Renesas RA Family
Installing and Updating Secure Keys

Introduction
The Secure Crypto Engine 9 (SCE9) on Renesas RA Family MCUs can operate in two different modes, called Compatibility Mode and Protected Mode. SCE9 supports secure key installation and update when the crypto engine is used in Protected Mode. Protected Mode provides optimum protection against security attacks by providing SPA/DPA resistance and secure key installation and update, with a usage model that enforces secure best practices key handling. This solution enables creating an unbroken chain of trust from the delivery of the device from Renesas all the way to the end product. For the SCE9 operational modes definitions and use cases, refer to the Application Note Secure Crypto Engine Operational Modes listed in the References section.

This application note provides the secure key installation and secure key procedures. In addition, use cases for secure key installation and update are discussed, and an example walk-through of installing and updating an AES-256 key is provided, with a corresponding software example project to demonstrate the key update operation using the FSP Crypto API.

The example software project provided in the application project is designed to be used with the EK-RA6M4. Follow the instructions in section 3, section 4, and section 5 to install the initial AES-256 key, install a Key-Update Key, and encrypt a new AES-256 key using the Key-Update Key prior to exercising the example project.

The key installation procedure described in this application note applies to any RA Family MCUs with the SCE9 crypto engine (for example, RA4M2, RA4M3, RA6M4, RA6M5). The example project can be easily adapted to the above-mentioned RA Family MCUs.

Required Resources
Development tools and software
- The e² studio ISDE v2021-10 or greater
- Renesas Flexible Software Package (FSP) v3.5.0 or later
- SEGGER J-Link® USB driver and RTTViewer
- Renesas Flash Programmer (RFP) v3.08 or later
- Renesas Security Key Management Tool v1.0.0 or later

The FSP, J-Link USB drivers, and e² studio are bundled in a downloadable platform installer available on the FSP webpage at renesas.com/ra/fsp. SEGGER RTTViewer is available for download free-of-charge from https://www.segger.com/products/debug-probes/j-link/tools/rtt-viewer/. RFP is available for download from https://www.renesas.com/software-tool/renesas-flash-programmer-programming-gui. The “FREE-OF-CHARGE EDITION” can be used for the functionality required by this Application Project. The Security Key Management Too can be downloaded at https://www.renesas.com/software-tool/security-key-management-tool.

Hardware
- EK-RA6M4, Evaluation Kit for RA6M4 MCU Group (http://www.renesas.com/ra/ek-ra6m4)
- Workstation running Windows® 10
- One USB device cable (type-A male to micro-B male)

Prerequisites and Intended Audience
This application note assumes you have some experience with the Renesas e² studio IDE and Arm®-TrustZone®-technologybased development models with e² studio. In addition, the application note assumes that you have some knowledge of RA Family MCU security features. See chapter 49, Security Features in the Renesas RA6M4 Group MCU User’s Manual: Hardware for background knowledge preparation for the cryptographic key installation.
The intended audience are product developers, product manufacturers, product support, or end users who are involved with any stage of installing the secure keys or utilizing the installed secure keys with Renesas RA Family MCUs.

Contents

1. SCE Wrapped Key Creates Root of Trust ................................................................. 3
   1.1 Introduction to Root of Trust ......................................................................................... 3
   1.2 Introduction to Secure Crypto Engine 9 and Associated Keys ........................................ 3
      1.2.1 Secure Crypto Engine (SCE9) ......................................................................................... 3
      1.2.2 SCE9 Associated Keys ................................................................................................. 4
   1.3 Renesas Secure Key Installation Advantages .............................................................. 4
      1.3.1 Advantages of Key Wrapping over Key Encryption .................................................. 5
      1.3.2 Advantages of Key Wrapping using MCU HUK ....................................................... 6
   1.4 Renesas RA MCU Factory Boot Firmware Limitations .................................................. 6

2. Wrapped Key Installation Use Cases and Installation Procedure Overview ................. 7
   2.1 Wrapped Key Types ....................................................................................................... 7
   2.2 General Steps for Secure Key Installation ................................................................. 7
   2.3 General Steps for Secure Key Update ......................................................................... 8
   2.4 Tools Used in the Secure Key Installation and Update ............................................... 9
   2.5 Example Project Operational Flow ............................................................................. 9

3. Wrap a User Factory Programming Key Using the Renesas Key Wrap Service ............ 10
   3.1 PGP Key Exchange between Customer and Renesas DLM Server ................................ 10
      3.1.1 Create PGP Key Pair ................................................................................................... 10
      3.1.2 Registration with DLM Server ..................................................................................... 13
      3.1.3 Exchange Customer and Renesas PGP Public Keys ................................................... 15
   3.2 Wrapping the User Factory Programming Key ............................................................. 20
      3.2.1 Renesas Security Key Management Tool .................................................................. 20
      3.2.2 Creating the User Factory Programming Key ........................................................... 20
      3.2.3 Wrapping the UFPK .................................................................................................... 21

4. Preparing Keys for Installation and Update Using Renesas Security Key Management Tool ... 27
   4.1 Encrypt a User Key with the UFPK .............................................................................. 27
   4.2 Encrypt a Key-Update Key with the UFPK ................................................................. 28
   4.3 Encrypt a New User Key with the KUK ....................................................................... 29

5. Secure Key Installation via Serial Programming Interface ........................................ 30
   5.1 Setting up the Hardware .............................................................................................. 30
   5.2 Install the User Key and Key-Update Key via the Serial Programming Interface ............ 31

6. Example Project ........................................................................................................... 33
   6.1 Overview ....................................................................................................................... 33
   6.2 Using the RFP Installed Keys ....................................................................................... 35
1. **SCE Wrapped Key Creates Root of Trust**

1.1 **Introduction to Root of Trust**

Roots of trust are highly reliable hardware, firmware, and software components that perform specific, critical security functions (https://csrc.nist.gov/projects/hardware-roots-of-trust). In an IoT system, a root of trust typically consists of identity and cryptographic keys rooted in the hardware of a device. It establishes a unique, immutable, and unclonable identity to authorize a device to exist in the IoT network.

Secure boot is part of the services provided in the Root of Trust in many security systems. Authentication of the application uses Public Key Encryption. The associated keys are part of the Root of Trust of the system. Device Identity, which consists of Device Private Key and Device Certificate, is part of the Root of Trust for many IoT devices.

From the above Root of Trust discussion, we can see that leakage of cryptographic keys can bring the secure system into a risky state. Protection of the Root of Trust involves limiting key accessibility to within the cryptographic boundary only, with keys that are securely stored and preferably unclonable. The Root of Trust should be locked from read and write access by unauthorized parties.

