Abstract
This application note describes how to leverage the EdgeLock SE05x ease of use configuration for secure cloud onboarding to Google Cloud IoT core platform. It provides detailed instructions to run the software example provided as part of the support package using an OM-SE050ARD and an MCU board.

Document information
<table>
<thead>
<tr>
<th>Information</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords</td>
<td>EdgeLock SE05x, Google Cloud IoT Core, Secure cloud onboarding</td>
</tr>
<tr>
<td>Abstract</td>
<td>This application note describes how to leverage the EdgeLock SE05x ease of use configuration for secure cloud onboarding to Google Cloud IoT core platform. It provides detailed instructions to run the software example provided as part of the support package using an OM-SE050ARD and an MCU board.</td>
</tr>
</tbody>
</table>
# Revision history

<table>
<thead>
<tr>
<th>Revision number</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2019-06-08</td>
<td>First document release</td>
</tr>
<tr>
<td>1.1</td>
<td>2019-06-21</td>
<td>Added correct reference to AN12396</td>
</tr>
<tr>
<td>1.2</td>
<td>2019-10-15</td>
<td>Added SE050 ease of use configuration</td>
</tr>
<tr>
<td>1.3</td>
<td>2019-11-22</td>
<td>Updated project import from SDK instead of CMake.</td>
</tr>
<tr>
<td>1.4</td>
<td>2020-12-07</td>
<td>Updated to latest template and fixed broken links</td>
</tr>
</tbody>
</table>
Abbreviations

Table 1. Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCP</td>
<td>Google Cloud IoT Core Platform</td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer</td>
</tr>
</tbody>
</table>
1 EdgeLock SE05x ease of use configuration

The IoT device identity should be unique, verifiable and trustworthy so that device registration attempts and any data uploaded to GCP can be trusted by the OEM. GCP verifies the device identity using PKI cryptography. This authentication scheme requires that the associated private key remains secret and hidden from users, software or malicious attackers during the product's lifecycle.

The EdgeLock SE05x security IC is designed to provide a tamper-resistant platform to safely store keys and credentials needed for device authentication and device onboarding to cloud service platforms such as GCP. Using the EdgeLock SE05x security IC, OEMs can safely connect their devices to GCP without writing security code or exposing credentials or keys.

However, key generation and injection into security ICs can introduce vulnerabilities if not done properly. Manual provisioning can lead to errors and is difficult to scale when more devices are needed. Also, to ensure keys are kept safe, injection should take place in a trusted environment, in a facility with security features like tightly controlled access, careful personnel screening, and secure IT systems that protect against cyberattacks and theft of credentials among others.

In order to allow OEMs to get rid of the complexity of key management and to offload the cost of ownership of a PKI infrastructure, the EdgeLock SE05x is offered pre-provisioned for ease of use. This means that OEMs are not required to program additional credentials and can leverage the EdgeLock SE05x ease of use configuration for most of the use cases, including for secure cloud onboarding of their devices to GCP.

Note: For more information about the EdgeLock SE05x ease of use configuration, please refer to AN12436 - SE050 configurations.
2 Leveraging EdgeLock SE05x ease of use configuration for GCP

GCP uses private / public key pairs to authenticate devices into their platform. This authentication scheme requires that the associated private key remains secret and hidden from users, software or malicious attackers during the product’s lifecycle. In addition, the process of secure provisioning private / public key pairs into IoT devices throughout the OEM’s supply-chain is challenging, requiring placing dedicated teams and secure hardware in remote manufacturing sites, resulting in slower time to market and increased costs.

The EdgeLock SE05x is offered off-the-shelf pre-provisioned so that OEMs are not required to program any additional credentials to onboard their devices to GCP. On the one hand, EdgeLock SE05x provides a tamper-resistant platform to safely store the private key needed for device authentication and registration to GCP service. On the other hand, the public key or the device certificate can be read out from the EdgeLock SE05x (e.g. at manufacturing time) and installed on the GCP platform.

