Renesas RA Family

Device Lifecycle Management Key Installation

Introduction

Device Lifecycle Management (DLM) is the management of the process by which a product goes from inception to development to production and then eventually end-of-life. The RA Family MCU debug capability and serial programming capability are defined by the device lifecycle states.

The first-generation RA DLM system uses the plaintext MCU debug Identification Code (ID) for the Non-TrustZone®-based RA Family MCU Groups. The new generation DLM system for the RA Family Arm® TrustZone®-enabled MCUs takes advantage of an authenticated device lifecycle state transition process based on Wrapped Key management. This new generation DLM system offers enhanced IP protection while maintaining the capability to regress the device lifecycle state.

This application note focuses on the use cases of the new generation TrustZone®-enabled RA Family MCU DLM system. It walks you through the DLM Key Wrapping services, Key Installation steps, and Device Lifecycle Regression steps. It also briefly discusses the first generation DLM system.

The RA6M4 is used as an example for the walk-through of the key wrapping, key installation and lifecycle state regression. The general steps apply to all TrustZone®-enabled RA Family MCUs.

Required Resources

The following resources are referenced throughout this application note:

Development tools and software

- Renesas Flash Programmer (RFP) v3.08 or later https://www.renesas.com/us/en/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html
- e^2 studio IDE v2020 10 or later
- RA Family Flexible Software Package (FSP) v2.0.0 or later
- SEGGER J-Link® USB driver V6.86 or later

The FSP, J-Link USB drivers, and e^2 studio are bundled in a downloadable platform installer available on the FSP webpage at renesas.com/ra/fsp.

Hardware

- EK-RA6M4, Evaluation Kit for RA6M4 MCU Group (renesas.com/ra/ek-ra6m4)
- Test PC running Windows® 10 OS
- 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 e^2 studio IDE and Renesas Flash Programmer (RFP). In addition, the application note assumes that you have some knowledge of RA Family MCU security features. See the Security Features section in the *Renesas RA6M4 Group MCU User's Manual: Hardware* for background information.

The intended audience includes product developers, product manufacturers, product support, or end users who are involved with any stage of the device lifecycle management of the RA Family MCUs.

Contents

1. Introduction to Device Lifecycle Management for RA Family MCU Groups

The RA Family DLM system can play a key role in the customer application development, production, product deployment, and failure analysis management.

1.1 Device Lifecycle Management using Debug ID Code

First generation RA Family MCU Groups use the Debug Identification Code (ID Code) to re-enable the debug interface after device deployment. Product development is typically managed by a single trusted software team.

The following graphic shows the typical lifecycle management of an Arm® Cortex®-M0+ and M4 based RA Family MCU Group.

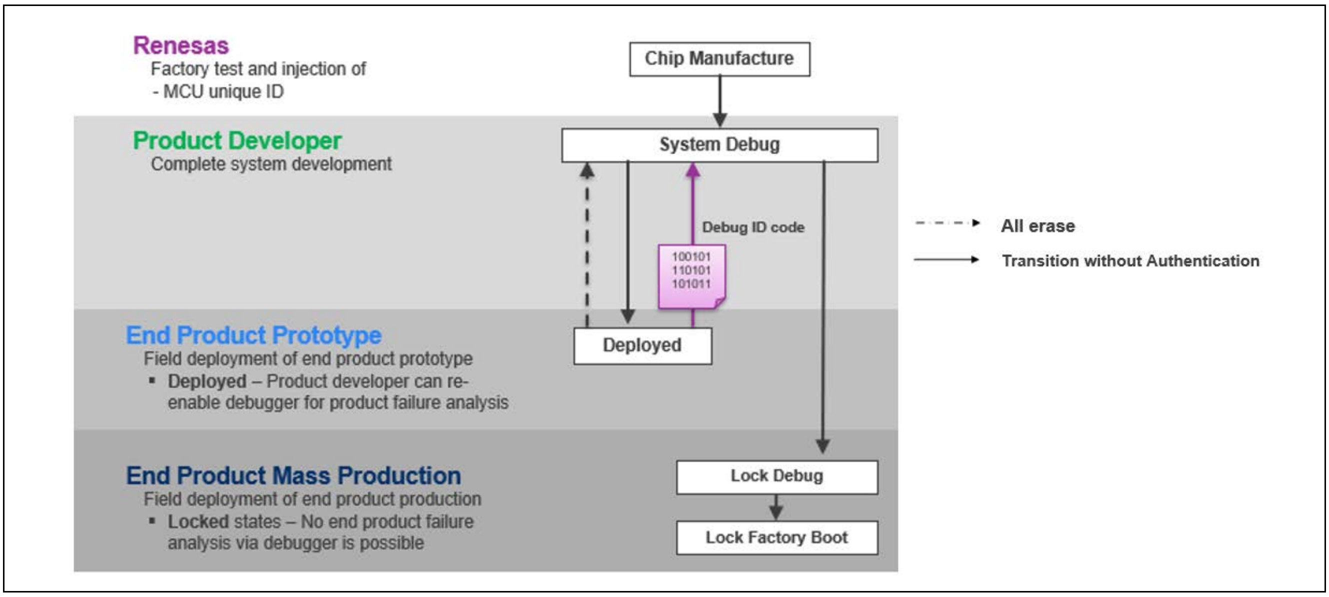

Renesas
Factory test and injection of
- MCU unique ID
Chip Manufacture
Product Developer
Complete system development
System Debug
Debug ID code
100101
110101
101011
All erase
Transition without Authentication
End Product Prototype
Field deployment of end product prototype
▪ Deployed – Product developer can re-enable debugger for product failure analysis
Deployed
End Product Mass Production
Field deployment of end product production
▪ Locked states – No end product failure analysis via debugger is possible
Lock Debug
Lock Factory Boot

Figure 1. DLM System for RA Family MCU Groups Using Debug ID Code

- Upon receiving devices from the Renesas factory, the Product Developer creates the application code with full access to the MCU assets – flash, RAM, peripherals, and so forth.
- After development is finished, for the product prototype, the Product Developer sets up the ID Code for temporary protection. This disables further access to the debug system for any attempts which do not provide the ID Code during debug connection. For failure analysis, the End Product User can send the MCU back to the Product Developer. The Product Developer can use the ID Code to reenable the debugger for failure analysis.
- For mass production, the manufacturer programs the product and sets up the ID Code to permanently disable further access to the debug system. The factory bootloader can also be permanently disabled via the Flash Access Window (FAW) and Security MPU. See the *Securing Data at Rest using Security MPU* Application Project for the operational steps of ID Code protection, FAW setup, and Security MPU setup.

This DLM system offers adequate protection for many applications, but it is prone to eavesdropping and replay attacks, since the unlock code is transmitted in plain text.

1.2 Device Lifecycle Management System using Arm® TrustZone® Technology

The next-generation RA Family MCUs have enhanced DLM features. The first enhancement, enabled by TrustZone technology, is the ability to separate the application into two regions, often called secure/trusted and non-secure/non-trusted. A typical use case for this separation is to isolate the product's Root of Trust, consisting of the device's identity and supporting services that access or manipulate the device identity. After the secure system is developed and programmed into the device, the lifecycle state is moved from Secure System Debug (SSD) to Non-Secure System Debug (NSECSD). This allows the application that uses the functionality provided by the Root of Trust to be developed and debugged while protecting the Root of Trust itself.

1.2.1 Definitions of the Device Lifecycle States

Device lifecycle identifies the current phase of the device and controls access permissions to the debug interface, the serial programming interface, and Renesas test mode. The next-generation RA family MCUs offer cryptographic authentication to restore various levels of programming and debugging capabilities. During the production process, the MCU is provisioned with one or more DLM state keys. When the developer requests a DLM state regression to unlock the MCU, the MCU issues a challenge. The developer must then use the appropriate DLM state key to construct a response. If the response is correct, the MCU will regress to the requested DLM state, unlocking the defined programming and debugging capabilities. This mechanism prevents eavesdropping and replay attacks, providing enhanced IP protection while retaining the ability to perform End Product failure analysis.

There are three debug levels for TrustZone®-based RA Family MCUs:

- DBG2: The debugger connection is allowed, and there is no restriction on access to memories and peripherals
- DBG1: The debugger connection is allowed, and access is restricted to only non-secure memory regions and peripherals
- DBG0: The debugger connection is not allowed

The serial programming interface can communicate with the factory bootloader of the device when the device is in boot mode. Users can use the Renesas Flash Programming (RFP) application to communicate with the factory bootloader via the serial programming interface to update the device lifecycle states.

The following table provides the definition of the various device lifecycle states and the corresponding debug levels as well as the serial programming interface capability.

Table 1. TrustZone-Enabled RA Family MCU Group Device Lifecycle States

Lifecycle State	Definition and State Features	Debug Level	Serial Programming	Renesas Test Mode
CM	• "**C**hip **M**anufacturing" • The device is in Renesas factory. • Developer receives the device in this state. • MCU Unique ID and Hardware Unique Key (HUK) are injected.	DBG2	• Available. • Cannot access code/data flash area.	Not available
SSD	• "**S**ecure **S**oftware **D**evelopment" • The secure part of the application is being developed. • SECDBG_KEY and RMA_KEY can be injected.	DBG2	• Available. • Can program/erase/read all code/data flash areas.	Not available
NSECSD	• "**N**on-**SEC**ure **S**oftware **D**evelopment" • The non-secure part of the application is being developed. • NONSECDBG_KEY can be injected. • It is possible to regress to SSD state without flash erase if SECDBG_KEY is injected in SSD state. • It is possible to erase the entire flash to SSD state.	DBG1	• Available. • Can program/erase/read non-secure code/data flash areas.	Not available

Lifecycle State	Definition and State Features	Debug Level	Serial Programming	Renesas Test Mode
DPL	• "**D**e**PL**oyed" • The device is in the field. • It is possible to regress to NSECSD state without flash erase if NONSECDBG_KEY is injected in NSECSD state. • It is possible to erase the entire flash to SSD state.	DBG0	• Available. • Cannot access code/data flash areas.	Not available
LCK_DBG	• "**L**o**CK**ed **D**e**B**u**G**" • **The debug interface is permanently disabled.**	DBG0	• Available • Cannot access code/data flash areas.	Not available
LCK_BOOT	• "**L**o**CK**ed **BOOT** interface" • **The debug interface and the serial programming interface are permanently disabled.**	DBG0	• Not available.	Not available
RMA_REQ	• "**R**eturn **M**aterial **A**uthorization **REQ**uest" • Request for RMA. • The customer must send the device to Renesas in this state.	DBG0	• Available. • Cannot access code/data flash areas.	Not available
RMA_ACK	• "**R**eturn **M**aterial **A**uthorization **ACK**nowledged" • Failure analysis by Renesas.	DBG2	• Available. • cannot access code/data flash areas.	Available

1.2.2 Summary of the Device Lifecycle States and Transitions

Figure 2 is a summary of all the possible state transitions for the entire lifecycle of the MCU.

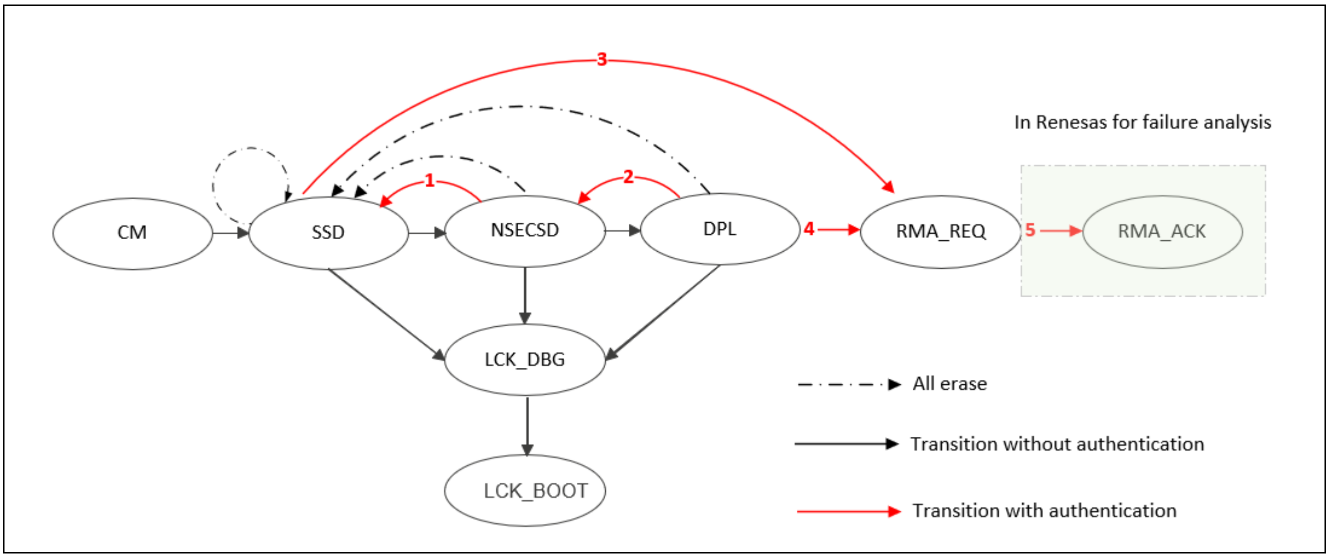

3
In Renesas for failure analysis
1
2
CM
SSD
NSECSD
DPL
4
RMA_REQ
5
RMA_ACK
LCK_DBG
LCK_BOOT
All erase
Transition without authentication
Transition with authentication

Figure 2. RA MCU Device Lifecycles

As we can see from Figure 2, there are three types of transitions.