The Renesas user key management system can provide all the above desired protection.

1.2 **Introduction to Secure Crypto Engine 9 and Associated Keys**

1.2.1 **Secure Crypto Engine (SCE9)**

The Renesas RA Secure Crypto Engine 9 exists in the Renesas RA Cortex-M33 based MCUs, which also include Arm® TrustZone® technology for enhanced security.
The Secure Crypto Engine 9 is an isolated subsystem within the MCU. The crypto engine contains hardware accelerators for both symmetric and asymmetric cryptographic algorithms, as well as various hashes and message authentication codes. It also contains a True Random Number Generator (TRNG), providing an entropy source for the cryptographic operations. The Secure Crypto Engine is protected by an Access Management Circuit, which can shut down the crypto engine in the event of an illegal external access attempt.

1.2.2 SCE9 Associated Keys

![Diagram of security keys]

RA Family MCUs with the SCE9 support several security keys as shown in Figure 2.

The first is an MCU-unique Hardware Unique Key (HUK), a 256-bit random key that is preprogrammed in the Renesas factory. This key is stored in unmapped flash, accessible only by the SCE9, not by application code. It is further protected by being stored not in plaintext, but rather wrapped by the HRK (Hardware Root Key) and MCU unique ID. The SCE9 can access this HUK to perform user key wrapping.

Since the HUK is stored in a wrapped format unique to the MCU, even if an attacker were able to extract the stored key, another MCU won’t be able to use it.

The DLM Keys are associated with the Renesas Device Lifecycle Management (DLM) system. For the installation and usage of the DLM Keys, please reference application note [Renesas RA Family Device Lifecycle Management Key Installation Application Note (R11AN0469)](#).

The Key-Update Keys (KUKs) are used for updating the User Keys when a device is deployed in the field.

- “User Keys” refer to the keys used in various cryptographic operations performed by the user application.
- The KUKs are installed during end-product manufacturing via the MCU’s programming interface. To install a new key to a device that is deployed in the field, the new key must be encrypted with one of the installed KUKs. In addition to replacing keys that have been compromised, many security policies require key rotation or key update (re-keying) on a regular basis. It is also recommended to consider installing multiple KUKs.

1.3 Renesas Secure Key Installation Advantages

Secure key installation and update, combined with the crypto engine’s support of wrapped keys, address many vulnerabilities associated with using plaintext keys:
• Plaintext keys are never stored in code flash. In the event of a program memory breach, the sensitive key material is protected.
• Plaintext keys are never stored in RAM. In the event of malicious code executing on the system, the sensitive key material is still protected.
• Keys can be securely stored in code flash, data flash, or even copied into external memory, enabling unlimited secure key storage.

In addition, Renesas key wrapping techniques protect against device cloning, as discussed below.

1.3.1 Advantages of Key Wrapping over Key Encryption

![Figure 3. Key Wrapping vs. Key Encryption](image)

It is important to understand the difference between wrapping and encrypting for secure asset storage. When data is encrypted and sent to another recipient, if that recipient has the same key, they can decrypt the data. This results in a confidential exchange of information. However, what if there was a problem with the transmission of the encrypted data? If the recipient unknowingly receives corrupted information, the decryption algorithm will generate garbage data, with no indication that the original data has been corrupted.

Wrapping solves this problem for us by appending a Message Authentication Code to the encrypted output for integrity checking.
1.3.2 Advantages of Key Wrapping using MCU HUK

Using the MCU Hardware Unique Key to wrap the stored keys adds another protection feature – clone protection.

If the wrapped key is transmitted or copied to another MCU, that MCU’s HUK will not be able to either unwrap nor use the copied key. Even if the entire MCU contents are copied onto another device, the keys cannot be used nor exposed.

1.4 Renesas RA MCU Factory Boot Firmware Limitations

Secure Key installation via the serial programming interface is not supported for RSA 3K, RSA 4K, ECC secp256k1, and Key-Update Keys on some older versions of the Renesas RA MCUs due to factory Boot Firmware limitations. The user needs to use a Renesas Flash Programmer (RFP) to read out the Boot Firmware version and confirm the support for the Secure Key Installation of the above-mentioned keys. Reference the RFP user’s manual section Flow of Operations to access the Bootloader Firmware version by using the Read Device Information menu.

- V1.2.04 – WS1: secure user key install command is not supported
- V1.3.10 – WS2: user key install command is not supported
- V1.5.22 – CS: user key install command is supported, but it does not support RSA 3K, RSA 4K, secp256k1, nor KUK
- V1.6.25 and above – MP: no limitations

The part information silkscreened on the device can also be checked, though it is recommended to confirm the boot firmware version as described above. Boot firmware limitations exist for the following MCUs:

- RA4M2 - All WS and ES devices
- RA4M3 - All WS, ES and CS devices (date code 014AZ00)
- RA6M4 - All WS, ES and CS devices (date code 014AZ00), MP device with date codes 028AZ00, 031AZ00
- RA6M5 - All WS and ES devices

Please note that some EK-RA6M4 and EK-RA4M3 Evaluation Kits may contain affected silicon. The following list shows the affected kit serial numbers. Note that all early adopter kits with WS or ES silicon are also affected.

- EK-RA4M3 – Serial numbers 219243 – 219542
- EK-RA6M4 – Serial numbers 215938 – 216237 and 218497 - 218996

If your application requires secure key installation of RSA 3K, RSA 4K, ECC secp256k1, or Key-Update Keys and your evaluation kit does not support it, please contact your local Renesas Sales representative.
2. Wrapped Key Installation Use Cases and Installation Procedure Overview

This section provides an overview of the wrapped key installation use cases and the general steps for installation procedure of each use case. A step-by-step walk through of the wrapped key installation procedures is provided in later sections.

2.1 Wrapped Key Types

Table 1 below summarizes the key types that can be directly installed into Renesas RA Family MCUs with the SCE9 Secure Crypto Engine. Installed keys are stored wrapped by the MCU's HUK.