Figure 1 illustrates the device registration flow to GCP leveraging the EdgeLock SE05x ease of use configuration:

1. NXP delivers to the OEM’s device manufacturer a quantity of EdgeLock SE05x ICs based on a purchase order. The EdgeLock SE05x samples come pre-provisioned with die-individual credentials.
2. The OEM’s device manufacturer assembles the EdgeLock SE05x ICs and deploys the software into the final IoT devices. It also needs to take care to read out the device public key from the EdgeLock SE05x samples.
3. The OEM, as the system operator, manages the GCP account and registers on it every device by uploading its public key (Optionally, the device certificate can be used instead of the public key).
4. The OEM ships IoT devices to end customers.
5. IoT devices boot up and automatically connect to GCP service using the private key pre-provisioned inside EdgeLock SE05x ease of use configuration.

Figure 1. GCP device registration flow with EdgeLock SE05x ease of use configuration

Disclaimer: The described device registration flow spans multiple roles given the various entities involved. How each roles is mapped in the registration flow might be scenario-dependent for each OEM.
3 Running the GCP device onboarding project example

The GCP project example included in the EdgeLock SE05x Plug & Trust Middleware is an illustrative software example that showcases how to leverage EdgeLock SE05x security IC to set up trusted connections to GCP cloud.

This section explains the steps required to run the GCP demo leveraging the EdgeLock SE05x ease of use configuration. We also use the FRDM-K64F board as an example, but the same steps can be replicated using any the MCU/MPUs supported by the EdgeLock SE05x Plug & Trust Middleware.

On the other hand, if you prefer to generate and inject your own credentials in EdgeLock SE05x for the GCP demo, please refer to Section 4. It explains how to use the provisioning scripts included as part of EdgeLock SE05x Plug & Trust Middleware for that purpose.

Note: The GCP device onboarding procedure described in this section and the EdgeLock SE05x Plug & Trust Middleware GCP demo example are provided only for evaluation purposes. Therefore, the subsequent procedure must be adapted and adjusted accordingly for a commercial deployment.

3.1 Hardware required

This guide provides detailed instructions to the GCP project example using the hardware described below. However, you could use other MCU / MPU boards supported by EdgeLock SE05x Plug & Trust Middleware for this purpose as well.

1. OM-SE050ARD development kit:

   Table 2. OM-SE050ARD development kit details

<table>
<thead>
<tr>
<th>Part number</th>
<th>12NC</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>OM-SE050ARD</td>
<td>935383282598</td>
<td>EdgeLock SE050 development board</td>
</tr>
</tbody>
</table>

2. FRDM-K64F board:

   Table 3. FRDM-K64F details

<table>
<thead>
<tr>
<th>Part number</th>
<th>12NC</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRDM-64F</td>
<td>935326293598</td>
<td>Freedom development platform for Kinetis K64, K63 and K24 MCUs</td>
</tr>
</tbody>
</table>
3.2 Read out public key from EdgeLock SE05x ease of use configuration

GCP requires devices to register before they can connect to their cloud platform. The registration process consists of creating a logical device instance and uploading on it the public key of the device (or optionally, the device certificate). The device public key can be directly read from the EdgeLock SE05x ease of use configuration.

Table 4 shows the ECC256 key pair we will use for GCP device onboarding. This ECC256 key pair has been selected as an example, for a complete detail of the EdgeLock SE05x ease of use configuration, refer to AN12436 - SE050 configurations.

Table 4. ECC256 public key used for GCP device onboarding

<table>
<thead>
<tr>
<th>Key name and type</th>
<th>Certificate</th>
<th>Usage policy</th>
<th>Erasable by customer</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud connection key 0, ECC256, Die Individual</td>
<td>Cloud Connectivity Certificate 0, ECC signed</td>
<td>Default</td>
<td>Yes</td>
<td>Key: 0xF0000100 Cert: 0xF0000101</td>
</tr>
</tbody>
</table>

This section explains how to read out the public key from the EdgeLock SE05x using a FRDM-K64F board as a host platform.