`All Erase` Operation

- Successful execution of the **`All erase`** command changes the device lifecycle back to SSD.
- The **`All erase`** command will erase the entire flash memory if flash block locking is temporary rather than permanent.
- If permanent block protection is enforced by the developers, then the **`Initialize`** command does not execute.
- **`All erase`** can be achieved by the **`Initialize`** command in RFP unless an **`Initialize`** command itself is disabled. The **`Initialize`** command can be issued by anyone, so the contents of the flash memory are easily erased.
- Developers who do not want this functionality can disable the **`Initialize`** command permanently using RFP as shown in Figure 8 . However, once the **`Initialize`** command is disabled, it can never be recovered.
- Manufacturers can disable the **`Initialize`** command during production if desired.
- Refer to section 1.2.4 and section 1.2.5 for the operational details.

DLM Authentication Keys

As shown in Figure 2, DLM keys are needed to regress the lifecycle states without erasing the contents of the MCU flash memory for failure analysis. The DLM keys include RMA_KEY, SCEDBG_KEY and NONSCEDBG_KEY.

- As described in Table 1, the secure developer can inject SECDBG_KEY and RMA_KEY into the device in the SSD state. The RMA_KEY can be used for eventual product RMA and the SECDBG_KEY can be used to return the device to the secure development state.
- As described in Table 1, a non-secure development team can inject NONSECDBG_KEY into the MCU when the MCU is in NSECSD state. The NONSECDBG_KEY can be used to return the device to the non-secure development state.

Authenticated Transitions

Authenticated transitions are typically used for failure analysis during development and after deployment. Failure analysis typically involves Device Lifecycle State regression to a previous development state or advancing to Return Material Authorization Request state.

- The following lifecycle state transitions can regress to a previous development state without erasing the flash contents to allow failure analysis being carried out:
  - Secure Development (SSD) to Non-Secure Development (NSECSD)
    - Transition 1: uses SECDBG_KEY
  - Prototype Deployed state (DPL) to Non-Secure Development (NSECSD)
    - Transition 2: uses NONSECDBG_KEY
- In addition, it is possible to advance to the Return Material Authentication Request state from the Prototype Deployed state or the Secure Development state with the contents on the flash memory erased except the permanently locked flash block.
  - Prototype Deployed state (DPL) to Return Material Authorization Request (RMA_REQ)
    - Transition 3: uses RMA_KEY
  - Secure Development State (SSD) to Return Material Authorization Request (RMA_REQ)
    - Transition 4: use RMA_KEY or MCU Unique ID
- Transition 5 involves a Renesas proprietary operation. No details on this transition will be provided in this application note. Please contact a Renesas sales representative for relevant information.

For more details on the use cases of device lifecycle transitions, please refer to section 2.

1.2.3 Advantages of Arm® TrustZone® Enabled Device Lifecycle Management

The following is a summary of the TrustZone-enabled Device Lifecycle Management System:

- Protects Root of Trust and IP system
- Reenables debug interface and serial programming interface without erasing MCU flash to enable failure analysis
- Erases entire flash to allow the device to return to Secure Software Development stage to avoid scrapping the MCU
- Prevents eavesdropping and replay attacks, providing enhanced IP protection while retaining the ability to perform End Product failure analysis.

1.2.4 Overview of Device Lifecycle State Management using Renesas Device Partition Manager

Renesas Device Partition Manager is a utility integrated with e^2 studio for Device Lifecycle State management during production development. A user can use the Renesas Device Partition Manage to perform the following functions:

- Query current device lifecycle
- Query device IDAU region setup
- Initialize device to SSD state all unlock flash blocks erased
- Set up IDAU regions

Note that the user needs to power cycle the board prior to working with **Renesas Device Partition Manager** after a debug session if using J-Link as the connection interface.

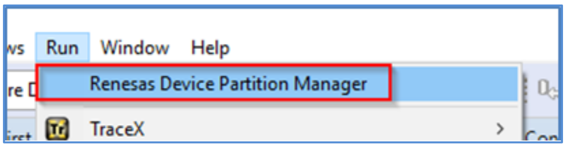
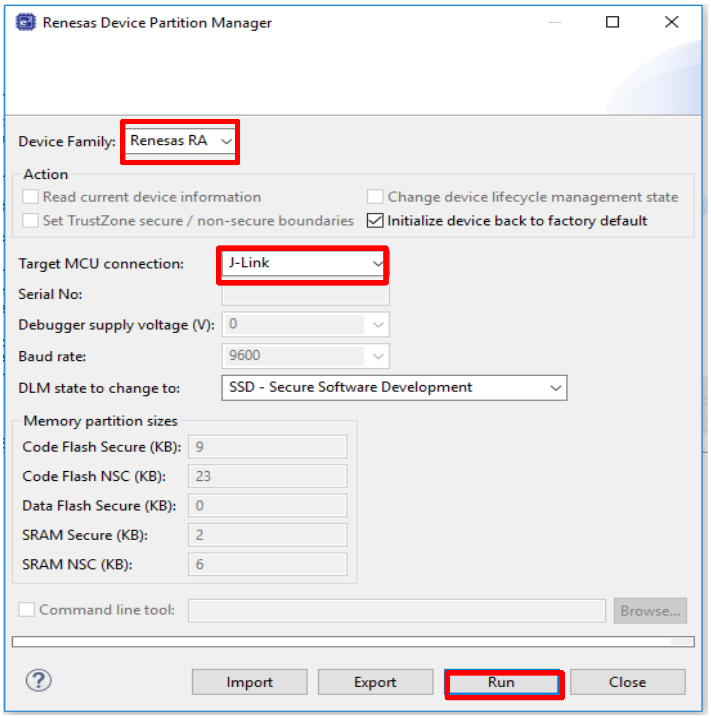


Figure 3. Open the Renesas Device Partition Manager

Following is an example setting for performing **Initialize device back to factory default**. Choose the connection method and then click **Run**.


Renesas Device Partition Manager

Device Family: Renesas RA

Action

□ Read current device information
□ Change device lifecycle management state
□ Set TrustZone secure / non-secure boundaries
☑ Initialize device back to factory default

Target MCU connection: J-Link

Serial No:

Debugger supply voltage (V): 0

Baud rate: 9600

DLM state to change to: SSD - Secure Software Development

Memory partition sizes

Code Flash Secure (KB): 9

Code Flash NSC (KB): 23

Data Flash Secure (KB): 0

SRAM Secure (KB): 2

SRAM NSC (KB): 6

□ Command line tool:

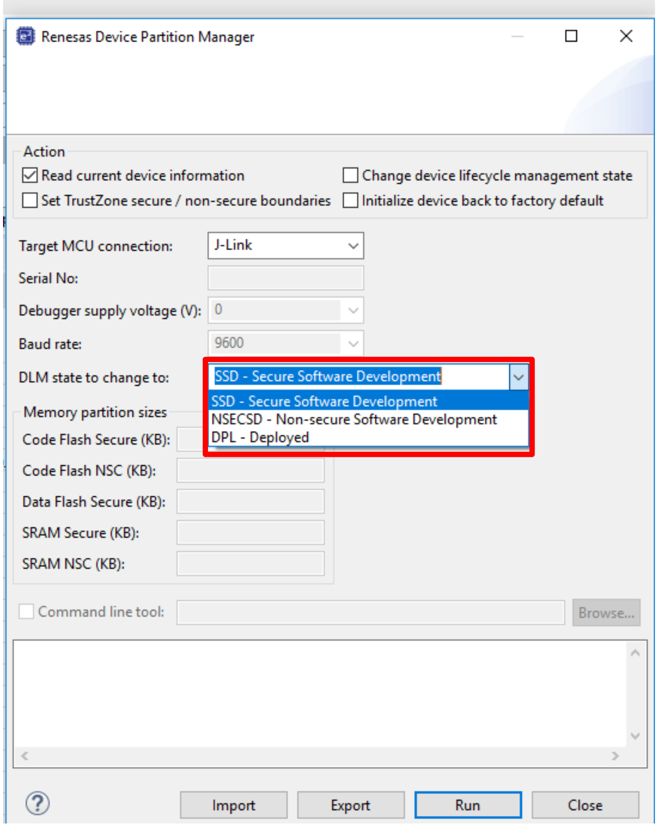
Figure 4. Initialize RA6M4 Using Renesas Device Partition Manager

During Product Development, users typically use the **Renesas Device Partition Manager** to advance the device lifecycle:

- SSD -> NSECSD
- NSECSD-> DPL

Limitations of the Renesas Device Partition Manager

- Renesas Device Partition Manager does not support authenticated transitions. For failure analysis, use RFP.
- Renesas Device Partition Manager supports transitions to limited device lifecycle states and requires e^2 studio IDE environment to operate. For mass production where transitioning to LCK_DBG or LCK_BOOT is required, the user should use RFP.



Renesas Device Partition Manager

Action

☑ Read current device information ☐ Change device lifecycle management state

☐ Set TrustZone secure / non-secure boundaries ☐ Initialize device back to factory default

Target MCU connection: J-Link

Serial No:

Debugger supply voltage (V): 0

Baud rate: 9600

DLM state to change to: SSD - Secure Software Development

SSD - Secure Software Development

NSECSD - Non-secure Software Development

DPL - Deployed

Memory partition sizes

Code Flash Secure (KB):

Code Flash NSC (KB):

Data Flash Secure (KB):

SRAM Secure (KB):

SRAM NSC (KB):

☐ Command line tool:

Figure 5. Advance Device Lifecycle States Using Renesas Device Partition Manager

1.2.5 Overview of Device Lifecycle State Transitions using Renesas Flash Programmer

The Renesas Flash Programmer provides end to end production flow support. In addition to the Device Lifecycle Management functionalities provided by Renesas Device Partition Manager, RFP provides the following functions.

- Supports authenticated device lifecycle state transitions. The DLM key needs to be installed and used in the authentication process. Refer to section 3.6 for authenticated transitions using RFP.
- Support all unauthenticated device lifecycle state transitions except transitioning to RMA_ACK (which could only happen within an authorized team within Renesas as shown in Figure 2).
- Disable the **`Initialize`** command. This may be desired if the device is deployed in DPL state and there is a requirement to avoid accidental flash content erase. However, once the **`Initialize`** command is disabled, it can never be recovered.

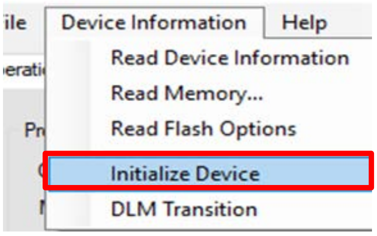


Figure 6. Using the `Initialize Device` Command to execute `All erase`

For the unauthenticated transitions, use **DLM Transitions** as shown in Figure 7 to perform the transitions.

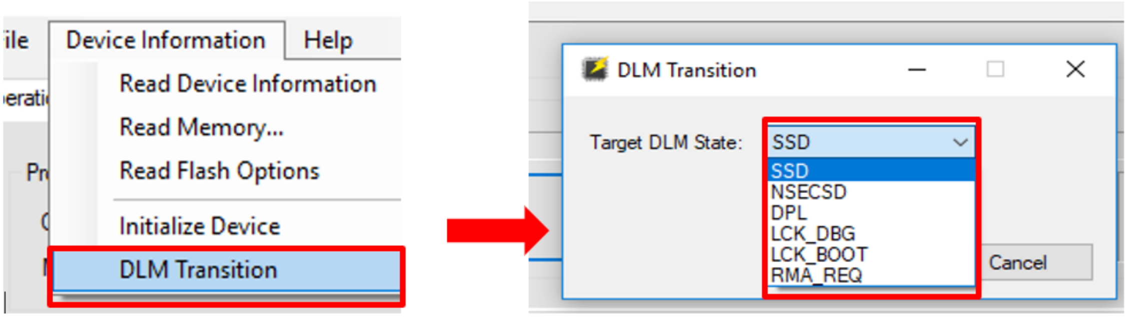
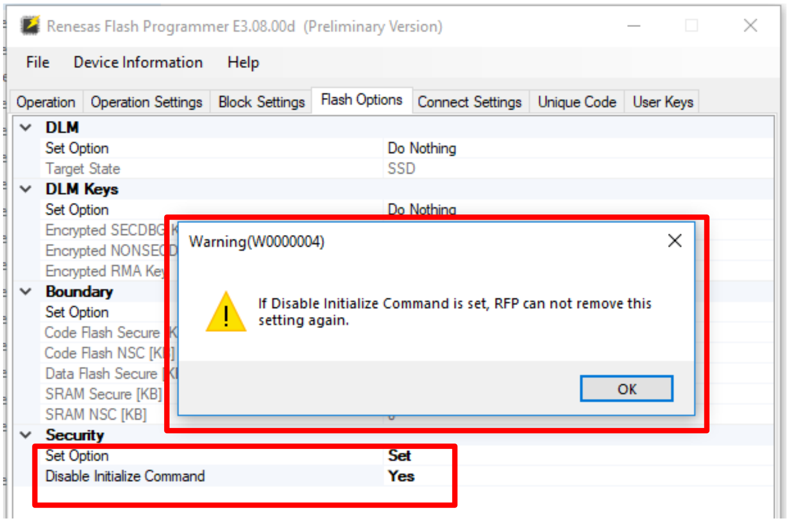


Figure 7. Available Lifecycle State Transitions Supported by RFP



DLM
Set Option Do Nothing
Target State SSD
DLM Keys
Set Option Do Nothing
Encrypted SECDBG K
Encrypted NONSECD
Encrypted RMA Key
Boundary
Set Option
Code Flash Secure [K
Code Flash NSC [KB]
Data Flash Secure [K
SRAM Secure [KB]
SRAM NSC [KB]
Security
Set Option Set
Disable Initialize Command Yes

Warning(W0000004)

If Disable Initialize Command is set, RFP can not remove this setting again.