<table>
<thead>
<tr>
<th>Lifecycle Transition Keys</th>
<th>SECDBG_KEY, NONSECDBG_KEY, RMA_KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>AES-128, AES-192, AES-256</td>
</tr>
<tr>
<td>RSA</td>
<td>RSA-1024, RSA-2048, RSA-3072, RSA-4096 (Public and Private)</td>
</tr>
<tr>
<td>ECC</td>
<td>secp192r1 (NIST P-192), secp224r1 (NIST P-224) (Public and Private)</td>
</tr>
<tr>
<td></td>
<td>secp256r1 (NIST P-256), secp384r1 (NIST P-384) (Public and Private)</td>
</tr>
<tr>
<td></td>
<td>secp256k1 (Public and Private)</td>
</tr>
<tr>
<td></td>
<td>Brainpool P256r1, P384r1, and P512r1 (Public and Private)</td>
</tr>
<tr>
<td>HMAC</td>
<td>HMAC-SHA224, HMAC-SHA256</td>
</tr>
<tr>
<td>Utility Keys</td>
<td>Key-Update Keys</td>
</tr>
</tbody>
</table>

This Application Project provides an example of AES-256 key installation and update.

2.2 General Steps for Secure Key Installation

“Key Installation” refers to installation of keys via the serial programing interface, demonstrated here with the Renesas Flash Programmer (RFP). All the key types described in section 2.1 can be installed through this process. Key preparation steps where key material is exposed in plaintext must be performed in a secure environment.

There are three high-level steps for key installation. The step-by-step walkthroughs of how to perform these steps are described in sections 3, 4, and 5 respectively.

1. The first step in the secure key installation process is to use the Renesas Device Lifecycle Management (DLM) service to wrap an arbitrary User Factory Programming Key (UFPK) using the Renesas Hardware Root Key (HRK). The UFPK is a 256-bit value selected by the user. The same UFPK can be used to install any number of keys.

2. Next, the user key must be encrypted with UFPK.

**Figure 5. Wrapping the UFPK using DLM server**

**Figure 6. Encrypt the User Key with UFPK**
3. Finally, the user key is installed via the serial programming interface. The input to the serial programming interface includes both the wrapped UFPK (W-UFPK) and the encrypted user key prepared in the previous steps.

![Figure 7. Install User Key over the Serial Programming Interface]

2.3 General Steps for Secure Key Update

To enable secure key installation in the field, one or more Key-Update Keys (KUK) must be installed during production programming/provisioning. Since installing new keys in the field is usually done to replace older keys (key rotation or re-keying), this process is also referred to as “key update”.

KUKs, like other cryptographic keys, can be stored in either code flash or data flash (if available on the MCU). Since the KUK is the only mechanism by which new keys can be installed/wrapped, it is highly recommended that multiple KUKs be installed during production provisioning. This enables the KUK to be rotated or revoked to adhere to an infrastructure security policy or to respond to a key exposure security breach.

Note that additional KUKs CANNOT be installed after the programming interface is disabled. Once a product is in the field with its programming interface disabled, new keys can ONLY be installed via a pre-existing KUK.

The KUKs may be stored in any code or data flash location during production. Users can install multiple KUKs and provide a scheme to rotate the keys based on a timed schedule or key leakage event. We recommend that users disable the programming interface prior to deploying to the field for security considerations. This location will be passed to the FSP Crypto key update API for the installation of the new user key.

The KUK can be used to install new keys in the field.

There are three steps high-level steps for key update.

1. The first step is to install a KUK, as described in General Steps for Secure Key Installation.
2. The second step is to use the KUK to encrypt the new user key with the KUK.

![Figure 8. Encrypt the New User Key with a KUK]
3. The last step is to use FSP and the previously installed KUK to install the new user key. The Renesas RA Flexible Software Package provides an `r_sce_protected` module that supports Protected Mode usage of the Secure Crypto Engine. This module is supported on all RA Family MCUs with SCE9. Review the FSP User’s Manual section Secure Crypto Engine (`r_sce_protected`) when developing applications with the SCE Protected Mode.

![Figure 9. Update the User Key](image)

### 2.4 Tools Used in the Secure Key Installation and Update

There are three tools used in the secure key installation and update besides using e² studio as the software project development environment. Reference the corresponding section mentioned below for details on obtaining, setting up and using these tools.

- **Gpg4win**
  This tool is used in section 3 to generate customer PGP key pair, perform key exchange with Renesas DLM server, and wrap the User Factory Programming Key.

- **Renesas Security Key Management Tool**
  This tool is used in section 3 and section 4 to generate below three key files:
  - User key
  - Key update key
  - New user key encrypted with the key update key

- **Renesas Flash Programmer (RFP)**
  This tool is used in section 5 to install the user key and key update key

### 2.5 Example Project Operational Flow

The example project in this application project demonstrates the secure key update capability for Renesas RA Family MCUs. Prior to running the example project, the user needs to follow sections 3, 4, and 5 to make the following preparations:

- Create an AES-256 user key and install it to the MCU
- Create a Key-Update Key and install it to the MCU
- Create a second AES-256 key and encrypt it with the Key-Update Key

The follow graphic shows the flow of this preparation work plus the example project. The block outlined in red is the scope of the functionality of the example project.
### 3. Wrap a User Factory Programming Key Using the Renesas Key Wrap Service

This section walks you through creating and wrapping a UFPK with the Renesas Key Wrap Service on the Renesas DLM server. Key material is exchanged with the Renesas Key Wrap Service using PGP encryption.

#### 3.1 PGP Key Exchange between Customer and Renesas DLM Server

##### 3.1.1 Create PGP Key Pair

If you already have a PGP key pair, that key can be used for the key exchange process. Otherwise, the instructions below describe one method for creating a PGP key pair.

The PGP software demonstrated here is GPG4Win, which can be downloaded from this URL: [http://www.gpg4win.org/](http://www.gpg4win.org/)

The screen shots included in this application note are based on `gpg4win-3.1.14.exe`. There may be minor graphic interface updates with later versions. However, the functionality used in this application note should persist.
Download and install Kleopatra:

![Image of Kleopatra setup]

**Figure 11. Download and Install Kleopatra**

Launch Kleopatra and create a PGP Key Pair.

1. Click **File->New Key Pair**
2. Choose **Create a personal OpenPGP key pair**.

![Image of key pair creation]

**Figure 12. Create a Personal Open PGP Key Pair**

3. Provide a Name and Email. **Note that even though these are marked as optional, at least one entity must be provided to move to the next stage. Check Protect the generated key with a passphrase.**

![Image of key pair creation]

**Figure 13. Provide Name and Email**
4. Click **Advanced Settings** and select **RSA** as the key type.

![Figure 14. Select RSA Encryption](image)

5. Click **Create** and provide a passphrase twice to protect the private key. Then click **OK**. **Be sure to save your passphrase.**

![Figure 15. Define a Passphrase](image)
6. The PGP key pair should be created successfully. Click **Finish**.