**Note:** Check AN12396- Quick start guide to Kinetis K64 for detailed instructions on how to bring up the FRDM-K64F board.

3.2.1 Download EdgeLock SE05x Plug & Trust Middleware

Follow these steps to download the EdgeLock SE05x Plug & Trust Middleware in your local machine:

1. Download EdgeLock SE05x Plug & Trust Middleware from the NXP website
2. Create a folder called se05x_middleware in C: directory as shown in Figure 2:

![Figure 2. Create se05x_middleware folder](image)
3. Unzip the EdgeLock SE05x Plug & Trust Middleware inside the se05x_middleware folder. After unzipping, you will see a folder called simw-top created. The contents of the simw-top directory should look as they appear in Figure 3:

![Figure 3. Unzip se05x middleware](image)

Note: It is recommended to keep se05x_middleware with the shortest path possible and without spaces in it. This avoids some issues that could appear when building the middleware if the path contains spaces.

3.2.2 Flash FRDM-K64F with VCOM software

The VCOM software allows the FRDM-K64F board to be used as a bridge between the Windows machine and the EdgeLock SE05x and enables the execution of the EdgeLock SE05x ssscli tool and other utilities from the laptop. To flash the VCOM software into the FRDM-K64F, follow these steps:

1. Unplug and plug again the USB cable to the openSDA USB port as shown in Figure 4:

![Figure 4. Unplug and plug OpenSDA port](image)
2. When you plug the board, your laptop should recognize the board as an external drive as shown in Figure 5:

![Figure 5. FRDM-K64F drive](image1)

3. Flash the VCOM software to FRDM-K64F. The VCOM software binary can be found in the EdgeLock SE05x Plug & Trust Middleware package, inside the `simw-top\binaries` folder as shown in Figure 6:

![Figure 6. VCOM binary folder](image2)
4. Drag and drop or copy and paste the a7x_vcom-TloI2C-frdmk64f-SE050x.bin file into the FRDM-K64F drive from your computer file explorer as shown in Figure 7:

![Figure 7. Drag and drop VCOM binary](image)

5. The serial and VCOM ports should be recognized by your Device Manager. To check that the ports are recognized, follow the steps indicated in Figure 8:
   a. Unplug the USB cable from the OpenSDA USB port.
   b. Plug the USB cable to the OpenSDA USB port.
   c. Check that the serial port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as **USB Serial Device (COM7)** but this naming might change depending on your computer. Therefore, it is important that you identify which device is recognized at the moment you plug the SDA USB port to the computer.
   d. Plug the USB cable to the K64F USB port.
   e. Check that the VCOM port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as **Virtual Com Port (COM8)** but this naming might change depending on your computer (e.g. It could also appear named as...
USB Serial Device). Therefore, it is important that you identify which device is recognized at the moment you plug the K64F USB port to the computer.

![Diagram showing setup](image)

**Figure 8. Check VCOM and serial ports**

*Note:* Please note that it is possible that either of the two COM ports is not detected when using low-quality or charge-only USB cables.

### 3.2.3 Read public key using `sssci` tool

To read the public key using the `sssci` tool, follow these steps:

1. Mount OM-SE050ARD on top of the FRDM-K64F. Then, connect FRDM-K64F OpenSDA port and K64F port to your laptop as shown in **Figure 9**

![Diagram showing connections](image)

**Figure 9. Connect boards**
2. Create a folder to store the extracted key and start the `ssscli` tool by sending the commands shown in Figure 10:
   a. Go to `simw-top\binaries\pySSSCLI` folder:
      Send: ```cd se050 Middleware\simw-top\binaries\pySSSCLI```  
   b. Create a folder to store the extracted key:
      Send: ```mkdir data```  
   c. Check your VCOM port number in your Device Manager. Open the connection using the `ssscli`:
      Send: ```ssscli connect se050 vcom <COM_NUMBER>```  
   d. Send the reset command:
      Send: ```ssscli se05x reset```  

