade files
5. Require encrypted firmware upgrade files
6. Enable secure boot
7. Click Generate in the upper right corner
8. Now you can build your bootloader project.



Plugin configuration
Use this section to select or unselect the plugins that you want to use in your application

- [ ] Communication
  - [ ] BGAPI UART DFU
  - [ ] EZSP-SPI
  - [ ] UART XMODEM
  - [ ] XMODEM Parser, provides API: xmodemParser
- [x] Core
  - [x] Bootloader Core, provides API: core
  - [x] Image Parser, provides API: imageParser
- [ ] Drivers
  - [ ] Delay, provides API: delayDriver
  - [ ] SPI Master, provides API: spiDriver
  - [ ] SPI Slave, provides API: spiSlaveDriver
  - [ ] UART, provides API: uartDriver
- Storage
  - [x] Common Storage, provides API: storageCommon
  - [x] Internal Storage, provides API: storage
  - [ ] SPI Flash Storage, provides API: storage
- Utils
  - [x] Crypto, provides API: aes, sha, ecdsa
  - [x] Cyclic Redundancy Check, provides API: crc
  - [ ] Debug
  - [x] EMLIB
  - [ ] GPIO activation
  - [x] Token Management, provides API: tokenManagement
  - [x] mbed TLS

Plugin: Bootloader Core
Quality: Unknown plugin quality
Description:
Core library for bootloader

Options: Reset to defaults
- [x] Require signed firmware upgrade files
- [x] Require encrypted firmware upgrade files
- [x] Enable secure boot
- [ ] Prevent bootloader write/erase

Details (double-click on files to show content):
- **Common source files (2)**
- **Per-architecture source files (2)**
- **Architectures (2)**
- **Setup contributions (2)**
- **Options (4)**
- **APIs (1)**
- **Precompiled libraries (6)**

Generate and flash security keys

In order to use the security features of the Gecko Bootloader, encryption and signing keys need to be generated. These keys must be then written to the EFR32 device. The encryption key is used with the GBL file for secure firmware update. The signing keys are used both with the GBL file for secure firmware update and to sign the application image for Secure Boot.

You can create security keys using Simplicity Commander in command line (from the path to your bootloader project folder):

1. For ease of use add the Commander path in a temporary variable:

```
Set
commander=C:\SiliconLabs\SimplicityStudio\v4\developer\adapter_packs\commander\commander.exe
```

or add it to the Path system environment variable, if you do not want to repeat this step every time you open up a command line.

2. Generate signing keys with the following command in command line

```
%commander% gbl keygen --type ecc-p256 --outfile app-sign-key.pem
```

This creates three files: app-sign-key.pem, app-sign-key.pem.pub, and app-sign-key.pem-token.txt.

3. Generate encryption key with the following command in command line

```
%commander% gbl keygen --type aes-ccm --outfile app-encrypt-key.txt
```

4. Write the two token files containing the encryption key and public key as manufacturing tokens to the device

```
%commander% flash --tokengroup znet --tokenfile app-encrypt-key.txt --tokenfile app-sign-key.pem-
tokens.txt
```

Create Bluetooth app with secure OTA DFU capability

Since OTA DFU is not fully implemented in the Bootloader, a Bluetooth application has to be created and flashed to the device first along with the Bootloader to support the upgrade. This can be any app that supports restarting the device in DFU mode. The easiest is to use the SoC-Empty software example.

1. Open Simplicity Studio, and select your device in the Devices tab
2. Check the Preferred SDK at the top of the main window
3. Click New Project button
4. Select Bluetooth SDK, click Next
5. Select the latest installed SDK, click Next
6. Select **SOC - Empty** sample application, click Next
7. Name your project, click Next
8. Check your device and choose toolchain. Click Finish
9. Build you project

Since Secure Boot was enabled in the bootloader, only signed application images can be booted. If you flash the image as it is, the application will not start. You have to sign it first. Also, it is very important, that OTA DFU requires the apploader and the user application parts of the image to be signed separately, because they will be upgraded separately!! To get a signed apploader and a signed application image:

1. Copy app-sign-key.pem and app-encrypt-key.txt into the Bluetooth project (SOC_Empty). The simplest way to do this is to open your project in Simplicity Studio and drag and drop the files into the project tree.
2. Run create_bl_files.bat found in your Bluetooth project