![Figure 16. PGP Key Pair Created](image)

7. A new item will be created in the Kleopatra. Right-click on the keypair just created and select **Export**.

![Figure 17. Export the Customer PGP Public Key](image)

8. Save the public key to a file with an `*.asc` extension. In this example, this file is renamed to `customer_public.asc`. Click **Save**.

![Figure 18. Save the PGP public key to a folder](image)

### 3.1.2 Registration with DLM Server

If you have already registered with the Renesas DLM Server, this section can be skipped. Otherwise, follow the steps below to create login credentials for the Renesas DLM Server.
1. Open the URL https://dlm.renesas.com/keywrap in a browser and click **New registration**.

![Login screen of Key Wrap service](image)

**Figure 19.  Start Registration with Renesas DLM Server**

2. Follow the prompt to provide a **valid** email address and click **Send mail**.

![Figure 20.  Register Customer Email Address](image)

After clicking **Send mail**, the following screen will appear. Click **Return**.

**Figure 21.  Acknowledge Email Transmission**
3. You should receive an email similar to the one shown below. Click on the URL provided to confirm your registration.

```
Dear customer,

Thank you for registering with the KeyWrap service.
To start using this service, you need to click on the following URL to register your information.

https://dtm.renesas.com/keywrap/?menu=reg%2Fregreg%2Fregreg&param=sd%2Fbf%2Fcb000700jiLBN%2520imi5X0htfuH9mP%252F1gQ1aH8%2520

* Notes
Please register your information within three hours of receiving this e-mail.
The URL expires after the three hours, after which you will need to start the registration process anew.

Please delete this email if you were not aware that you were going to receive it.

* This email was sent from a send-only address.
Please understand that there will be no response, even if you reply to this address.
* If you have forgotten your password, reset your password via the link "If you have forgotten your password..." on the login page of this system.

Thank you.
Renesas Electronics Corporation
```

**Figure 22. Registration Confirmation Email**

4. Follow the prompts to provide your name and company name, and create a password. Click the **Next (confirmation)** button. Note that the password must consist of 8 to 32 alphanumeric characters and may include the symbols “!” and “@”.

![Registration Form](image)

**Figure 23. Confirm Registration**

After the confirmation screen is displayed, click on the **Register** button to complete the user registration.

![Registration Confirmation](image)

**Figure 24. Finish the Registration**

### 3.1.3 Exchange Customer and Renesas PGP Public Keys

If you have not already exchanged PGP keys with the Renesas DLM server, follow the steps below.
1. After successfully registering the user information, the following screen will open. Click the **Start service** button to start using the key encryption system.

![Start DLM Key Wrapping Service](image1)

**Figure 25.** Start DLM Key Wrapping Service

2. When the agreement warning shows up, scroll down to the bottom of the **Trusted Secure IP Key Wrap Agreement** and click **I agree**. You will then be logged into the DLM server. Note that the Agreement will come up every time you log into the DLM server.

--- CAUTION!! ---

**PLEASE READ THE FOLLOWING BEFORE USING THE SERVICE**

This Trusted Secure IP Key Wrap Service Agreement (this "Agreement") is between you and Renesas Electronics Corporation. Please carefully note that this Agreement is legally valid agreement relating to Trusted Secure IP key encryption (the "Service").

**Article 15 (ENTIRE AGREEMENT)**

This Agreement sets forth the entire agreement of the parties with respect to the subject matter hereof and supersedes any prior or contemporaneous agreements, written or oral, concerning the subject matter hereof. Any change, modification or amendment of the terms of this Agreement shall not be effective unless reduced to writing and authorized by both parties.

![I agree](image2)

**Figure 26.** Agreement for Using the Renesas DLM Server

3. Upon logging into the DLM system, the window below will appear. Click **PGP key exchange**.

![PGP key exchange](image3)

**Figure 27.** Start PGP Key Exchange
4. Click **Reference** and select the public key generated earlier (**customer_public.asc**). Notice that the fingerprint of the Renesas PGP public key is displayed. This will be used to certify the Renesas public key after you receive it.

```
4. Click **Reference** and select the public key generated earlier (**customer_public.asc**). Notice that the fingerprint of the Renesas PGP public key is displayed. This will be used to certify the Renesas public key after you receive it.
```

<table>
<thead>
<tr>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select your PGP public key that exported format, and click on “PGP key exchange” button. Your public key will be sent to Renesas, and the PGP public key of Renesas will be sent to your e-mail address.</td>
</tr>
<tr>
<td>The fingerprint of PGP public key of Renesas is below.</td>
</tr>
<tr>
<td>FB18 EB66 1F81 20E9 9613</td>
</tr>
<tr>
<td>8DF7 F517 1E9C 1E4B E36D</td>
</tr>
</tbody>
</table>

**Figure 28. Browse the Customer PGP Public Key**

5. Click **PGP key exchange**.

```
5. Click **PGP key exchange**.
```

<table>
<thead>
<tr>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select your PGP public key that exported format, and click on “PGP key exchange” button. Your public key will be sent to Renesas, and the PGP public key of Renesas will be sent to your e-mail address.</td>
</tr>
<tr>
<td>The fingerprint of PGP public key of Renesas is below.</td>
</tr>
<tr>
<td>FB18 EB66 1F81 20E9 9613</td>
</tr>
<tr>
<td>8DF7 F517 1E9C 1E4B E36D</td>
</tr>
</tbody>
</table>

**Figure 29. Browse the customer PGP public key**

6. Once the PGP public key is submitted, click **Return**.

```
6. Once the PGP public key is submitted, click **Return**.
```

<table>
<thead>
<tr>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your PGP public key submit is being processed in your application for registration. After completion of registration, registration-completion e-mail to which the PGP public key of Renesas is attached will be sent. Please wait for a while.</td>
</tr>
</tbody>
</table>

**Figure 30. Wait for Renesas’s PGP Public Key**
7. You will receive an email from Renesas at the email address registered with the DLM server with the contents as shown below if the key exchange is successful. It typically takes about one to two minutes to receive this email.
   Note that a PGP public key can be registered any number of times. The latest PGP public key that has been registered successfully is used for encryption. All previously registered PGP public keys are discarded.

   ![Receive the Renesas PGP Public Key](image)

   **Figure 31. Receive the Renesas PGP Public Key**

   Save the Renesas PGP public key file (**keywrap-pub.key**).