![Figure 10. Start ssscli tool](image)

**Note:** If you see the following message: WARNING: sss.connect: Session already open, close current session first message as shown in Figure 11, it means that you have a session open. To close it, send: (1) > ssscli disconnect and then send once again (2) > ssscli connect se050 vcom <COM_NUMBER> and later (3) > ssscli se05x reset.

![Figure 11. Close an already opened session](image)
3. Read the public key from the EdgeLock SE05x and save it with the name `cloud_ecc_key.pem` as shown in Figure 12:
   
   Send: `>ssscli get ecc pair 0xF0000100 data\cloud_ecc_key.pem`

   **Note:** `0xF000100` is the identifier of the pre-configured ECC256 key pair that we are using in this example (EdgeLock SE05x ease of use configuration).

   ![Figure 12. Extract public ECC key](image)

4. The extracted public key can be found in the `data` folder as shown in Figure 12:

   ![Figure 13. Extract public ECC key](image)

5. Close the connection as shown in Figure 14:
   
   Send: `>ssscli disconnect`

   ![Figure 14. Disconnect ssscli](image)

If you have completed this section, go to Section 3.3.

### 3.3 Prepare GCP cloud platform

This section describes how to get started with GCP using the so-called Google Cloud Platform Console (GCP Console). The GCP console is a web dashboard offering a friendly user interface. This chapter describes how to:

- Create a GCP account.
- Create a project.
• Enable billing option.
• Create a registry.
• Create a device.

To perform the above steps, GCP supports the GCP Console, the API and gcloud. For the sake of simplicity, this application note only uses the GCP Console. For details on how to perform any of these steps using the API or gcloud, refer to Cloud IoT Core documentation.

**Note:** The GCP account preparation procedure is the same independent of the MCU / MPU platform you choose for evaluation purposes.

### 3.3.1 Create a GCP account

GCP offers a 12-month free trial for new accounts. To create your GCP account:

1. Go to [https://cloud.google.com/iot-core/](https://cloud.google.com/iot-core/) and click the **Try it free** button (Figure 15):

   ![Create a free account in Cloud IoT Core](image)

   **Figure 15.** Create a free account in Cloud IoT Core
2. Sign-in with your Google account. If you do not have one, you need to create one beforehand (Figure 16):

![Figure 16. Sign in to your Google account](image)

3. Select (1) your country, (2) accept the terms of service, and (3) click **Agree and continue** button (Figure 17):

![Figure 17. Create a free account in Cloud IoT Core (II)](image)
4. Fill in your customer information (Figure 18)

![Image of customer information fill-in fields]

Figure 18. Create a free account in Cloud IoT Core (III)
5. Supply a valid credit or debit card and click **Start my free trial** button as shown in Figure 19. You will not be charged, GCP uses your credit card to verify your identity. When the free trial ends, you will need to manually upgrade your paid account.

![Create a free account in Cloud IoT Core (IV)](image)

### 3.3.2 Create a project

A project called *My First Project* is created by default when the GCP account is created. You can use this project or you can create a new one. To create a new project:
1. Go to the blue ribbon on the top and click My First Project. In the Select a project form, click New Project on the top right hand side as shown in Figure 20.

![Figure 20. Create a project](image-url)
2. Type your project name and click **Create** as shown in Figure 21.

![Google Cloud Platform interface with project creation options](image)

**Figure 21. Create a project**

3. Wait a few seconds until the project creation is completed. To select the new project, go to the blue ribbon on the top and click **My First Project**. In the **Select a project** form, click your project from the list as shown in Figure 22.