This will create application-signed.gbl and apploader-signed.gbl files into the output_gbl folder. These are the signed upgrade files, which contain the signed images. To extract the images from the upgrade files and to merge the apploader image, the application image and the gecko bootloader image into one upload file do the following:

1. Copy the bootloader image that ends with **–combined.s37** from the output folder of your bootloader project (the output folder is named after the compiler) to the output_gbl folder of your Bluetooth project
2. Run the following command in this directory(output_gbl):

```
%commander% convert bootloader-storage-internal-ble_3-combined.s37 apploader-signed.gbl application-signed.gbl --outfile bootloader+apploader+app.hex
```

Now you can flash **bootloader+apploader+app.hex** to the device (right click on the file in Simplicity Studio -> Flash to Device). This file contains the bootloader (first and second stage), the signed apploader image and the signed application image. Upon reset the application starts and the device is ready to be upgraded OTA.

Note: do not erase the chip before flashing the image in order to preserve the security keys already flashed to the device. Or, if you want to erase the device, do not forget to re-flash the security keys.

OTA Image

Now create an image to do the OTA,

1. Edit your earlier project SOC_Empty code, so that a change to the firmware image is seen after the OTA update. (Eg. Add code to turn ON LED0)
2. Build your bluetooth project
3. Run create_bl_files.bat found in your Bluetooth project

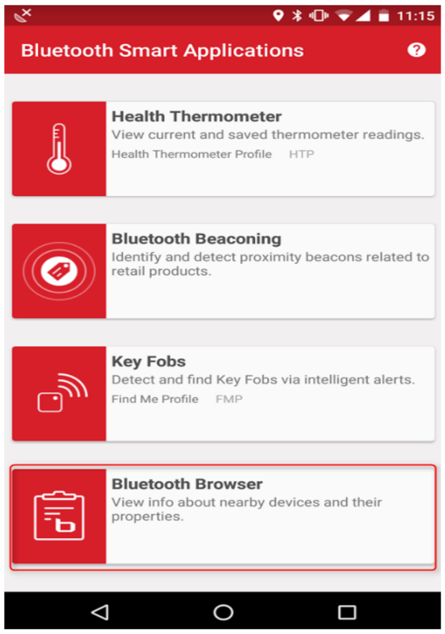
This will create application-signed-encrypted.gbl and apploader-signed-encrypted.gbl files into the output_gbl folder. These are the signed and encrypted upgrade files, which can be sent OTA to the target device.

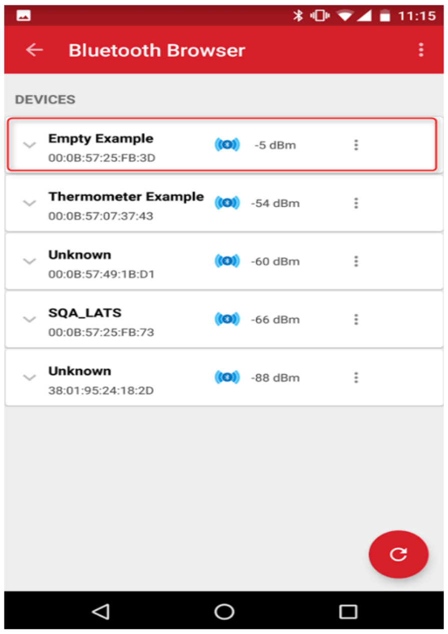
Upgrade your device

Now your device can be upgraded OTA from the basic application to the upgrade application. For this you need another device which will send the new firmware to the target device via Bluetooth. In this example a smartphone will be used for this purpose.

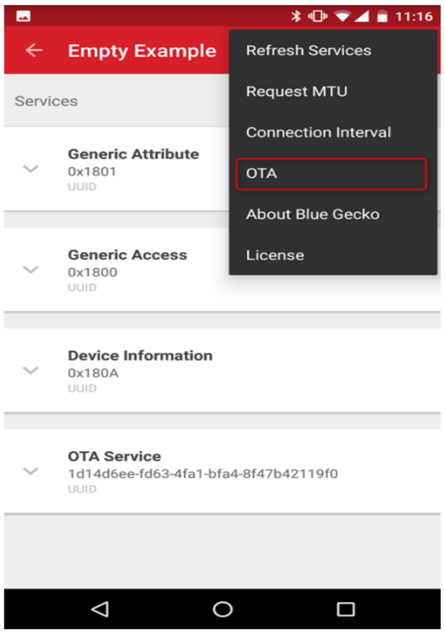
1. Download and install Blue Gecko app (http://silabs.com/bluegeckoapp) to your smartphone
2. Create a new folder under /SiliconLabs_BGApp/OTAFiles/ on your smartphone. E.g. call it my_app
3. Open the output_gbl folder in your project
4. Copy application-signed-encrypted.gbl and apploader-signed-encrypted.gbl from your computer to your smartphone into /SiliconLabs_BGApp/OTAFiles/my_app folder
5. Open Blue Gecko app
6. Select Bluetooth Browser