8. Go back to the Kleopatra application and import the Renesas PGP Public key to Kleopatra as shown below.

   ![Import Renesas Public Key](image)

   **Figure 32. Import Renesas Public Key**

9. After **Open** is clicked, a new item is added in Kleopatra as **not certified**.

   ![Renesas Public Key is Imported](image)

   **Figure 33. Renesas Public Key is Imported**
10. Confirm that the Fingerprint displayed is same as what is shown on the screen represented in Figure 29. Click **Certify**.

![Figure 34. Confirm the Fingerprint and Certify the Renesas Public Key](image)

11. Click **Certify** again from following screen.

![Figure 35. Certify the Certificate](image)
12. Provide the passphrase to unlock the secure key.

![Figure 36. Provide the Passphrase](image)

13. The following item will pop up upon successful certification. Click **OK**.

![Figure 37. Successful Certification](image)

3.2 Wrapping the User Factory Programming Key

If you do not already have a wrapped UFPK, follow the steps below to wrap a UFPK with the Renesas Hardware Root Key as described by Figure 5.

3.2.1 Renesas Security Key Management Tool

Prior to wrapping the UFPK with HRK, you need to convert the UFPK to a binary format. This can be achieved by using the Renesas Security Key Management Tool.


Download `SecurityKeyManagementTool_CLI_V100.zip` from the above link. Unzip to reveal `\SecurityKeyManagementTool_CLI_V100\Windows\skmt.exe`. The user’s manual of this tool is located under `\SecureKeyManagementTool_CLI_V100\`. We recommend that you read through the user’s manual before proceeding to the following section.

![Figure 38. User's Manual for the Security Key Management Tool](image)

3.2.2 Creating the User Factory Programming Key

Open a Command Prompt window and navigate to the folder where `skmt.exe` resides.
Use the following command to generate a random UFPK and place it in a key file (`ufpk.key`). If desired, a complete file name with path may be specified. Refer to section 2.5 of the Security Key Management Tool user’s manual to understand the usage of `/genufpk` option.

```
skmt.exe /genufpk /output "C:\User key installation protected mode\keys\ufpk.key"
```

This command will generate a random 256-bit UFPK as shown below. Note that the real key is not shown but has been substituted with dummy digits.

```
UFPK: 6666666677777755555544444444111111112222222233333333888888888
Output File: C:\User key installation protected mode\keys\ufpk.key
```

**Figure 39. Create a Random UFPK Using skmt.exe**

It is also possible to specify a specific UFPK, as shown by the following command:

```
skmt.exe /genufpk /ufpk "000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F" /output "C:\User key installation protected mode\keys\ufpk.key"
```

```
UFPK: 000102030405060708090A0A0C0D0E0E0E000102030405060708090A0B0C0D0E0F
Output File: C:\User key installation protected mode\keys\ufpk.key
```

**Figure 40. Create a Known UFPK Using skmt.exe**

Either command can be used for the purpose of exercising the example project.

### 3.2.3 Wrapping the UFPK

Follow the steps below to wrap the UFPK using Renesas HRK.

1. Encrypt the UFPK with the Renesas public key, which has been imported to Kleopatra. This is in preparation for sending the UFPK to DLM Server for wrapping. Select `Sign/Encrypt...` from Kleopatra and select to encrypt the `ufpk.key` file generated above. Then click `Open`.

**Figure 41. Encrypt the UFPK File for PGP Transfer**
2. When asked which entity this file is to be encrypted for, uncheck **Encrypt for me** and check **Sign as**, **Encrypt for others**, and **Encrypt / Sign each file separately**.

![Figure 42. Select PGP Encryption Options](image)

3. Click the **Open Selection Dialog** (the icon). This will open a **Certificate Selection** dialog box.

![Figure 43. Open the Selection Dialog](image)
4. In this window, select **keywrap** to select the Renesas public key, then click **OK**.

![Figure 44. Select the Renesas PGP public key](image)

5. Ensure that the correct destination folder for the encrypted key is selected under **Output**. Finally, click **Sign/Encrypt**.

![Figure 45. Encrypt UFPK using PGP-encrypt the UFPK](image)

6. If you do not check **Encrypt for me**, you will get an **Encrypt-To-Self Warning** that you cannot decrypt the data. Click **Continue**.

![Figure 46. Start the UFPK Encryption process](image)
7. Provide your private key passphrase, then click **OK**.

![Figure 47. Provide Passphrase](image)

8. The UFPK encrypted with the Renesas public key will be generated, with `.gpg` added to the extension of the key. In this case, the file `ufpk.key.gpg` is generated. Click **Finish**.

![Figure 48. Provide Customer Passphrase](image)

9. Now we can send the UFPK Encrypted with Renesas Public Key to the Renesas DLM Server for wrapping.

   Return to the DLM Server web page, select the **RA** Family and click **RA6M4 Encryption of customer’s data**.

![Figure 49. Select the MCU Family](image)
Next choose the RA6M4 as the MCU Group.

**Figure 50. Select the MCU Group**

10. Click **Encryption service for products** at the next screen.

**Figure 51. Choose Encryption service for products**

11. Click **Reference**, select the `ufpk.key.gpg` created previously, and click **Open**. Note that in the DLM server description, **Key2** refers to the UFPK.

**Figure 52. Select the PGP-Encrypted UFPK file**

12. Click **Settle**. The following message will be printed. Then click **Return to menu**. You can now log out of the Renesas Server.

**Figure 53. Return to the DLM Server Main Menu**
13. The wrapped UFPK Key (W-UFPK) encrypted with your PGP public key should arrive in your email typically in about 1-2 minutes. Save the attached file.

![Figure 54. Receiving the W-UFPK via Email](image)

14. With the Kleopatra program, click Decrypt/Verify, select the W-UFPK file, click Open.

![Figure 55. Decrypt the W-UFPK](image)
15. Follow the prompt to provide your PGP private key passphrase and click **OK**.

![Figure 56. Decrypting the Encrypted W-UFPK](image)

16. Save the decrypted W-UFPK key file to the same folder as the UFPK key file. Both key files are required to generate key installation bundles.

### 4. Preparing Keys for Installation and Update Using Renesas Security Key Management Tool

This section walks the user through the key encryption process required for secure key installation and update. Step-by-step instructions for generating the three types of keys are provided.