![Select a project interface with project options](image)

**Figure 22. Select the new project**
3.3.3 Enable billing option

After selecting the project, you might be asked to enable Cloud IoT API. The Cloud IoT API must be enabled before you can view Cloud IoT in the console. To enable Cloud IoT API:

1. Click **Enable API** as shown in Figure 23

![Enable Cloud IoT API](image)

**Figure 23. Enable Cloud IoT API**

3.3.4 Create a registry

After enabling the API, you will be asked to **create a device registry**. A device registry is a container of devices, it belongs to a cloud project and is created in a specific cloud region. To create a registry:
1. Click on **Create a device registry** button as shown in [Figure 24](#).

![Figure 24. Create device registry](image)

2. Fill in the following information for your device registry creation as shown in:
   a. **Registry ID**: Type the ID for the registry. This identifier is permanent.
   b. **Region**: Choose the cloud region where data will be stored for devices in this registry.
   c. **Protocol**: Select the protocols that your devices will use to connect to Cloud IoT Core (MQTT / HTTP).
   d. **Default telemetry topic**: Topics are aggregators that allow you to send and receive messages between the devices and Cloud IoT Core. Device events will be published to this topic by default. Select a default telemetry topic or create a new one. To create a new topic, select **create a topic**. In the create a topic dialog, enter your topic name in the **Name** field.
   e. **Device state topic**: This field is optional. You can define a topic where devices will publish updates in regards to device status or configuration changes.
   f. **CA certificate**: This field is optional. You can upload, or enter manually, a CA certificate in .pem format. Adding a CA certificate will enable Cloud IoT core to
verify the chain of trust of device certificates against this registry-level CA. In this case, device certificates need to be signed by this CA certificate.

g. **Stackdriver logging:** Choose which device activity events are sent to Stackdriver Logging. Device activity logs include information such as device connections and errors.

![Stackdriver logging configuration](image)

**Figure 25. Create registry**

### 3.3.5 Create a device

GCP requires devices to register before they can connect to the cloud. The registration process consists of adding a device to the GCP registry we created in Section 3.3.4.
and uploading the device public key as well as defining other properties. To add a new device:

1. On the left menu, click on Devices button as shown in Figure 26:

![Figure 26. Create a device](image)

2. On the top menu, click on Create a device as shown in Figure 27:

![Figure 27. Create a device II](image)
3. Fill in the device creation form with the following information:
   a. *Device ID*: Type an ID for your device. This field cannot be changed later.
   b. *Device communication*: Select whether you allow or block device communication with the cloud.
   c. *Authentication*: Select ES256 and copy paste the device public key *(cloud_ecc_key.pem)* extracted in Section 3.2.
   d. *Public key expiry date*: You can set an expiration date for the key. If the key expires, the device will not be able to connect to GCP.
   e. *Device metadata*: Use this field to add optional device metadata, such as a serial number.
   f. *Stackdrive logging*: Choose which device activity events are sent to Stackdriver Logging. Device activity logs include information such as device connections and errors.
   g. Click on *Create* button.
Figure 28. Create a device form
4. Check whether the new device is registered in the registry dashboard as shown in Figure 29:

![Figure 29. View the new device in the registry dashboard](image)

3.4 GCP project execution

To execute the GCP project example, we need to:

- Download and install FRDM-K64F SDK.
- Import GCP project example.
- Change GCP project account settings.
- Run GCP project example.

**Note:** Before running the GCP demo example, you need to have installed MCUXpresso IDE and FRDM-K64F SDK in your local environment and imported the GCP project example. Check AN12396- Quick start guide to Kinetis K64 for detailed instructions on:

- How to install MCUXpresso.
- How to obtain FRDM-K64F SDK.
- How to import FRDM-K64F project examples.

3.4.1 Download and install the FRDM-K64F SDK

The GCP device onboarding project example is included as part of the FRDM-K64F SDK. Install it to your MCUXpresso workspace as shown in Figure 30:

1. Download the FRDM-K64F SDK, publicly available from the NXP website.
2. Drag and drop the FRDM-K64F SDK zip file in the Installed SDKs section in the bottom part of the MCUXpresso IDE.
3. Check that the FRDM-K64F SDK is installed successfully.