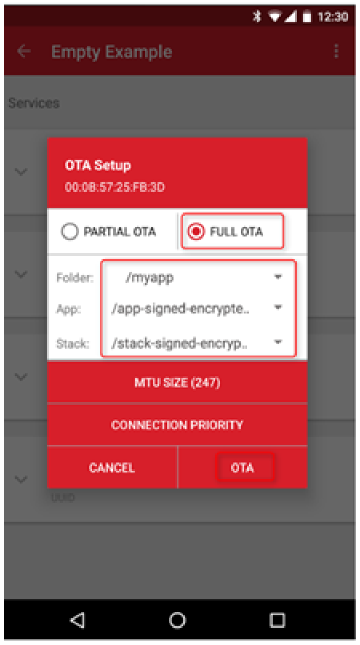
7. Find your device. The basic app advertises itself as Empty Example. Since there might be other devices advertising around you with the same name, place your smartphone close to your device and find the one with the highest RSSI
8. Click on your device to connect to it
9. Open the context menu in the upper right corner and select OTA
10. Choose Full OTA
11. Select my_app folder, then select apploader-signed-encrypted.gbl as stack and application-signed-encrypted.gbl as app
12. Click OTA
13. After upload finished, you should see your device reflecting the changes you did last time.


11:15
Bluetooth Smart Applications
Health Thermometer
View current and saved thermometer readings.
Health Thermometer Profile HTP
Bluetooth Beaconing
Identify and detect proximity beacons related to retail products.
Key Fobs
Detect and find Key Fobs via intelligent alerts.
Find Me Profile FMP
Bluetooth Browser
View info about nearby devices and their properties.



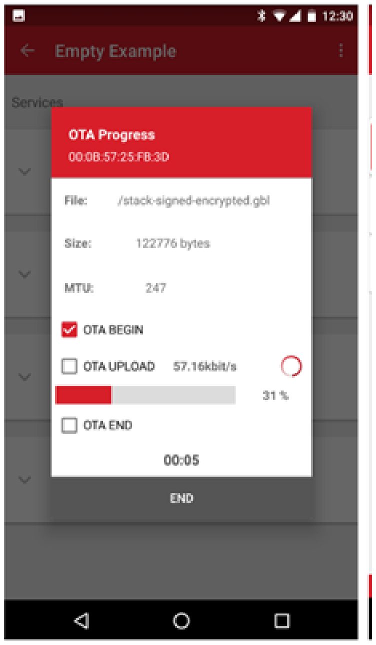
Empty Example	00:0B:57:25:FB:3D	-5 dBm
Thermometer Example	00:0B:57:07:37:43	-54 dBm
Unknown	00:0B:57:49:1B:D1	-60 dBm
SQA_LATS	00:0B:57:25:FB:73	-66 dBm
Unknown	38:01:95:24:18:2D	-88 dBm





PARTIAL OTA [✓ FULL OTA]

Folder: /myapp
App: /app-signed-encrypte..
Stack: /stack-signed-encryp..



File: /stack-signed-encrypted.gbl
Size: 122776 bytes
MTU: 247
[✓] OTA BEGIN
OTA UPLOAD 57.16kbit/s
31 %
OTA END
00:05

References:

1. https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/09/12/secure_ota_dfu-Wb22
2. https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/04/26/upgrading_gecko_boot-IfFz
3. https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/04/12/adding_gecko_bootloa-osqt
4. https://www.silabs.com/documents/login/application-notes/an1086-gecko-bootloader-bluetooth.pdf
5. https://www.silabs.com/documents/public/user-guides/ug266-gecko-bootloader-user-guide.pdf
6. https://www.silabs.com/documents/public/user-guides/ug103-06-fundamentals-bootloading.pdf
7. https://www.silabs.com/documents/login/application-notes/an1045-bt-ota-dfu.pdf