Figure 8. Disable the `Initialize` Command

2. Use Cases of Device Lifecycle Management System in Arm® TrustZone®-enabled Application

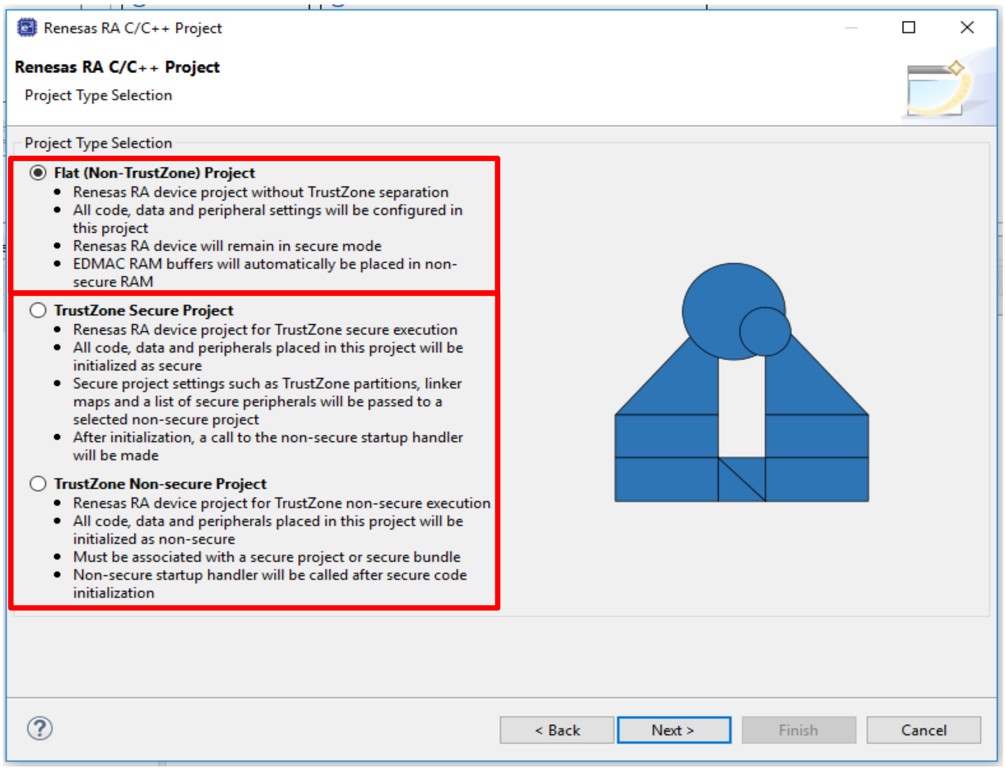
This section explains the use cases of the Device Lifecycle Management system in the development, production, and deployment stages of the Renesas TrustZone technology-enabled products.

The Renesas RA IDE supports three development models for a TrustZone technology-enabled MCU. The Renesas RA Project Generator provides three project types to support these three development models as shown in Figure 9.

- Flat Project Development Model
  - No TrustZone technology awareness during development
  - Benefits from the Device Lifecycle Management during production stage
- Split Project Development Model
  - Uses both Secure and Non-secure project types
  - This development model allows product development by two teams.

- Secure Developers - create Root of Trust (5.1) or an isolated subsystem. Development is carried out in SSD state.
- Non-secure Developers - create application that is built on the Root of Trust and uses the isolated subsystem. Development is carried out in NSECSD state.
- Secure system is locked from access from the non-secure developers.

- Combined Project Development Model
  - Use both Secure and Non-secure Project type.
  - Both secure and non-secure application development is carried by one single trusted team in SSD state.
  - Developer has access to both Secure and Non-secure assets (hardware, code, data, debugging).



● Flat (Non-TrustZone) Project
- Renesas RA device project without TrustZone separation
- All code, data and peripheral settings will be configured in this project
- Renesas RA device will remain in secure mode
- EDMAC RAM buffers will automatically be placed in non-secure RAM

○ TrustZone Secure Project
- Renesas RA device project for TrustZone secure execution
- All code, data and peripherals placed in this project will be initialized as secure
- Secure project settings such as TrustZone partitions, linker maps and a list of secure peripherals will be passed to a selected non-secure project
- After initialization, a call to the non-secure startup handler will be made

○ TrustZone Non-secure Project
- Renesas RA device project for TrustZone non-secure execution
- All code, data and peripherals placed in this project will be initialized as non-secure
- Must be associated with a secure project or secure bundle
- Non-secure startup handler will be called after secure code initialization

Figure 9. Renesas RA Arm® TrustZone® Technology Enabled Project Types

2.1 Overview of TrustZone Technology Enabled Development Models

Developing with the TrustZone technology enabled RA Family MCU Groups typically uses two development models as explained previously: the Split Project Development model and the Combined Project Development model. This section describes the use case of the Device Lifecycle Development for these two development models.

2.1.1 Split Project Development Model

The general flow of the Split Project Development Model is described as follows:

- At one of the final steps during the MCU manufacturing, Renesas will inject an MCU Unique ID and HUK to the MCU. The MCU is then delivered to the Secure Developer.
- The Secure Product Developer develops the secure application and locks the secure region from access by the Non-secure region. The MCU is then delivered to the non-secure developer.
- The Non-secure Product Developer develops the non-secure application. In this stage, the secure system is not visible to the non-secure product developer.
- The End Product User may receive the MCU with the debug interface temporarily locked or permanently locked. The serial programming interface may be permanently locked or available (but with limited functionality).

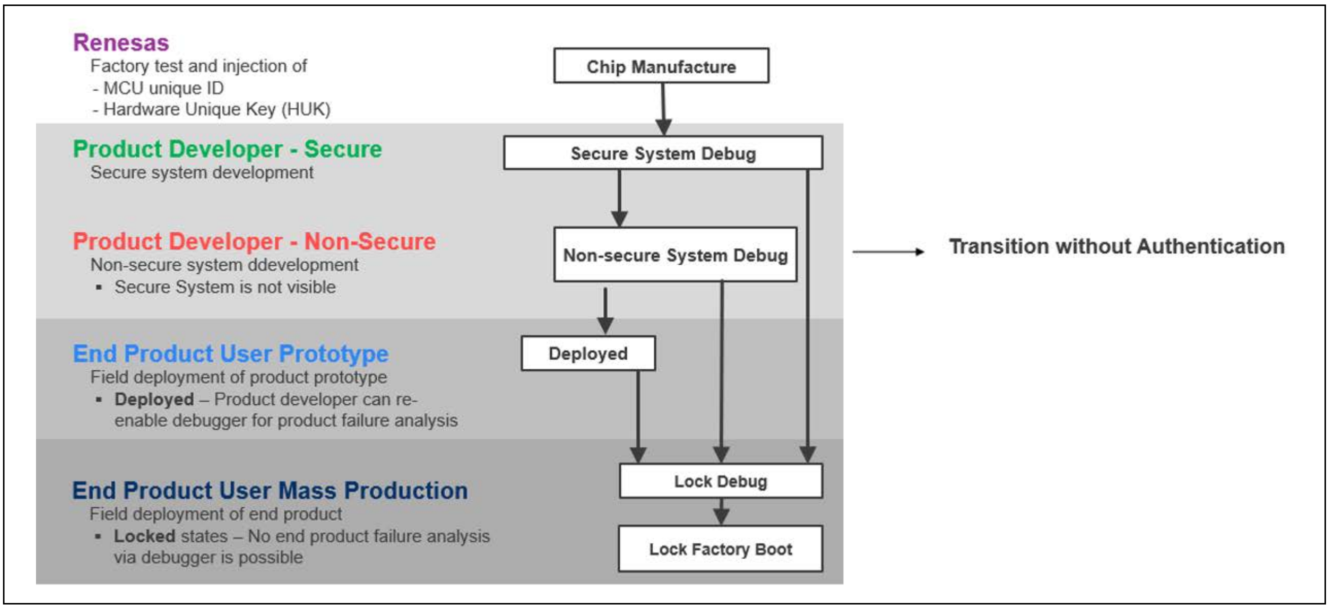

Renesas
Factory test and injection of
- MCU unique ID
- Hardware Unique Key (HUK)
Product Developer - Secure
Secure system development
Product Developer - Non-Secure
Non-secure system ddevelopment
▪ Secure System is not visible
End Product User Prototype
Field deployment of product prototype
▪ Deployed – Product developer can re-enable debugger for product failure analysis
End Product User Mass Production
Field deployment of end product
▪ Locked states – No end product failure analysis via debugger is possible
Chip Manufacture
Secure System Debug
Non-secure System Debug
Deployed
Lock Debug
Lock Factory Boot
Transition without Authentication

Figure 10. Split Project Development Model

Summary of Device Lifecycle State Transitions during Development, Production and Failure Analysis

Figure 11 adds the lifecycle states to the development flow to provide a complete picture of all possible transitions during the development, production flow, and failure analysis for the Split Project Development Model.

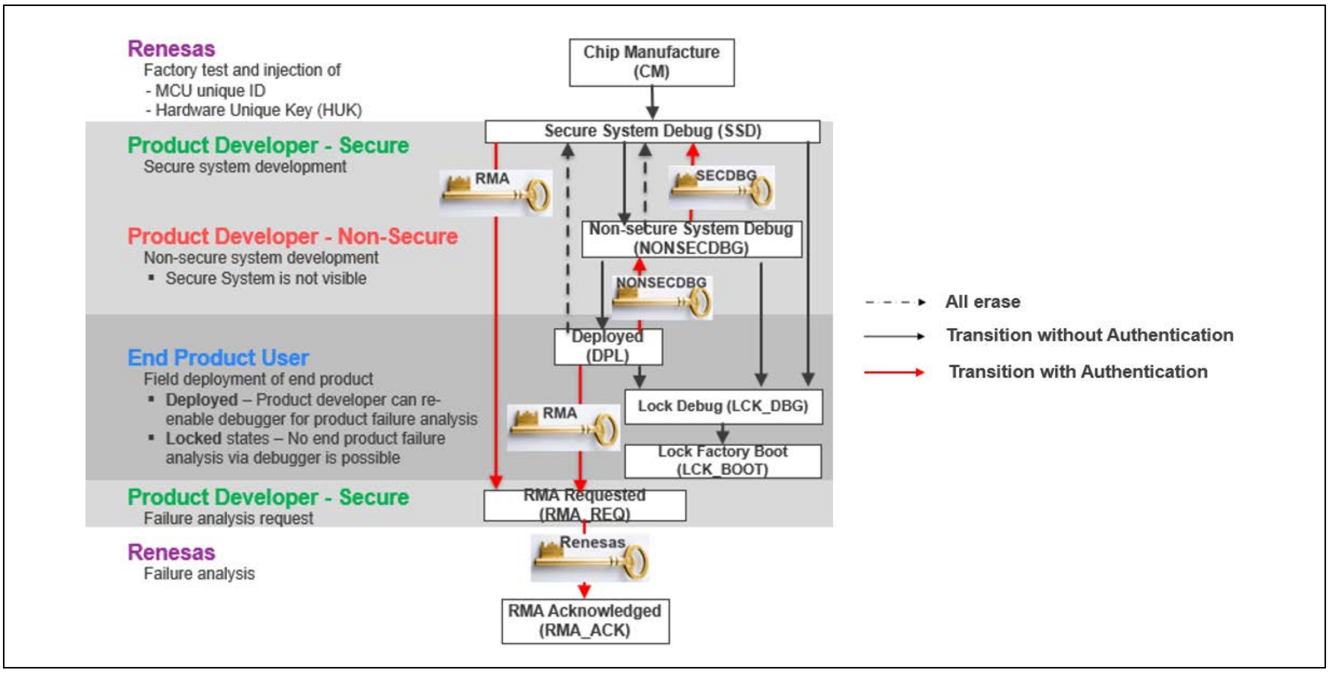

Renesas
Factory test and injection of
- MCU unique ID
- Hardware Unique Key (HUK)
Product Developer - Secure
Secure system development
Product Developer - Non-Secure
Non-secure system development
▪ Secure System is not visible
End Product User
Field deployment of end product
▪ Deployed – Product developer can re-enable debugger for product failure analysis
▪ Locked states – No end product failure analysis via debugger is possible
Product Developer - Secure
Failure analysis request
Renesas
Failure analysis
Chip Manufacture (CM)
Secure System Debug (SSD)
RMA
SECDBG
Non-secure System Debug (NONSECDBG)
NONSECDBG
Deployed (DPL)
RMA
Lock Debug (LCK_DBG)
Lock Factory Boot (LCK_BOOT)
RMA Requested (RMA_REQ)
Renesas
RMA Acknowledged (RMA_ACK)
All erase
Transition without Authentication
Transition with Authentication

Figure 11. Device Lifecycle States in the Split Project Development Model

2.1.2 Combined Project Development Model

With the Combined Project Development Model, both secure and non-secure application development is carried by one trusted team. The Combined Project Development Model device lifecycle advancement and regression states are similar to the Split Project Development Model with key differences as noted below:

- The Product Developer will develop both secure and non-secure systems in SSD state.
- The Product Developer has access to the entire MCU resource – secure and non-secure.