![Figure 30. Import FRDM-K64F SDK](image)

3.4.2 Import GCP project example

The FRDM-K64F SDK includes a project examples called se_SE050x_cloud_gcp. Import it to your MCUXpresso workspace as shown in:

1. Click *Import SDK examples* from the MCUXpresso IDE quick start panel.
2. Select *se_SE050x_cloud_gcp* project example and click the *Finish* button.
3. Check the project is now visible in your MCUXpresso workspace

*Note: For detailed instructions on how to import project examples from FRDM-K64F SDK, check [AN12396 - Quick start guide with Kinetis K64F](https://www.nxp.com/docs/en/application-note/AN12396.pdf)*.
3.4.3 Change GCP project account settings

We need to change a few variables in the MCUXpresso GCP demo related with your GCP project account settings we created in Section 3.3, in the MCUXpresso workspace:
1. Go to `frdmk64f_se_SE05x_cloud_gcp/source` folder and open the `gcp_iot_config.h` file as shown in Figure 32.

![Figure 32. Open gcp_iot_config.h file](image1)

2. Replace the `#define GCP_PROJECT_NAME` variable with your GCP project ID created in Section 3.3.2 as shown Figure 33.

![Figure 33. Change GCP_PROJECT_NAME variable](image2)
3. Replace the `#define GCP_LOCATION_NAME` variable with your GCP registry region chosen Section 3.3.4 in as shown Figure 34:

![Figure 34. Change GCP_LOCATION_NAME variable](image)

4. Replace the `#define GCP_REGISTRY_NAME` variable with your GCP registry name created in Section 3.3.4 as shown Figure 35:

![Figure 35. Change GCP_REGISTRY_NAME variable](image)

5. Replace the `#define GCP_DEVICE_NAME` variable with your GCP device name created in Section 3.3.5 as shown Figure 36:

![Figure 36. Change GCP_DEVICE_NAME variable](image)

6. Replace the `#define SSS_KEYPAIR_INDEX_CLIENT_PRIVATE` variable with the ID of the key pair we are using to connect to GCP (0xF0000100) and the `#define`
**SSS_CERTIFICATE_INDEX** with the ID of the associated certificate (0xF0000101) as shown in Figure 37:

*Note: the key and certificate IDs must be the same used in Public key extraction using FRDM-K64F.*

**7. Save changes.**

**3.4.4 Run GCP project example**

To run the GCP demo, follow these steps:

1. Connect FRDM-K64F OpenSDA port and K64F port to your laptop, and connect the board to Internet using an Ethernet cable as shown in Figure 38:
2. Open TeraTerm, go to Setup > Serial Port and configure the terminal to 115200 baud rate, 8 data bits, no parity and 1 stop bit and click OK as shown in:

![Configure TeraTerm](image1.png)

Figure 39. Configure TeraTerm

3. Go to the MCUXpresso Quickstart Panel and click Debug button, wait a few seconds until the project executes and click on Resume to allow the software to continue its execution as shown in Figure 40.

![Debug GCP project](image2.png)

Figure 40. Debug GCP project
4. **(Figure 41)** Your device should now be connected to GCP. Check that your device is connected by:
   a. Checking the TeraTerm logs.
   b. Checking that the last time the device was seen in the GCP dashboard matches with the current time.

![Figure 41. Device is connected to GCP](image-url)
4 Appendix: Key generation with EdgeLock SE05x Plug & Trust Middleware provisioning scripts

This section explains how to generate and inject your own credentials in EdgeLock SE05x using the provisioning scripts included as part of EdgeLock SE05x Plug & Trust Middleware. Please, use this procedure only if you prefer to generate your own keys instead of leveraging the EdgeLock SE05x ease of use configuration.