- **User Key** encryption with the UFPK for secure key installation
- **Key-Update Key** encryption with the UFPK for secure key installation
- **User Key** encryption with the KUK for secure key update

The `/genkey` command of the Security Key Management Tool will be used to encrypt keys for secure installation and update. These are the options for this command:

- `/keytype` - This input can take either ASCII or a one-byte hexadecimal input parameter indicating the key type. Valid values for this parameter are shown in the various tables in the “key type options” section of the Security Key Management Tool user’s manual.
- `/ufpk` - The encryption key for preparing keys for secure key installation (both user keys and KUKs).
- `/wufpk` - The Renesas HRK wrapped UFPK.
- `/kuk` - The encryption key for encrypting a new user key to update the previously installed user key.
- `/mcu` - The Security Key Management Tool supports RE, RX, RZ, RA, and Synergy. You can select the `/mcu` option setting based on “Table mcu options” of the Security Key Management Tool user’s manual.
- `/output` - Specify the output of the command. You can refer to the “filetype options” section of the Secure Key Management user’s manual to understand the various output files.

This application project uses an AES-256 key type to illustrate the secure key installation and update process.

#### 4.1 Encrypt a User Key with the UFPK

With the UFPK and W-UFPK generated, we can now use `skmt.exe` with the UFPK, the user key, and the W-UFPK to generate the key file to securely install the user key with RFP.

In the Command Prompt window opened earlier (section 3.2.2), use the following command to create the AES-256 key installation file (**AES256.rkey**). Refer to “section 2.6 genkey command options” of the Security Key Management Tool user manual for more information on how to construct the command.
Renesas RA Family Installing and Updating Secure Keys

```plaintext
skmt.exe /genkey /ufpk file="C:\User key installation protected mode\keys\ufpk.key" /wufpk file="C:\User key installation protected mode\keys\ufpk.key_enc.key" /mcu "RA-SCE9" /keytype "AES-256" /key 
9999999999999999888888888888888877777777777777776666666666666666 /filetype 
"rfp" /output "C:\User key installation protected mode\keys\AES256.rkey"
```

Note that in this example:

- We are using 9999999999999999888888888888888877777777777777776666666666666666 as the AES-256 plaintext user key.
- We have used the key type “AES-256”. The user can also use "07" in place of "AES-256" based on “Table User Key AES” in the Security Key Management Tool user's manual.
- "RA-SCE9" is used for the /mcu option.
- We are using a randomly generated IV. The IV is updated in each encryption instance.
- In this example, we have specified the complete file path for the key file AES256.rkey.
- In the output below, the result of W-UFPK, IV and the Encrypted Key are substituted with dummy data.

Output File: C:\User key installation protected mode\keys\AES256.rkey

| UFPK: | 661B1DEE69137C92F22B487A88F2B735963ADEFEDFCDB42C419C2EE96378D115 |
| W-UFPK: | 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF |
| IV: | 1234567890ABCDEF1234567890ABCDEF |
| Encrypted key: | 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF |

Figure 57. Create the AES-256 User Key Installation File

The generated key file AES256.rkey now contains the encrypted user key along with the W-UFPK. The plaintext AES-256 key and UFPK are NOT contained in the *.rkey file, enabling confidential transfer of the key installation file.

4.2 Encrypt a Key-Update Key with the UFPK

With the UFPK and W-UFPK generated, we can now use skmt.exe with the UFPK, a KUK, and the W-UFPK to generate the key file to securely install the KUK with RFP.

In the command line window opened earlier (section 3.2.2), use the following command to create the KUK installation file.

```plaintext
skmt.exe /genkey /ufpk file="C:\User key installation protected mode\keys\ufpk.key" /wufpk file="C:\User key installation protected mode\keys\ufpk.key_enc.key" /mcu "RA-SCE9" /keytype "key-update-key" /key 
000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f /filetype 
"rfp" /output "C:\User key installation protected mode\keys\KUK.rkey"
```

Note that in this example:

- We are using 000102030405060708090a0b0c0d0e0f000102030405060708090a0b0c0d0e0f as the plaintext KUK.
- For the option /keytype, we have used the key type ASCII value “key-update-key” based on “Table Key Update Key” of the Security Key Management Tool user's manual. You can also use “FF” for this option.
- We are using a randomly generated IV. The IV is updated in each encryption instance.
- In this example, we have specified complete file path for the key file (KUK.rkey).
- In the output below, the result of W-UFPK, IV, and the Encrypted Key are substituted with dummy data.
Figure 58. Create the Key-Update Key Installation File

The generated key file KUK.rkey now contains the encrypted KUK along with the W-UFPK. The plaintext KUK and UFPK are NOT contained in the *.rkey file, enabling confidential transfer of the key installation file.

### 4.3 Encrypt a New User Key with the KUK

Preparing a new user key for secure key update involves using the KUK instead of the UFPK to encrypt the user key. This requires a two-step process.

In the first step, the user needs to create a key file for the KUK. This is done with the following command:

```
skmt.exe /genkuk /kuk
"000102030405060708090A0B0C0D0E0F000102030405060708090a0b0c0d0e0f" /output "C:\User key installation protected mode\keys\kuk_for_new_key.key"
```

Note that in this example:
- We have specified complete file path for the key file.
- We need to use the same key update key as used in section 4.2.

KUK: 000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F

Output File: C:\User key installation protected mode\new_key\kuk_for_new_key.key

Figure 59. Create the KUK Key File

The generated key file kuk_for_new_key.key now contains the KUK, and it can be used like the UFPK to encrypt the new user key.

In the second step, the user needs to encrypt the new user key with the KUK. This is done with the following command.

```
skmt.exe /genkey /kuk file="C:\User key installation protected mode\keys\kuk_for_new_key.key" /mcu "RA-SCE9" /keytype "AES-256" /key "1111111111111222222222222333333333333444444444444" /filetype "csource" /output "C:\User key installation protected mode\keys\new_aes_key.c"
```

Note that in this example:
- We are using 1111111111111222222222222333333333333444444444444 as the new AES-256 plaintext key.
- We are using a randomly generated IV. This IV will be updated in each encryption process.
- The generated new_aes_key.c and new_aes_key.h files include the output information in a data structure. User can directly include these two files in the application project. This is demonstrated in the example project included.
- In the output below, the result of IV and the Encrypted Key are substituted with dummy data.
5. Secure Key Installation via Serial Programming Interface

Follow this section to install the AES-256 and Key-Update Key that were prepared above.

5.1 Setting up the Hardware

Set up the EK-RA6M4 evaluation board as follows.

Set the jumpers to their default settings. Refer to the EK-RA6M4 User’s Manual for details.

Connect the EK-RA6M4 J10 connector to the development PC using a USB micro-B cable to provide power and a debug connection using the on-board debugger.

Erase the entire MCU flash and ensure that MCU is in the SSD Device Lifecycle State. This can be done using the Renesas Flash Programmer, as shown here.