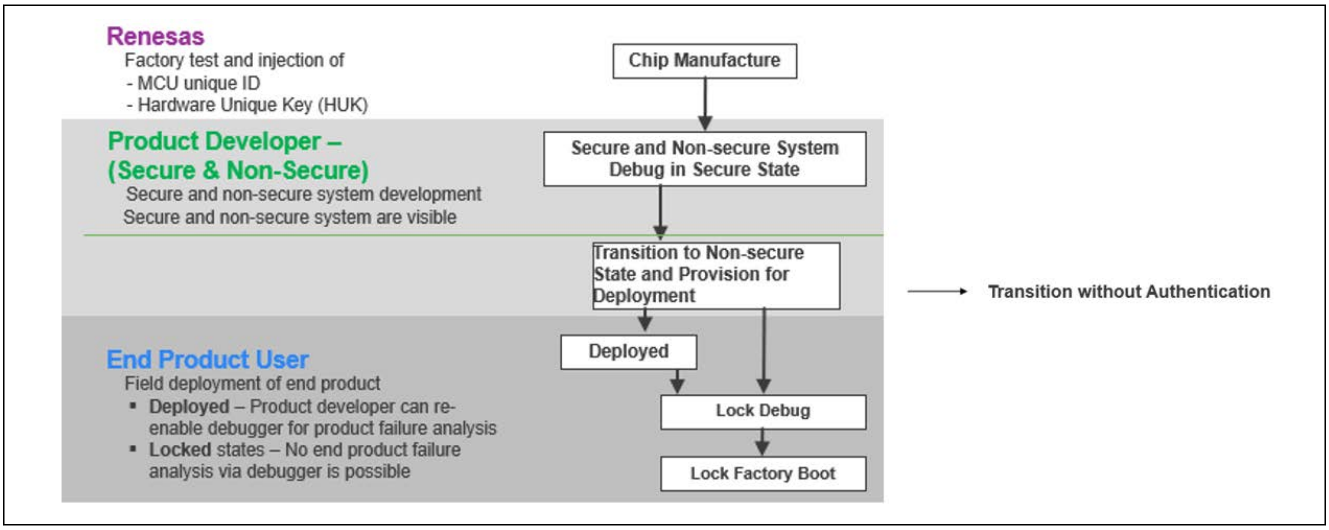

Renesas
Factory test and injection of
- MCU unique ID
- Hardware Unique Key (HUK)
Chip Manufacture
Product Developer –
(Secure & Non-Secure)
Secure and non-secure system development
Secure and non-secure system are visible
Secure and Non-secure System Debug in Secure State
Transition to Non-secure State and Provision for Deployment
Transition without Authentication
End Product User
Field deployment of end product
▪ Deployed – Product developer can re-enable debugger for product failure analysis
▪ Locked states – No end product failure analysis via debugger is possible
Deployed
Lock Debug
Lock Factory Boot

Figure 12. Combined Project Development Model

Summary of Device Lifecycle State Transitions during Development, Production, and Failure Analysis

Figure 13 describes all the possible transitions during the development, production flow, and failure analysis for the Combined Project Development Model.

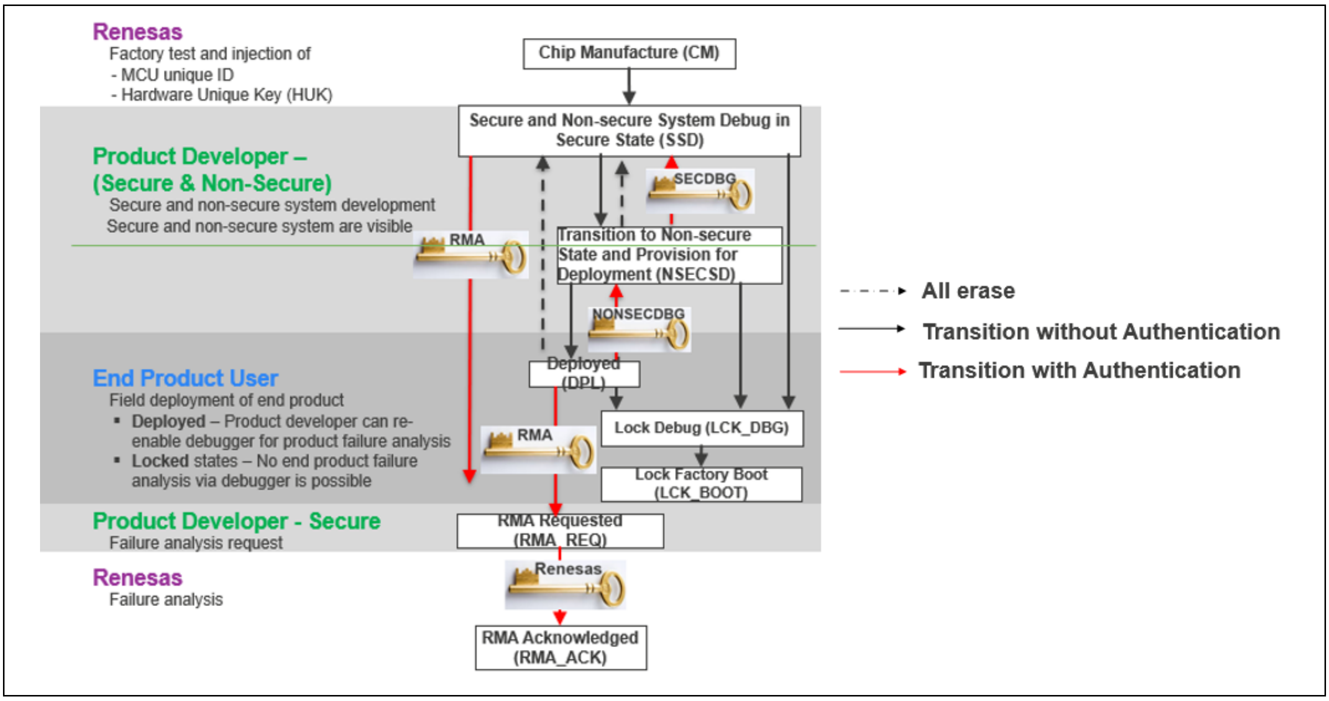

Renesas
Factory test and injection of
- MCU unique ID
- Hardware Unique Key (HUK)
Chip Manufacture (CM)
Secure and Non-secure System Debug in Secure State (SSD)
Product Developer –
(Secure & Non-Secure)
Secure and non-secure system development
Secure and non-secure system are visible
SECDBG
RMA
Transition to Non-secure State and Provision for Deployment (NSECSD)
NONSECDBG
All erase
Transition without Authentication
Transition with Authentication
End Product User
Field deployment of end product
▪ Deployed – Product developer can re-enable debugger for product failure analysis
▪ Locked states – No end product failure analysis via debugger is possible
Deployed (DPL)
RMA
Lock Debug (LCK_DBG)
Lock Factory Boot (LCK_BOOT)
Product Developer - Secure
Failure analysis request
RMA Requested (RMA_REQ)
Renesas
Failure analysis
Renesas
RMA Acknowledged (RMA_ACK)

Figure 13. Device Lifecycle States in Combined Project Development Model

2.2 Device Lifecycle State Transitions in the Secure Application Development Stage

The Split Project Development Model is used as an example in these analyses. Whenever there is a difference between the Split Project Development Model and the Combined Project Development Model, it will be explained explicitly.

As shown in Figure 11, as one of the final steps of the MCU manufacturing, Renesas will inject MCU Unique ID and HUK to the MCU. The MCU is delivered to the secure development team in CM state.

As shown in Figure 10, the possible Device Lifecycle State transitions are:

- Prior to start of secure application development:
  CM -> SSD: no authentication needed, no flash erase.
- During development, the secure developer can execute the `All erase` command to recover the MCU. The resulting Device Lifecycle State is still SSD.
  SSD -> SSD: no authentication, erase entire flash except permanently locked block.
  Note that the secure developer can use RFP to disable the `All erase` command in SSD state.
- After injecting the RMA_KEY during SSD lifecycle state, the secure developer can use the RMA_KEY to change the lifecycle of the MCU to RMA_REQ state after successful challenge response authentication.
  SSD -> RMA_REQ: authentication using RMA_KEY, entire flash erased except permanently locked block.
- After the secure application is developed and tested:
  SSD -> NSECSD: no authentication, no flash erase
  If the development follows the Combined Project Development Model, the product developer will finish both secure and non-secure application development in SSD state.

Device Lifecycle State Regression from NSECSD to SSD with Authentication

Having previously installed SECDBG_KEY in SSD state, the non-secure developer can return the MCU to the secure developer for failure analysis:

- The secure developer can use SECDBG_KEY to change the MCU state from NSECSD to SSD and perform the corresponding debug or failure analysis.
- This state transition is an authenticated transition and no flash erase happens in this state change.
  NSECSD -> SSD: Authentication with SECDBG_KEY, no flash erase.

2.3 Device Lifecycle State Transitions in the Non-Secure Application Development Stage

The Split Project Development Model is used as an example in these analyses. Whenever there is a difference between the Split Project Development Model and the Combined Project Development Model, it will be explained explicitly.

For Split Project Development, in the Non-Secure Application Development stage, the non-secure developer receives the MCU in NSECSD state and proceeds with non-secure application development. In NSECSD state, only debugging of the non-secure application is possible, secure system is protected from non-secure developer access.

Device Lifecycle State Change from NSECSD to SSD without Authentication

- If there is need to erase the entire flash to recover the MCU, the non-secure developer can erase the entire flash to regress the MCU lifecycle state from NSECSD to SSD. But the non-secure developer needs to return the MCU to the secure developer to reprogram the secure application.
  NSECSD -> SSD: erase entire flash
  A non-secure developer can use RFP to disable the `All erase` command in SSD state.
- This transition is typically not needed for the Combined Development Model as the Development is typically carried out in SSD for both Secure and Non-Secure Projects.

Device Lifecycle State Regression from DPL to NSECSD with Authentication

- As described in Table 1, if the End Product device lifecycle is set to DPL and the NONSECDBG_KEY is injected in the MCU during the NSECSD state, the End Product User has the option to return the End Product to the non-secure development team for failure analysis. The non-secure development team can use the NONSECDBG_KEY to regress the device lifecycle state to NSECSD state for non-secure application debug.

  DPL -> NSECSD: authentication using the NONSECDBG_KEY, no flash erase

Device Lifecycle State Change from NSECSD to DPL without Authentication

- During Prototyping stage, a non-secure developer can change the device lifecycle state from NSECSD to DPL to disable debugger access to the data and code.

  NSECSD -> DPL: no authentication, no flash erase

2.4 Device Lifecycle State Transitions in the Production Flow

The Device Lifecycle State Transitions for the Split Project Development Model and Combined Project Development Model are described as follows:

Split Project Development Model

For mass production where the device is not expected to be debugged or updated once deployed, the product manufacturer will typically perform the following lifecycle state transitions:

- At the secure product manufacturer:
  - **CM -> SSD**
  - Program the secure application
  - **SSD->NSECSD**

- At the non-secure product manufacturer:
  - Program the non-secure application
  - **NSECSD -> LCK_DBG**
  - **LCK_DBG -> LCK_BOOT**

Combined Project Development Model

For mass production where the device is not expected to be debugged or updated once deployed, the product manufacturer will typically perform the following lifecycle state transitions:

- At the product manufacturer:
  - **CM -> SSD**
  - Program both secure and non-secure application
  - **SSD -> LCK_DBG**
  - **LCK_DBG -> LCK_BOOT**

Flat Project Development Model

For mass production where the device is not expected to be debugged or updated once deployed, the application team can decide to protect portions of the code in the secure region. The product manufacturer then follows the same lifecycle state transitions as the Combined Project Development Model.

2.5 Possible Device Lifecycle State Transitions Initiated by the End User

The End Product User may receive the MCU in one of the following states: DPL, LCK_DBG or LCK_BOOT.

Case 1: End Product is in DPL State

- If there is need to erase the entire flash to recover the MCU, the end user can erase the entire flash to regress the MCU lifecycle state from DPL to SSD. The development can regress to secure application development stage.
  DPL -> SSD: entire flash erase, no authentication

Transitioning from DPL to Renesas Return Material Request (RMA) State

- As described in Table 1, if the End Product device lifecycle is set to DPL and there is need to return the end product to Renesas for failure analysis; the end user can use the MCU unique ID to change the MCU state from DPL to RMA_REQ. The end user can also return the product to the secure developer and the secure developer can use the MCU unique ID or the RAM_KEY (if previously installed) to transition the device state from DPL to RAM_REQ.
  DPL -> RMA_REQ: use MCU Unique ID or RMA_KEY

Note on Transition to RMA_REQ

- In the lifecycle transition to RMA_REQ, the contents on the flash memory are erased except for the permanently locked block or setting of the BPS_SEL register.
- The contents in the permanently locked block or register can be read by Renesas at failure analysis.
- **Please do not exercise this transition unless the end user wants to return the MCU to Renesas for failure analysis.**

Case 2: End Product is in LCK_BDG State

If the End Product device lifecycle is set to LCK_DBG, the debug interface of the device is permanently disabled, but the serial programming interface is still available. However, the serial programming interface cannot access the MCU code/data flash area, thus keeping the end user's application protected. The serial programming interface can still provide current MCU status, for example, bootloader version, device lifecycle states, and IDAU region setup.