*Note:* The key generation and injection procedure described in this section is only applicable for evaluation or testing purposes. In a commercial deployment, key provisioning must take place in a trusted environment, in a facility with security features such as tightly controlled access, careful personnel screening, and secure IT systems that protect against cyberattacks and theft of credentials.

4.1 Flash FRDM-K64F with VCOM software

Before running the EdgeLock SE05x Plug & Trust Middleware provisioning scripts, we need to flash the VCOM software into the FRDM-K64F board. To do so, follow the steps detailed in Section 3.2.2.

4.2 Running GCP key provisioning script

To run the GCP provisioning script, follow these steps:

1. Open a command prompt and go to the `C:\se050_middleware\simw-top\binaries\pySSSCLI` as shown in Figure 42:

   ```
   > cd C:\se050_middleware\simw-top\binaries\pySSSCLI
   ```

   ![Figure 42. Navigate to ssscli folder](image)

   **Figure 42. Navigate to ssscli folder**
2. Run the `Provision_GCP.exe` executable as shown in Figure 43:
   Send `Provision_GCP.exe <VCOM_NUMBER>`

![Figure 43. Generate and inject keys using the Provision_GCP.exe script](image)

3. The key pair and the certificates should have been injected in the EdgeLock SE05x and a copy of the credentials should have been created in the `gcp` folder as shown in Figure 44:

![Figure 44. List of provisioned keys and certificates](image)

4.3 Publish device certificate in GCP

After provisioning the EdgeLock SE05x, we need to register the device certificate in GCP. Go to the device creation menu in GCP and register the device certificate as shown in Figure 45:

1. Select ES256_X509 format
2. Copy in the *public key value* field the content of the `<device_Uid>_device_certificate.cer` file generated in Section 4.2

![Google Cloud Platform](image)

**Figure 45.** Publish device certificate in GCP

### 4.4 Change GCP project settings

The key identifiers used are different for the EdgeLock SE05x ease of use configuration and for the keys we have generated with the EdgeLock SE05x Plug & Trust Middleware GCP provisioning script. Therefore, the last step is to update the key identifiers in the MCUXpresso project as shown in Figure 46.

1. Replace the other MCUXpresso project account settings as described in Section 3.4.3.
2. Replace the `#define SSS_KEYPAIR_INDEX_CLIENT_PRIVATE` variable with the ID of the injected key pair (0x20181001).
3. Replace the `#define SSS_CERTIFICATE_INDEX` with the ID of the injected certificate (0x20181002).

![Figure 46. Update key identifiers in the MCUXpresso project](image)
5 Legal information

5.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

5.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.
Tables

Tab. 1. Abbreviations .............................................3
Tab. 2. OM-SE050ARD development kit details ............6
Tab. 3. FRDM-K64F details ......................................6
Tab. 4. ECC256 public key used for GCP device onboarding ..................................................7
Figures