1. Unzip rfp_project.zip
2. Launch the Renesas Flash Programmer GUI executable
3. Select File -> Open Project and select ra6m4_secure_key_install.rpj.
4. Click the tab Device Information -> Initialize Device.

Upon successful initialization, the following message will be printed.

Figure 60. Encrypt the New User Key with KUK

Output File: C:\User key installation protected mode\FSP_v3.5\keys\new_aes_key.h
Output File: C:\User key installation protected mode\FSP_v3.5\keys\new_aes_key.c

KUK: 000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F
IV: 1234567890ABCDEF1234567890ABCDEF
Encrypted key: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF

Figure 61. Open RFP Project and Initialize the Device

Figure 62. RA6M4 Initiation
Unless there are permanently locked flash blocks, the entire flash will be erased and the RA6M4 will be set to the SSD state through the above steps.

5.2 Install the User Key and Key-Update Key via the Serial Programming Interface

After initializing the RA6M4, power-cycle the board and follow the steps below to install the AES-256 user key and Key-Update Key prepared earlier.

- Navigate to the User Keys tab and check Write User Keys.

![Figure 63. Select Write User Keys](image)

- Click and browse to the AES256.rkey file generated in Figure 57. Set the Address property to a data flash or code flash address applicable for your specific application. In this example, the AES key will be installed to the second block of Data Flash at 0x08000040.

![Figure 64. Configure the AES-256 User Key Selection and Installation Address](image)

- Click Add Key. The selected AES key will be added for installation.

![Figure 65. AES-256 User Key is Selected for Installation](image)
Click and browse to the KUK.rkey file generated in Figure 58. Set the **Address** property to a data flash or code flash address applicable for your specific application. In this example, the Key-Update Key will be installed at code flash address 0x40000.

![Figure 66. Configure the Key-Update Key Selection and Installation Address](image)

- Click **Add Key**. The selected Key-Update Key will be added for installation.

![Figure 67. Key Update Key is Configured for Installation](image)

- Browse to the **Operation Settings** tab and note that **Erase**, **Program**, **Verify**, and **Erase Before Program** are selected.

![Figure 68. Select to Perform Flash Erase, Program, and Verify](image)
• Browse to the **Block Settings** tab and note that the entire Flash Region is selected for Erase.

![Figure 69. Entire Flash Region is Selected for Erase](image)

• Browse to the **Operation** tab. Click **Start** to install the AES-256 and the Key-Update Key. The installation should succeed with similar output message as shown below at the selected flash addresses.

![Figure 70. Secure Keys Successfully Installed](image)

In this example code, no application is programmed since we are interested only in the key installation. In a production flow, it is possible to program the application and user keys together. This operation can also be performed using the command line function of RFP.

### 6. Example Project

Prior to exercising the example project, the user must follow the steps in the sections 3, 4, and 5 above to install the AES-256 user key and KUK and to encrypt the new AES-256 key using the KUK.

#### 6.1 Overview

This pair of TrustZone-based secure and non-secure example projects provides the following functions:
Secure project (secure_key_install_update_ra6m4_s):
- Uses the installed (wrapped) AES-256 key to perform an encryption and description operation using AES256-CBC.
- Uses the installed (wrapped) Key-Update Key (KUK) and the new user key encrypted with the KUK to install a new wrapped AES-256 key, stores this new key in data flash.
- Uses the new AES-256 user key stored in data flash to perform an encryption and decryption operation using AES256-CBC.

Non-secure project (secure_key_install_update_ra6m4_ns):
- Establishes an RTT Viewer interface to allow users to select the intended Secure Crypto Engine and flash operation.
- Calls the non-secure callable APIs provided from the secure project based on user selection from the RTT Viewer interface.
- Prints the user operation results on the RTT Viewer.

The FSP modules used in this pair of example projects are:
- `r_sce_protected`: this module is used in the secure region and provides services to the non-secure region via non-secure callable APIs.
- `r.flash_hp`: this module is used in the secure region and provides services to the non-secure region via non-secure callable APIs.

For more information on designing applications with TrustZone support, refer to the application project *Renesas RA Family MCU Security Design with TrustZone – IP Protection.*
6.2 Using the RFP Installed Keys

The keys that are directly installed into the MCU flash using RFP cannot be directly used by the FSP Crypto APIs. The user needs to read them out to flash first prior to using them in the FSP Crypto API.

For example, the following code snippet reads the AES256 key from flash. The destination buffer can then be used in the cryptographic operations.

```c
sce_aes_wrapped_key_t installed_key;
installed_key.type = SCE_KEY_INDEX_TYPE_AES256;
memcpy(installed_key.value, (uint32_t *)DIRECT_KEY_ADDRESS, HW_SCE_AES256_KEY_INDEX_WORD_SIZE*4);
```

The following code snippet reads the installed key update key from the flash. The destination buffer can then be used to wrap new user keys.

```c
sce_key_update_key_t   kuk_key;
kuk_key.type = SCE_KEY_INDEX_TYPE_UPDATE_KEY_RING;
memcpy(kuk_key.value, (uint32_t *)(KUK_ADDRESS), HW_SCE_UPDATE_KEY_RING_INDEX_WORD_SIZE*4);
```

This application project does not include an example usage for other types of keys installed. The following code snippet reads the RSA2048 public key to flash prior to using it for a cryptographic operation.

```c
sce_rsa2048_public_wrapped_key_t installed_rsa_public_key;
installed_rsa_public_key.type = SCE_KEY_INDEX_TYPE_RSA2048_PUBLIC;
uint32_t rsa_2048_public_key_size = sizeof(installed_rsa_public_key.value);
uint8_t destination_rsa_2048_public_key[rsa_2048_public_key_size];
memcpy(destination_rsa_2048_public_key, (uint32_t *)RSA_2048_PUB_KEY_ADDRESS, rsa_2048_public_key_size);
```

6.3 FSP Crypto Module Support for User Key Update

This section introduces the FSP Crypto APIs for SCE Protected Mode that are used for secure user key installation and update. For a complete description of all FSP Crypto APIs, refer to the FSP User's Manual.

To use keys that have been installed via the secure key installation process using a device programmer, the application must reference those keys at the address where they were installed. If you install keys at addresses other than those demonstrated above, be sure to change your application code to reflect those addresses. See instructions on 6.5.