Case 3: End Product is in LCK_BOOT State

If the End Product device lifecycle is set to LCK_BOOT, the device cannot regress to any other state. In the LCK_BOOT state, the debug and serial programming interface are permanently disabled, and the device cannot regress to any other state. Exercise caution before making decisions to transition to the LCK_BOOT state.

2.6 Notes on Flat Project Development Model

When using the Flat Project Development Model, during the Development stage, the user does not need to develop the application with Arm® TrustZone® awareness. During production, the user can choose a region to protect as the secure region based on their application and set up the IDAU regions accordingly. Once the IDAU regions are set up, the user can follow the same production flow as the Combined Project Development Model. Refer to section 2.4 for details on the production flow.

3. DLM Key Creation and Installation Procedure

3.1 Wrapped Key Installation Overview

The information provided in this section applies to both Device Lifecycle Management Keys and Cryptographic User Keys. Note that in the *Renesas RA6M4 Group User's Manual: Hardware*, both the Device Lifecycle Management Keys and Cryptographic User Keys are referred to as "User Key". In this application note, we use DLM Key to explain the installation procedure.

There are three steps required to install the DLM keys into the MCU.

1. The customer creates the 128 bits installation key. This key is called User Factory Programming Key (UFPK) and is used to encrypt a user key. The customer gets the key of the wrapped version (W-UFPK) through the Renesas Key Wrapping Service. Section 3.2 walks the user through the wrapping process.

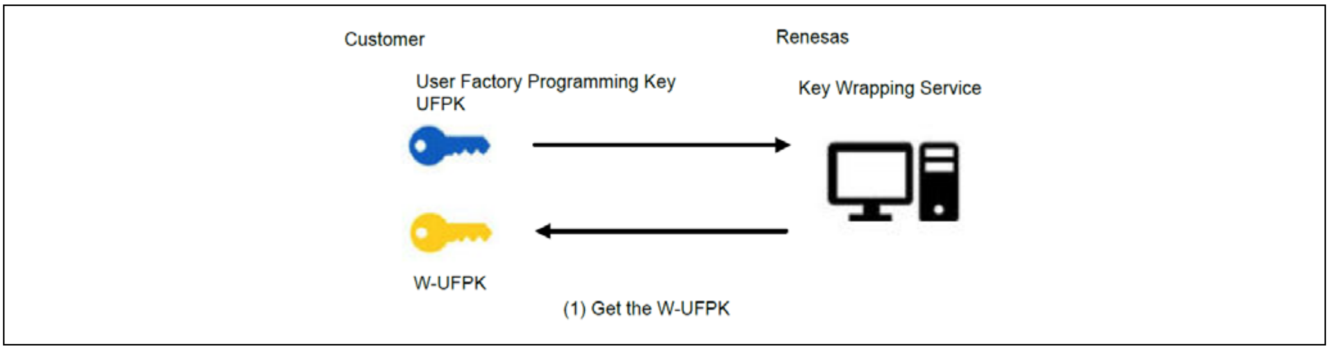


Figure 14. Wrapping the User Factory Programming Key (UFPK)

2. The customer encrypts the user key using UFPK as the AES key. Section 3.4 provides information for the user to reference and generate the encrypted UFPK key.

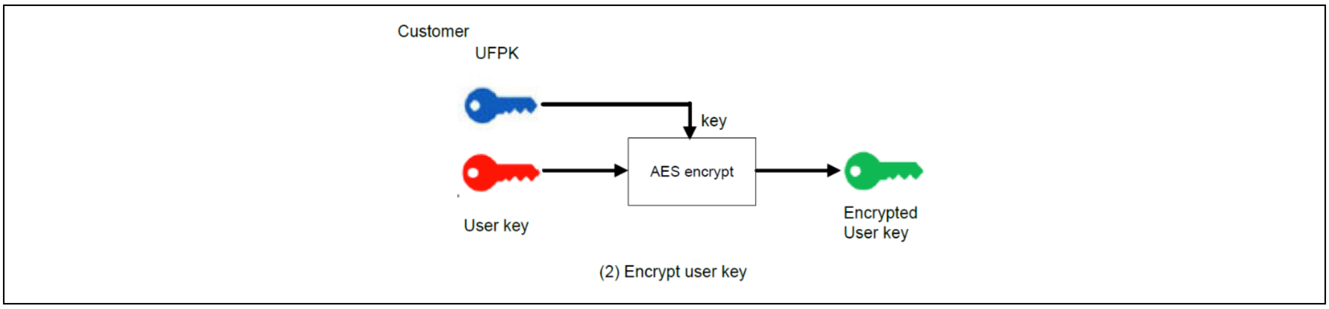


Figure 15. Encrypt the DLM Key Using UFPK

3. Customer sends W-UFPK (generated in step 1) and the encrypted user key (generated in step 2) to the MCU by using the serial programming interface. The sent user key is decrypted, wrapped with the hardware unique key (HUK), and then stored in the nonvolatile memory. Section 3.5 walks the user through the key installation process using RFP.

Section 3.6 provides information on how to use the DLM key to perform authenticated MCU device lifecycle transitions.

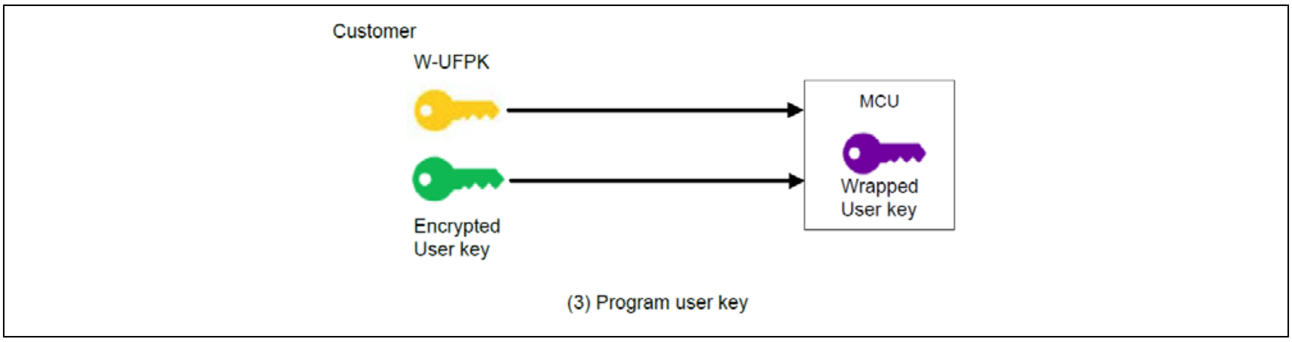


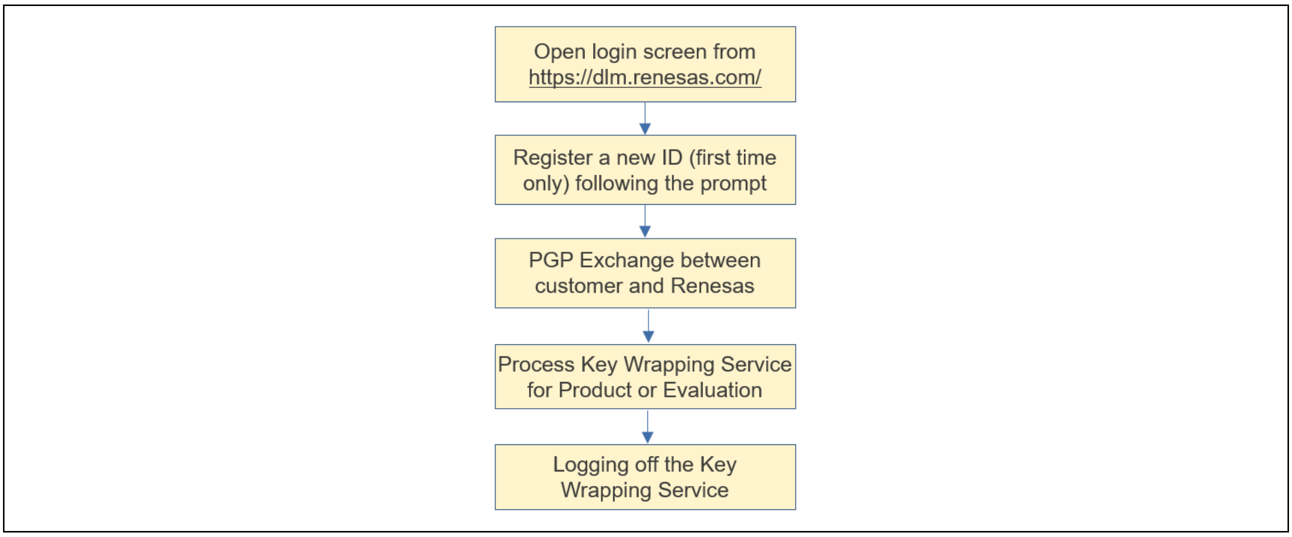
Figure 16. Created Wrapped DLM Keys

3.2 Create Customer PGP Key Pair and Exchange Public Key with Renesas

Exchanging a PGP public key between the customer and Renesas is needed as all information transmitted to and from the DLM server uses PGP encryption. This is a one-time process prior to establishing communication with the DLM server.

3.2.1 Overview of Device Lifecycle Management (DLM) Server

The following is the general operational flow of using the DLM server for DLM Key wrapping service. The user needs to go to **https://dlm.renesas.com/** in a web browser to access the Renesas DLM server.


Open login screen from https://dlm.renesas.com/

Register a new ID (first time only) following the prompt

PGP Exchange between customer and Renesas

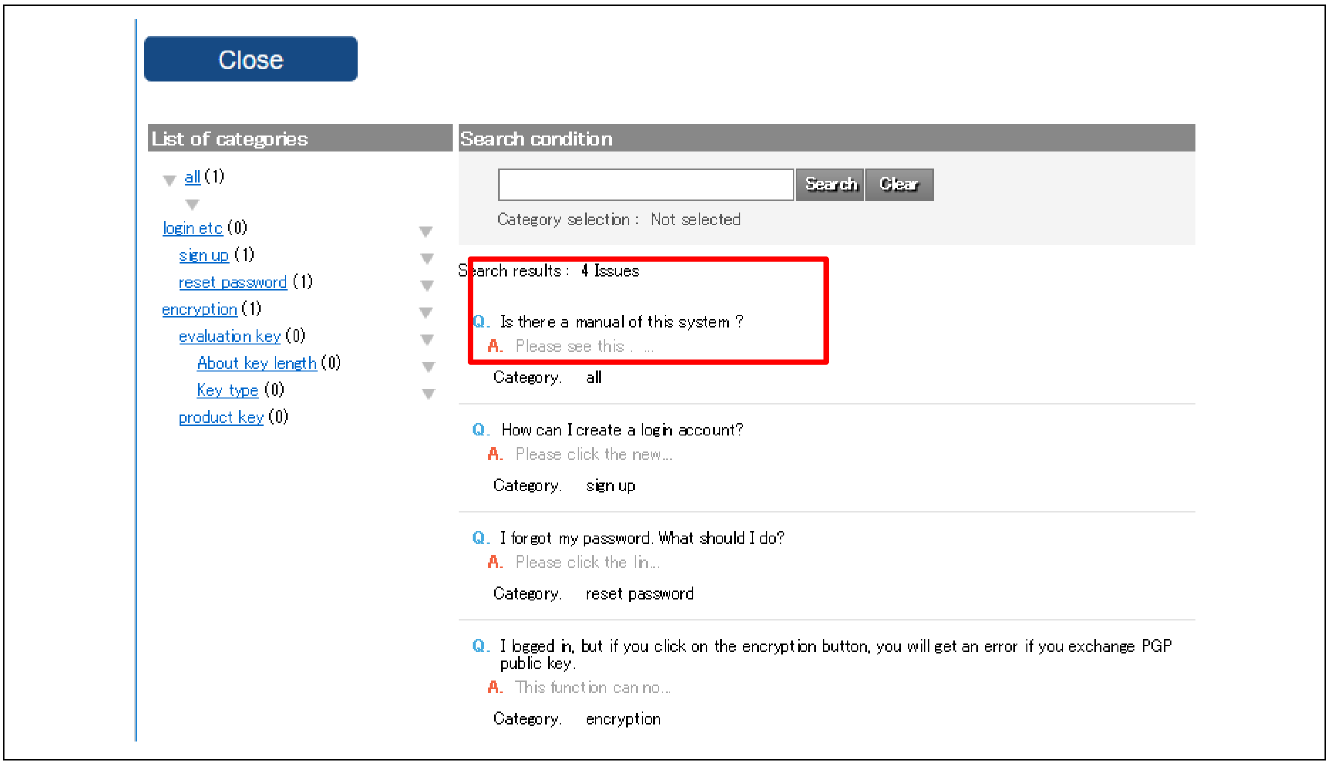
Process Key Wrapping Service for Product or Evaluation

Logging off the Key Wrapping Service

Figure 17. Operational Flow of Using Key Wrapping Using DLM Server

DLM Server FAQ and User's Manual

Note that once the https://dlm.renesas.com/ webpage is opened, the user can click on the FAQ link on the right. The user can find the link to the DLM server user's manual in the answer for the first FAQ question, "Is there a manual of this system?" as shown in Figure 11.