Fig. 1. GCP device registration flow with EdgeLock SE05x ease of use configuration .............................. 5
Fig. 2. Create se050_middleware folder ........................................ 7
Fig. 3. Unzip se05x middleware .................................................. 8
Fig. 4. Unplug and plug OpenSDA port ....................................... 8
Fig. 5. FRDM-K64F drive ......................................................... 9
Fig. 6. VCOM binary folder ...................................................... 9
Fig. 7. Drag and drop VCOM binary ............................................ 10
Fig. 8. Check VCOM and serial ports .......................................... 11
Fig. 9. Connect boards ........................................................... 11
Fig. 10. Start ssscli tool ......................................................... 12
Fig. 11. Close an already opened session ...................... 12
Fig. 12. Extract public ECC key .............................................. 13
Fig. 13. Extract public ECC key .............................................. 13
Fig. 14. Disconnect ssscli ....................................................... 13
Fig. 15. Create a free account in Cloud IoT Core ............. 14
Fig. 16. Sign in to your Google account ................................. 15
Fig. 17. Create a free account in Cloud IoT Core (II) .......... 15
Fig. 18. Create a free account in Cloud IoT Core (III) ...... 16
Fig. 19. Create a free account in Cloud IoT Core (IV) .... 17
Fig. 20. Create a project ......................................................... 18
Fig. 21. Create a project II ..................................................... 19
Fig. 22. Select the new project ............................................... 19
Fig. 23. Enable Cloud IoT API ............................................... 20
Fig. 24. Create device registry ............................................ 21
Fig. 25. Create registry .......................................................... 22
Fig. 26. Create a device .......................................................... 23
Fig. 27. Create a device II ...................................................... 23
Fig. 28. Create a device form ................................................. 25
Fig. 29. View the new device in the registry dashboard .......... 26
Fig. 30. Import FRDM-K64F SDK ........................................... 27
Fig. 31. Import GCP project in the workspace .................. 28
Fig. 32. Open gcp_iot_config.h file ...................................... 29
Fig. 33. Change GCP_PROJECT_NAME variable ................... 29
Fig. 34. Change GCP_LOCATION_NAME variable ............. 30
Fig. 35. Change GCP_REGISTRY_NAME variable ............. 30
Fig. 36. Change GCP_DEVICE_NAME variable .................. 30
Fig. 37. Change SSS_KEYPAIR_INDEX_CLIENT_PRIVATE and SSS_CERTIFICATE_INDEX variables ......................... 31
Fig. 38. Connect FRDM-K64F board ..................................... 31
Fig. 39. Configure TeraTerm ................................................. 32
Fig. 40. Debug GCP project .................................................. 32
Fig. 41. Device is connected to GCP .................................... 33
Fig. 42. Navigate to ssscli folder ........................................... 34
Fig. 43. Generate and inject keys using the Provision_GCP.exe script .......................................................... 35
Fig. 44. List of provisioned keys and certificates ............. 35
Fig. 45. Publish device certificate in GCP ......................... 36
Fig. 46. Update key identifiers in the MCUXpresso project .... 37
Contents

1  EdgeLock SE05x ease of use configuration ................................................. 4
2  Leveraging EdgeLock SE05x ease of use configuration for GCP ...................... 5
3  Running the GCP device onboarding project example ..................................... 6
   3.1  Hardware required .............................................................................. 6
   3.2  Read out public key from EdgeLock SE05x ease of use configuration ............ 7
   3.2.1  Download EdgeLock SE05x Plug & Trust Middleware .......................... 7
   3.2.2  Flash FRDM-K64F with VCOM software ....................................... 8
   3.2.3  Read public key using ssscli tool ............................................. 11
   3.3  Prepare GCP cloud platform ....................................................... 13
   3.3.1  Create a GCP account ......................................................... 14
   3.3.2  Create a project ................................................................. 17
   3.3.3  Enable billing option ........................................................... 20
   3.3.4  Create a registry ................................................................. 20
   3.3.5  Create a device .................................................................... 22
   3.4  GCP project execution .................................................................... 26
   3.4.1  Download and install the FRDM-K64F SDK ................................. 26
   3.4.2  Import GCP project example ............................................... 27
   3.4.3  Change GCP project account settings .................................... 28
   3.4.4  Run GCP project example ................................................... 31
4  Appendix: Key generation with EdgeLock SE05x Plug & Trust Middleware
   provisioning scripts ........................................................................... 34
   4.1  Flash FRDM-K64F with VCOM software ....................................... 34
   4.2  Running GCP key provisioning script .......................................... 34
   4.3  Publish device certificate in GCP .............................................. 35
   4.4  Change GCP project settings .................................................... 36
5  Legal information ................................................................................ 38

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section ‘Legal information’.

© NXP B.V. 2020. All rights reserved.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 7 December 2020
Document identifier: AN12401
Document number: 534913