To perform secure key update, use the following API to generate a new wrapped key using a previously installed Key-Update Key:

```c
fsp_err_t R_SCE_AES256_EncryptedKeyWrap(uint8_t *initial_vector, uint8_t *encrypted_key, sce_key_update_key_t *key_update_key, sce_aes_wrapped_key_t *wrapped_key)
```
The API parameters are:

- **[in]** initial_vector: Pointer to a buffer that holds the initialization vector that was used to encrypt the new key. This must be acquired from the IV that was output during the key encryption process shown in section 4.3 Figure 60. Since this value is randomly generated with every execution, this value must be updated in the example project.

- **[in]** encrypted_key: Pointer to a buffer that holds the encrypted new key. In this example, it is the encrypted AES-256 user key that was output during the key encryption process shown in section 4.3 Figure 60. Since this value is dependent on the randomly generated IV, this value must be updated in the example project.

- **[in]** key_update_key: Pointer to the Key-Update Key installed on the MCU. This address must match the address used when installing the KUK in section 5.2 Figure 70. Ensure that the address in the example project matches the installation address.

- **[in, out]** wrapped_key: This is the SRAM buffer to store the wrapped new user key. For security considerations, it is recommended to erase this buffer right after the wrapped key is saved to flash. In this application project, the newly generated wrapped key is stored to data flash and used in the example project.

### 6.4 Save the New Wrapped Key to Data Flash

Once the new wrapped key (stored in `wrapped_new_key_key` buffer) is generated by the following function call, user needs to use the flash driver `r_flash_hp` to manually store it to the data flash.

```c
sce_aes_wrapped_key_t wrapped_new_user_key;
error = R_SCE_AES256_EncryptedKeyWrap(iv_encrypt_new_key, encrypted_new_key, &kuk_key, &wrapped_new_user_key);
```

Refer to function `flash_hp_data_flash_operations` for storing the AES key to flash. See the following `r_flash_hp` function call where `write_buffer` is the SRAM buffer which holds the wrapped new user key. Macro definition `DIRECT_KEY_ADDRESS` is the data flash destination address to store the new wrapped user key.

```c
/* Write code flash data*/
error = R_FLASH_HP_Write(&g_flash0_ctrl, (uint32_t) write_buffer, DIRECT_KEY_ADDRESS, BLOCK_SIZE);
```

### 6.5 Import and Compile the Example Project

Follow the steps below to exercise the example project. Note that there are sections of the code that must be updated using the secure key installation results generated above prior to compiling and running the project.

1. Launch e² studio and import `r11an0496_iinstall_update_secure_key.zip` file to a workspace.
2. Open `crypto_operations.c` in the secure project, under the folder `\secure_key_install_update_s\src`.
3. At the top of `flash_storage.h`, find the definition for macro `DIRECT_KEY_ADDRESS` and `KUK_ADDRESS`:
   - **Update** `DIRECT_KEY_ADDRESS` to the address used in Figure 64
   - **Update** `KUK_ADDRESS` to address used in Figure 66
4. Add `new_aes_key.h` and `new_aes_key.c` generated in Figure 60 to folder `\secure_key_install_update_s\src`
5. If different file names are used, update the `#include` definition on this line to reflect the new file name.

```
#include "crypto_operations.h"
#include "hal_data.h"
#include "r_sce.h"
#include "flash_storage.h"
#include "new_aes_key.h"
```

**Figure 72. Include the Generated Header File**
6. Next, double click `configuration.xml` from the secure project. Once the configurator is opened, click **Generate Project Content** and then compile the secure project.

7. Expand the non-secure project and double click the `configuration.xml` file. Once the configurator is opened, click **Generate Project Content** and compile the non-secure project.

### 6.6 Running the Example Project

Once the source code compilation is successful, follow the steps below to exercise the example projects:

1. Choose to debug from the non-secure application. Right-click on `installing_utilizing_user_keys_ra6m4_ns` and select **Debug As > Renesas GDB Hardware Debugging**.

2. Execution will halt at the secure project reset handler.

3. Click **Resume** twice to run the project.

4. Open the J-Link RTT Viewer with the settings shown below.

![Figure 73. Running to the Secure Project Reset Handler](image)

![Figure 74. RTT Viewer Setting](image)
5. Click **OK**. The following menu should be printed.

```
MENU to Select
00> Press 1 exercise the cryptographic operation with directly installed user key
00> Press 2 create updated new user key using PUF
00> Press 3 exercise the cryptographic operation with newly updated user key
```

**Figure 75. Main RTT User Menu**

a. Input **1** to perform encryption and decryption using the AES-256 user key installed via the device programmer.

```
00> Cryptographic operation with directly installed user key, which is installed via the serial interface.
00> Result: Cryptographic operation is successful with directly installed user key
```

**Figure 76. Crypto Operation with Directly Installed AES-256 Key**

b. Input **2** to perform a key update to wrap a new AES-256 user key and save the new key to data flash. Note that the SCE handles the wrapping of the new key internally, without exposing the plaintext key.

```
< 2
00> Update the new user key encrypted with key update key and store the new wrapped key in data flash.
00> Result: User Key is updated and stored to Data Flash
```

**Figure 77. Wrapped New User Key using KUK and Stored to Data Flash**

c. Input **3** to perform encryption and decryption using the newly wrapped AES-256 user key.

```
< 3
00> Cryptographic operation with new wrapped user key stored in data flash
00> Result: Cryptographic operation is successful with Updated User Key
```

**Figure 78. Crypto Operation with the New User Key**

At this point, the demonstration of the example project is finished.

7. **References**

1. Renesas RA Family Device Lifecycle Management Key Installation Application Note (R11AN0469)
2. Renesas RA Family Secure Crypto Engine Operational Modes Application Note (R11AN0498)
3. Renesas RA Family MCU Security Design with TrustZone – IP Protection (R11AN0467)
8. Website and Support

Visit the following URLs to learn about the RA family of microcontrollers, download tools and documentation, and get support.

- EK-RA6M4 Resources: renesas.com/ra/ek-ra6m4
- RA Product Information: renesas.com/ra
- Flexible Software Package (FSP): renesas.com/ra/fsp
- RA Product Support Forum: renesas.com/ra/forum
- Renesas Support: renesas.com/support
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>May.19, 2021</td>
<td>-</td>
<td>First release document</td>
</tr>
<tr>
<td>1.10</td>
<td>Jan. 27, 2022</td>
<td>-</td>
<td>Update to use Security Key Management Tool CLI V1.0.0</td>
</tr>
</tbody>
</table>
Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robot; etc.
   - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; underwater repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

Corporate Headquarters
TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information
For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks
Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2022 Renesas Electronics Corporation. All rights reserved.