List of categories

- all (1)
- login etc (0)
  - sign up (1)
  - reset password (1)
- encryption (1)
  - evaluation key (0)
    - About key length (0)
    - Key type (0)
  - product key (0)

Search condition

Category selection : Not selected

Search results : 4 Issues

Q. Is there a manual of this system ?
A. Please see this . ...
Category. all

Q. How can I create a login account?
A. Please click the new...
Category. sign up

Q. I forgot my password. What should I do?
A. Please click the In...
Category. reset password

Q. I logged in, but if you click on the encryption button, you will get an error if you exchange PGP public key.
A. This function can no...
Category. encryption

Figure 18. DLM Server FAQ and User's Manual

DLM Server FAQ

As shown above, there is an FAQ to help customers with some common questions.

The information communicated between customer and DLM server need to be OpenPGP encrypted. The following is the overview of the operational flow for the Key Wrapping service with OpenPGP encryption as a security measure to protect the DLM Keys in transit.

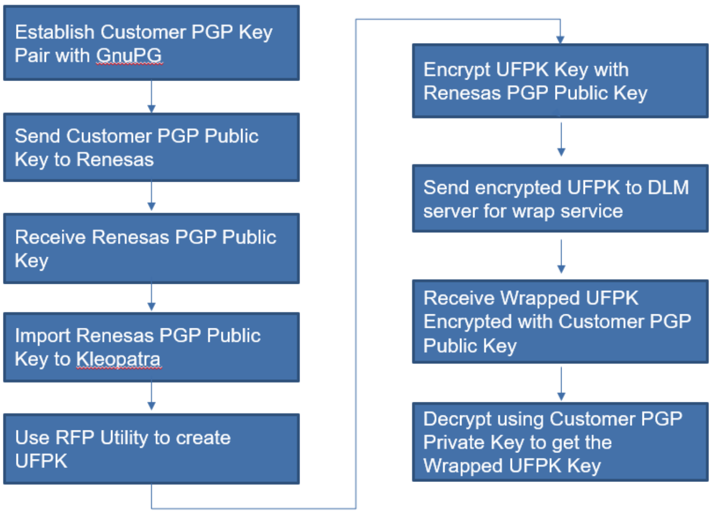

Establish Customer PGP Key Pair with GnuPG
Send Customer PGP Public Key to Renesas
Receive Renesas PGP Public Key
Import Renesas PGP Public Key to Kleopatra
Use RFP Utility to create UFPK
Encrypt UFPK Key with Renesas PGP Public Key
Send encrypted UFPK to DLM server for wrap service
Receive Wrapped UFPK Encrypted with Customer PGP Public Key
Decrypt using Customer PGP Private Key to get the Wrapped UFPK Key

Figure 19. Overview of DLM Key Wrapping Service Using PGP

3.2.2 Establish Customer PGP Key Pair

Use the following steps to create a customer OpenPGP key pair. This Application Note uses Gpg4win, the official GnuPG distribution for Windows®, for the PGP key generation, encryption, and decryption service. Note that the customer needs to take security measures to protect the DLM Keys from theft and leaking while going through the KeyWrap services.

1. Download PGP Software from the following website http://www.gpg4win.org/

2. Install and launch the application using the Kleopatra shortcut: Kleopatra.

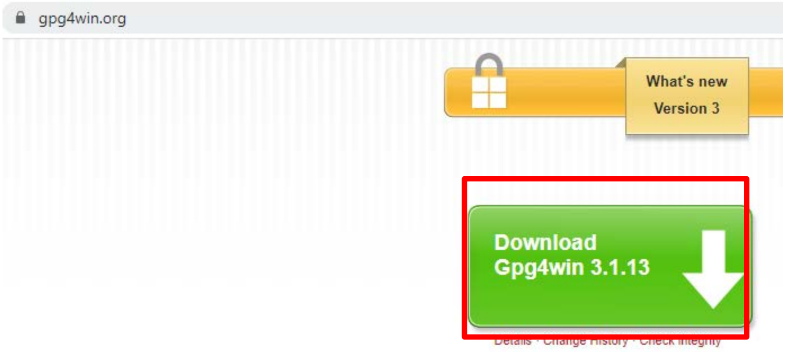
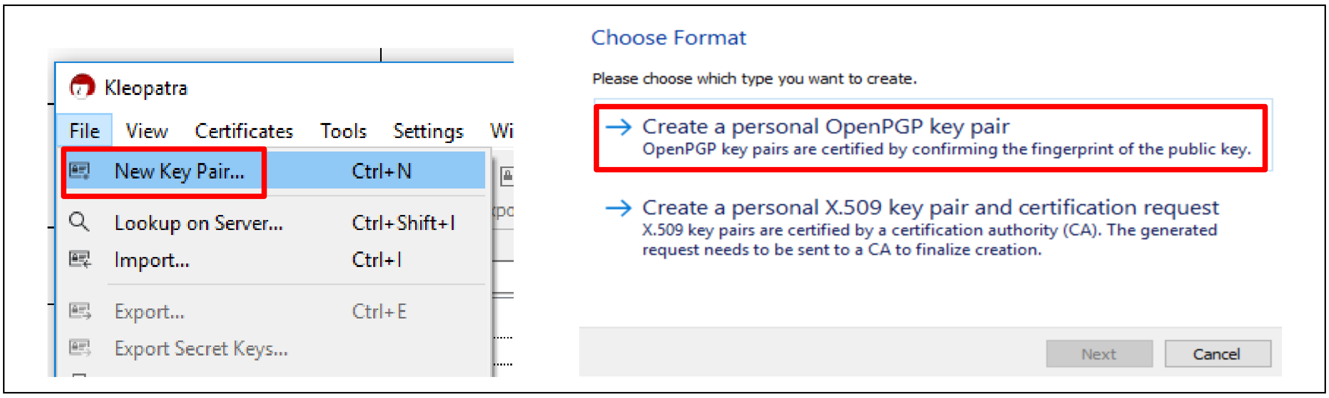


Figure 20. Download GPG Software

3. Click **File** > **New Key Pair** and choose format **Create a personal OpenPGP key pair**.



Choose Format

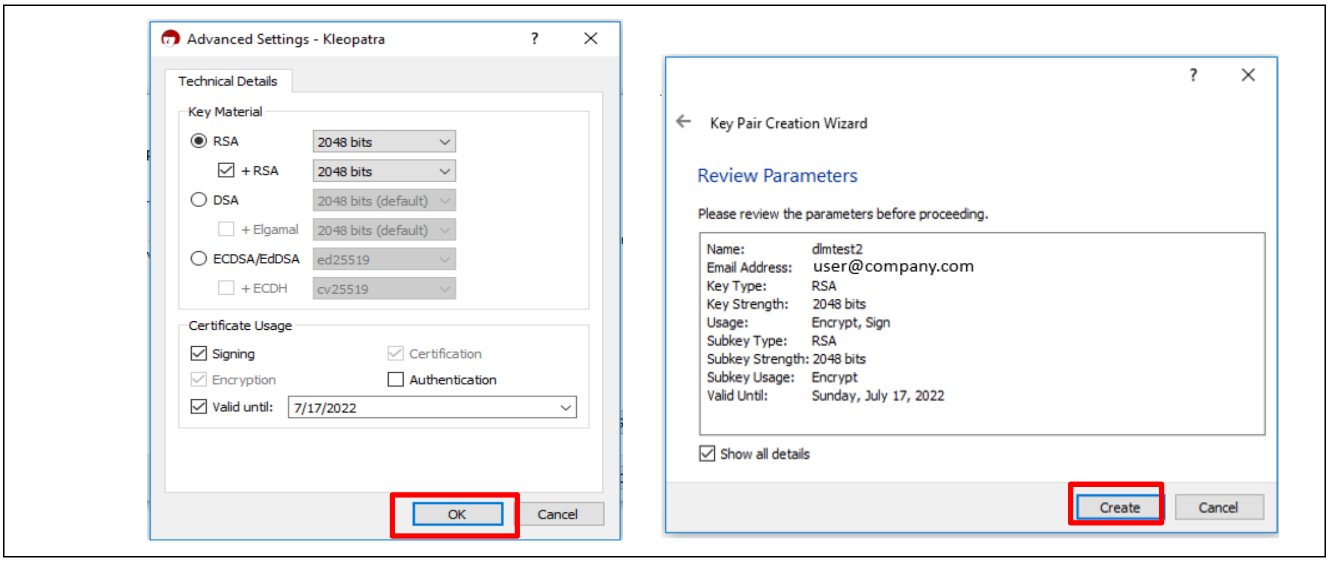
Please choose which type you want to create.

→ Create a personal OpenPGP key pair
OpenPGP key pairs are certified by confirming the fingerprint of the public key.

→ Create a personal X.509 key pair and certification request
X.509 key pairs are certified by a certification authority (CA). The generated request needs to be sent to a CA to finalize creation.

Figure 21. Generate PGP Key Pair

4. Click **Advanced Settings** to view the **Technical Details**. Click **OK** and then Click **Next**.



Key Material

● RSA 2048 bits
☑ + RSA 2048 bits
○ DSA 2048 bits (default)
□ + Elgamal 2048 bits (default)
○ ECDSA/EdDSA ed25519
□ + ECDH cv25519

Certificate Usage

☑ Signing ☑ Certification
☑ Encryption □ Authentication
☑ Valid until: 7/17/2022

Review Parameters

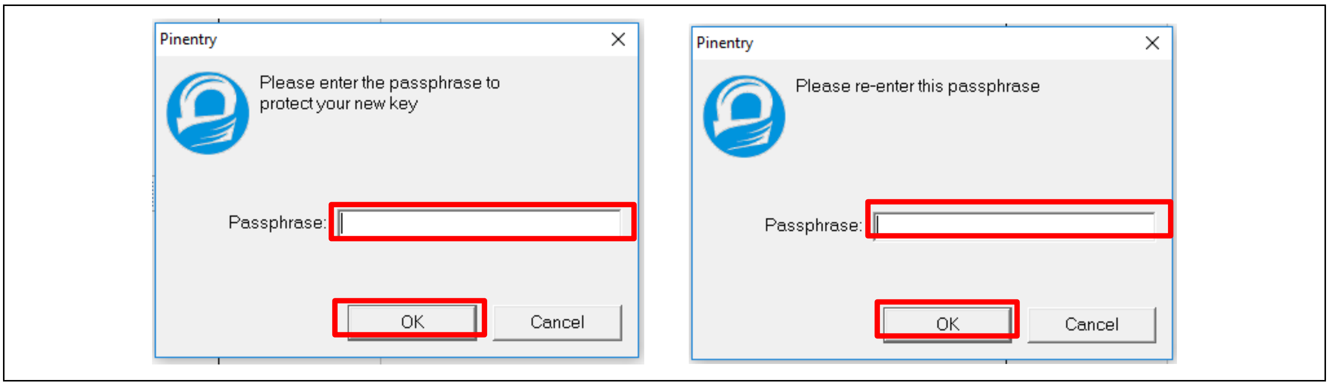
Please review the parameters before proceeding.

Name:	dlmtest2
Email Address:	user@company.com
Key Type:	RSA
Key Strength:	2048 bits
Usage:	Encrypt, Sign
Subkey Type:	RSA
Subkey Strength:	2048 bits
Subkey Usage:	Encrypt
Valid Until:	Sunday, July 17, 2022

☑ Show all details

Figure 22. Advanced Settings for the PGP Key pair

5. Provide a passphrase to protect the private key. Make sure to save your passphrase for later use.



Please enter the passphrase to protect your new key

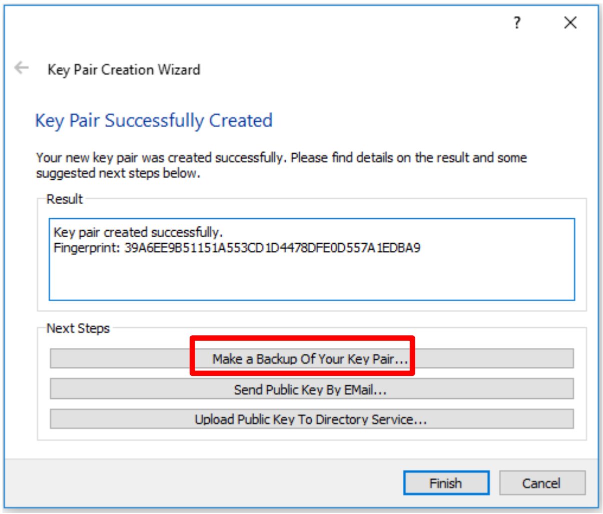
Passphrase:

Please re-enter this passphrase

Passphrase:

Figure 23. Provide Passphrase

6. Observe that the PGP Key Pair is created successfully.



Key Pair Creation Wizard

Key Pair Successfully Created

Your new key pair was created successfully. Please find details on the result and some suggested next steps below.

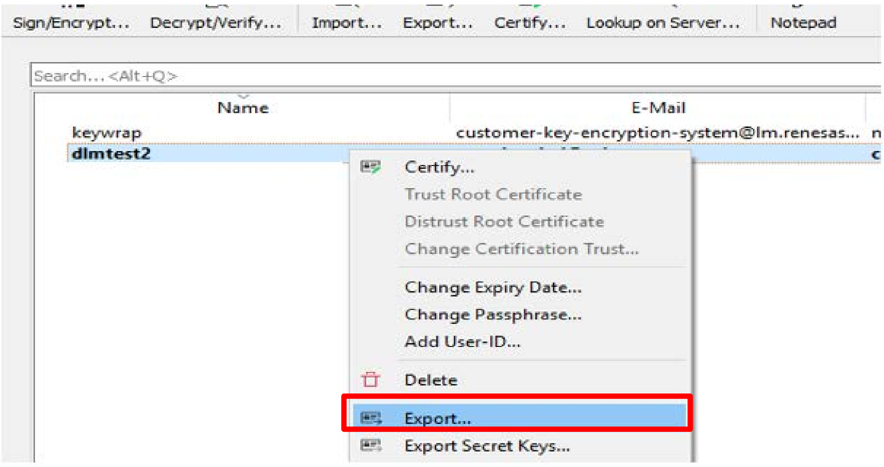
Result

Key pair created successfully.
Fingerprint: 39A6EE9B51151A553CD1D4478DFE0D557A1EDBA9

Next Steps
Make a Backup Of Your Key Pair...
Send Public Key By EMail...
Upload Public Key To Directory Service...
Finish
Cancel

Figure 24. Key Pair Successfully Created

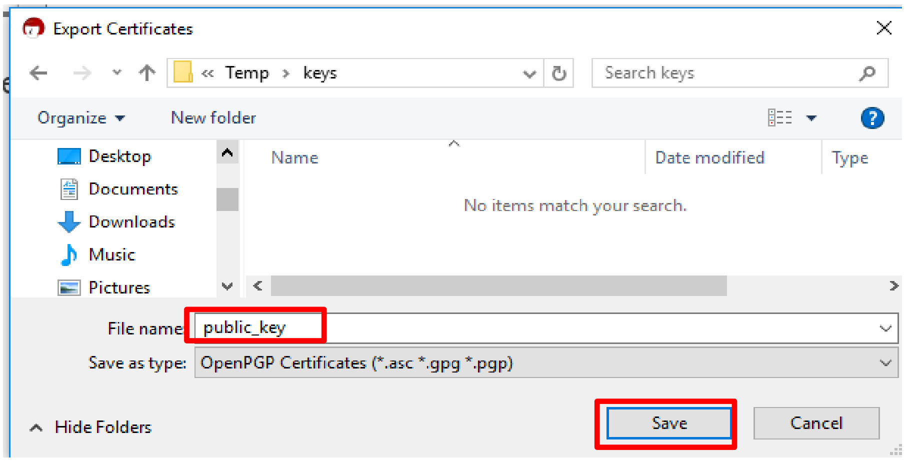
7. Use Kleopatra to export the customer public key as shown below.



Sign/Encrypt... Decrypt/Verify... Import... Export... Certify... Lookup on Server... Notepad
Search...<Alt+Q>
Name
E-Mail
keywrap
customer-key-encryption-system@lm.renesas...
dlmtest2
Certify...
Trust Root Certificate
Distrust Root Certificate
Change Certification Trust...
Change Expiry Date...
Change Passphrase...
Add User-ID...
Delete
Export...
Export Secret Keys...

Figure 25. Export the Customer Public key

8. Save the customer public key to a file with the `*.asc` extension, for example “`public_key.asc`”.



File name: public_key

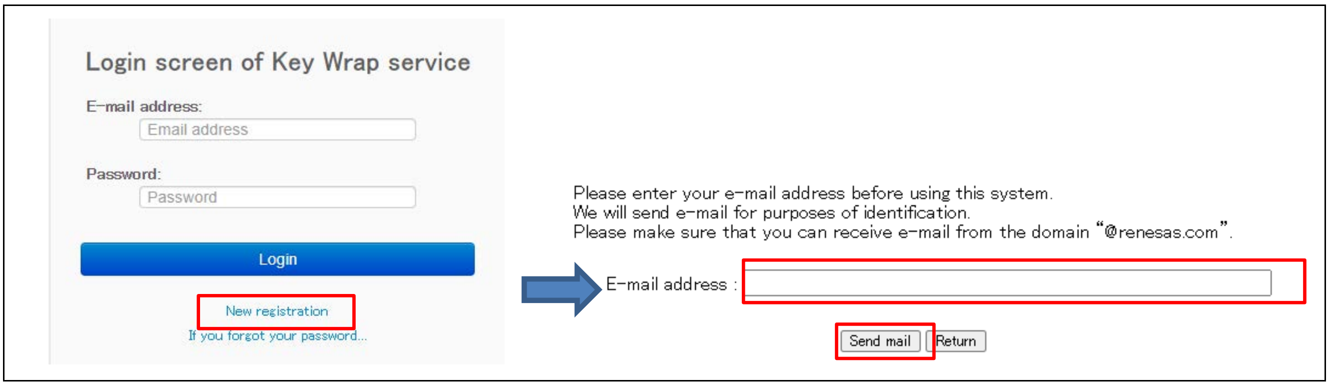
Save as type: OpenPGP Certificates (*.asc *.gpg *.pgp)

Figure 26. Save the Customer Public key

3.2.3 Registration with DLM Server

This section provides a brief walk through of the DLM server new registration steps. This is a one-time process for new customers. The customer is encouraged to review the *New registration* section in the DLM user manual for further details on the new registration steps.

Open URL https://dlm.renesas.com/ in a browser and click **New registration**. Follow the prompt to provide an email address and click **Send mail**.


Login screen of Key Wrap service

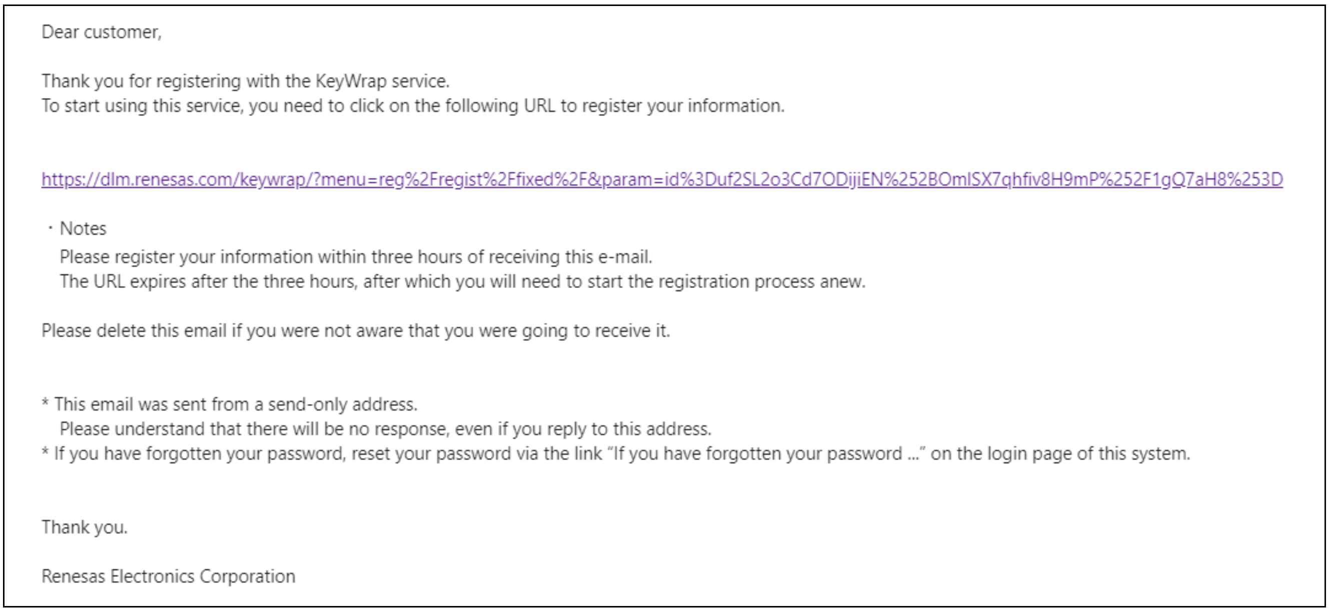
E–mail address:

Password:

Please enter your e–mail address before using this system.
We will send e–mail for purposes of identification.
Please make sure that you can receive e–mail from the domain "@renesas.com".

Figure 27. Provide Email Address for New Registration

You will receive the link for registration in email as shown below.


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://dlm.renesas.com/keywrap/?menu=reg%2Fregist%2Ffixed%2F&param=id%3Duf2SL2o3Cd7ODijiEN%252BOmlSX7qhfiv8H9mP%252F1gQ7aH8%253D

・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.

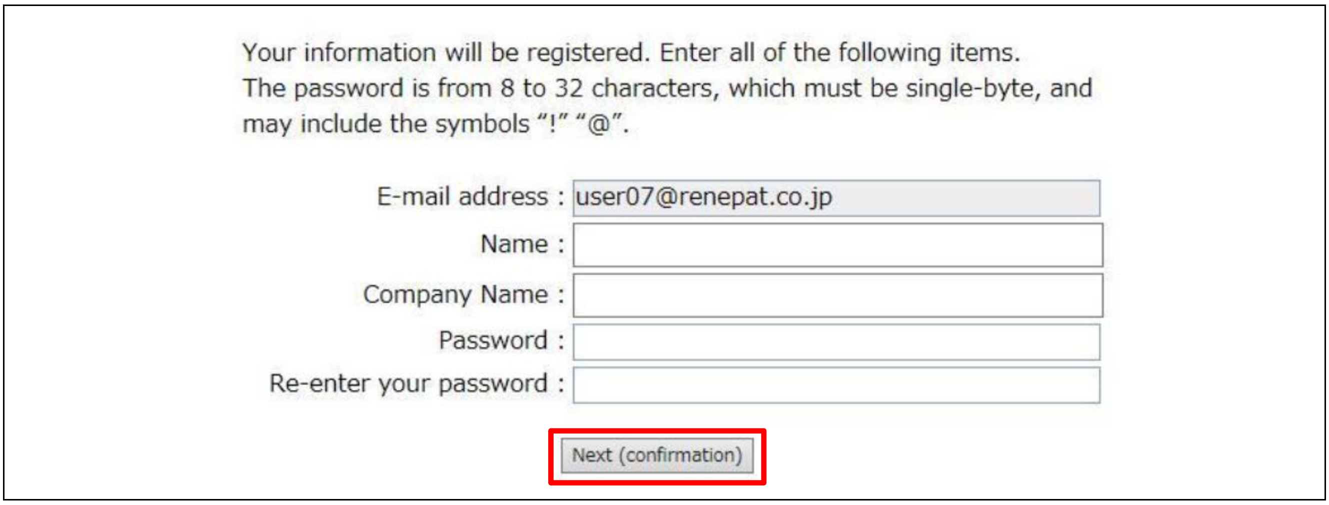
* This email was sent from a send-only address.
Please understand that there will be no response, even if you reply to this address.
* If you have forgotten your password, reset your password via the link "If you have forgotten your password ..." on the login page of this system.

Thank you.

Renesas Electronics Corporation

Figure 28. Email with Registration Link

Click on the URL in the confirmation mail and provide your name, company name, password, and re-enter the password. Click **Next (confirmation)** button. After the confirmation screen is displayed, click on the **Registration** button to complete the user registration.



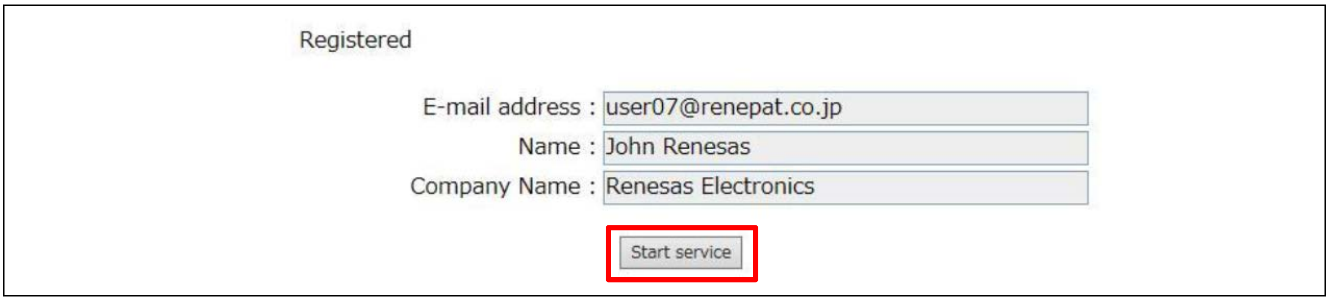
Your information will be registered. Enter all of the following items.
The password is from 8 to 32 characters, which must be single-byte, and
may include the symbols "!" "@".

E-mail address : user07@renepat.co.jp
Name :
Company Name :
Password :
Re-enter your password :

Figure 29. Register Customer Information

3.2.4 PGP Public Key Exchange Between Customer and Renesas

Once you have successfully registered the customer information, the following screen will open. Click on the **Start service** button to start using the customer's key encryption system.

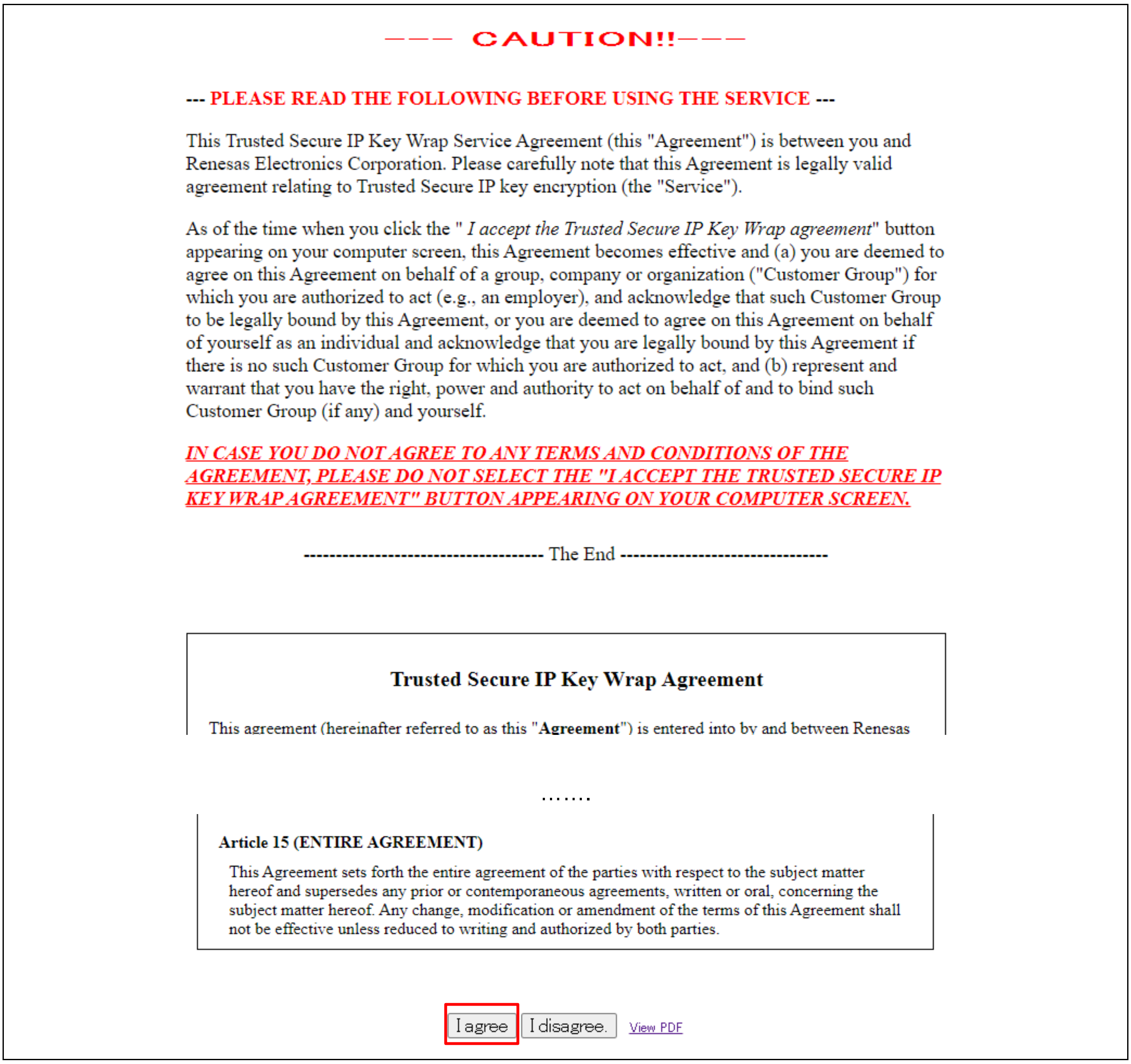


Registered

E-mail address : user07@renepat.co.jp
Name : John Renesas
Company Name : Renesas Electronics

Figure 30. Start Using the DLM Service

Accept the **Trusted Secure IP Key Wrap Agreement** as shown below by clicking **Agree**. Note that the following Agreement will come up every time the customer logs 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").

As of the time when you click the " *I accept the Trusted Secure IP Key Wrap agreement*" button appearing on your computer screen, this Agreement becomes effective and (a) you are deemed to agree on this Agreement on behalf of a group, company or organization ("Customer Group") for which you are authorized to act (e.g., an employer), and acknowledge that such Customer Group to be legally bound by this Agreement, or you are deemed to agree on this Agreement on behalf of yourself as an individual and acknowledge that you are legally bound by this Agreement if there is no such Customer Group for which you are authorized to act, and (b) represent and warrant that you have the right, power and authority to act on behalf of and to bind such Customer Group (if any) and yourself.

IN CASE YOU DO NOT AGREE TO ANY TERMS AND CONDITIONS OF THE AGREEMENT, PLEASE DO NOT SELECT THE "I ACCEPT THE TRUSTED SECURE IP KEY WRAP AGREEMENT" BUTTON APPEARING ON YOUR COMPUTER SCREEN.

------------------------------------ The End -------------------------------

Trusted Secure IP Key Wrap Agreement

This agreement (hereinafter referred to as this "**Agreement**") is entered into by and between Renesas

....

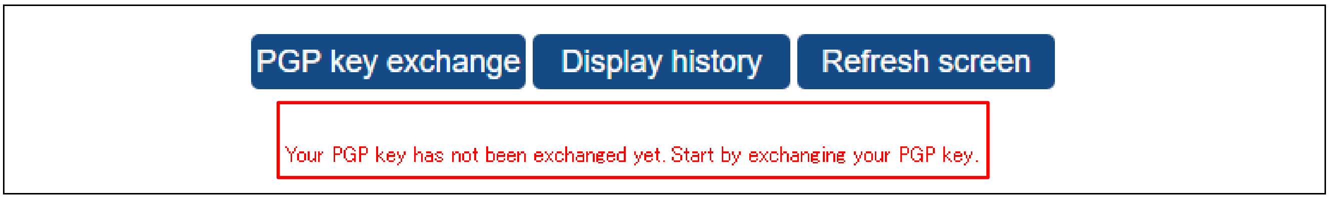
Article 15 (ENTIRE AGREEMENT)

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.

Figure 31. Agree with the Key Wrap Agreement

Communication between the user and the DLM server uses PGP to encrypt all data exchanged. Prior to providing the DLM keys to the DLM sever, there needs to be a PGP key exchange between the customer and Renesas beforehand. The PGP key exchange can be achieved via the **PGP key exchange** button as shown in Figure 32.

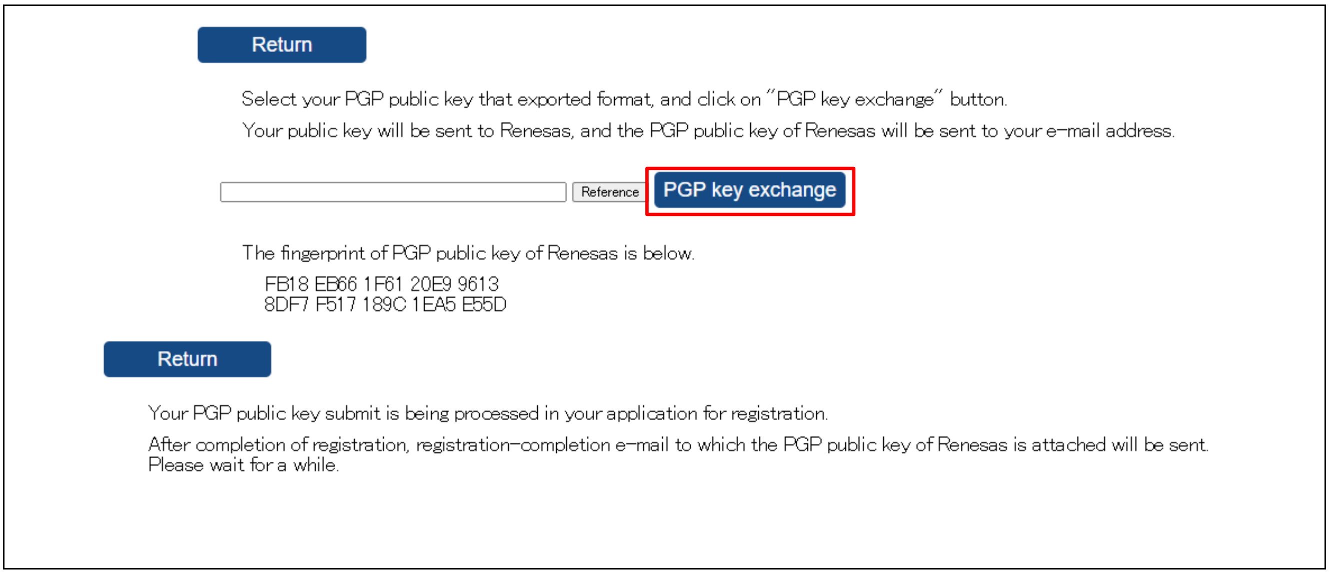
Prior to exchanging PGP keys with the Renesas DLM server, the customer will see the message shown as follows in the red box after logging in. Once the PGP key exchange is successful, the red text, **Your PGP key has not been exchanged yet. Start by exchanging your PGP key** will not show up anymore.



Your PGP key has not been exchanged yet. Start by exchanging your PGP key.

Figure 32. Request for PGP Key Exchange

Click the **PGP key exchange** button. The user interface shown in Figure 33 will open. Click **Reference** as shown in Figure 33 and select the customer public key exported from section 3.2.2 as shown in Figure 26 (`public_key.asc`). Next, click **PGP key exchange** in the interface shown in Figure 33 and wait to receive the Renesas PGP public key from the customer's email address provided at the New user registration stage.



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.

The fingerprint of PGP public key of Renesas is below.

FB18 EB66 1F61 20E9 9613
8DF7 F517 189C 1EA5 E55D

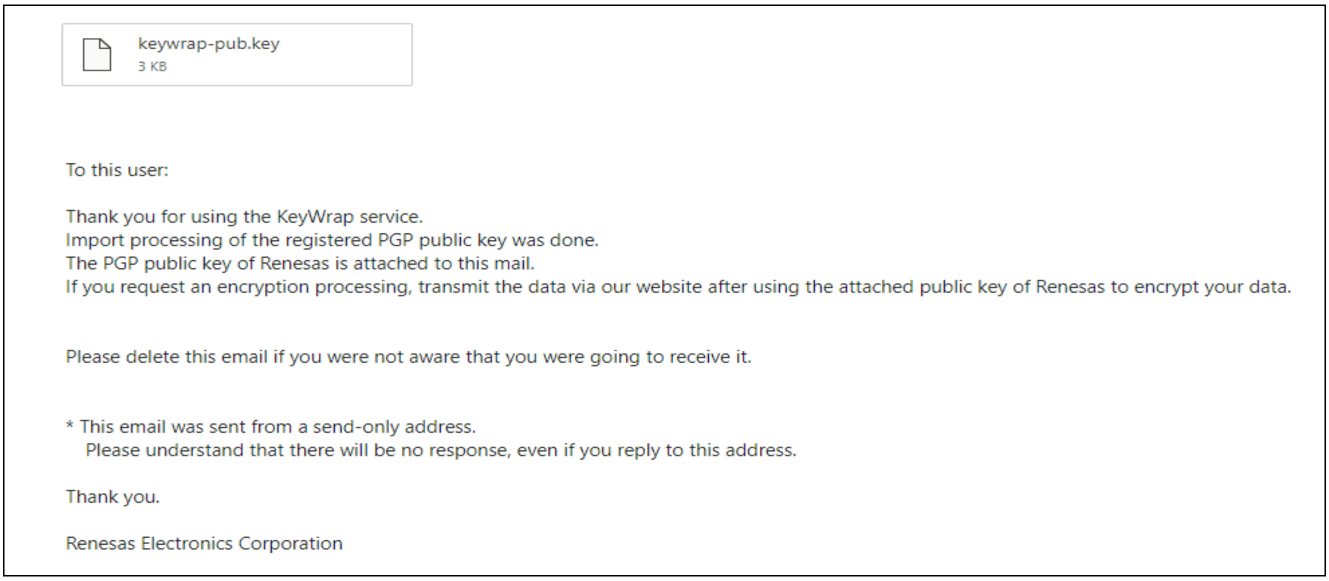
Your PGP public key submit is being processed in your application for registration.

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.

Figure 33. Provide Customer PGP Public Key to DLM Server

The customer will receive an email with content as shown in Figure 34 if the registration is successful.

Note that the PGP public key can be registered any number of times. If the key is registered multiple times, the latest PGP public key that has been registered successfully is used in encryption. All previous PGP public keys registered are discarded.



To this user:

Thank you for using the KeyWrap service.
Import processing of the registered PGP public key was done.
The PGP public key of Renesas is attached to this mail.
If you request an encryption processing, transmit the data via our website after using the attached public key of Renesas to encrypt your data.

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.

Thank you.

Renesas Electronics Corporation

Figure 34. Receive the Renesas PGP Public Key

Save the Renesas PGP public key received (`keywrap-pub.key`). This key will be used in the following sections.

3.2.5 Import Renesas PGP Public Key into Kleopatra

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

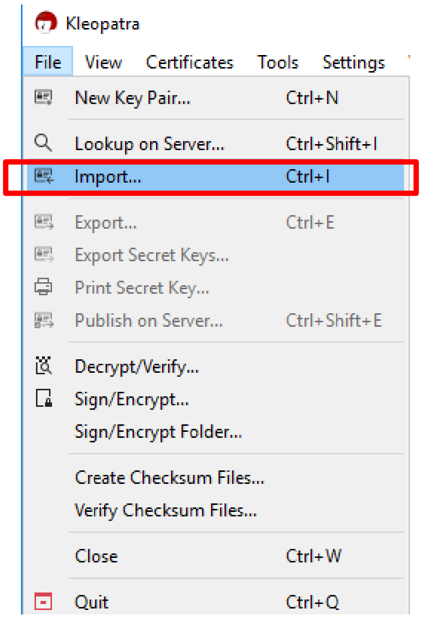
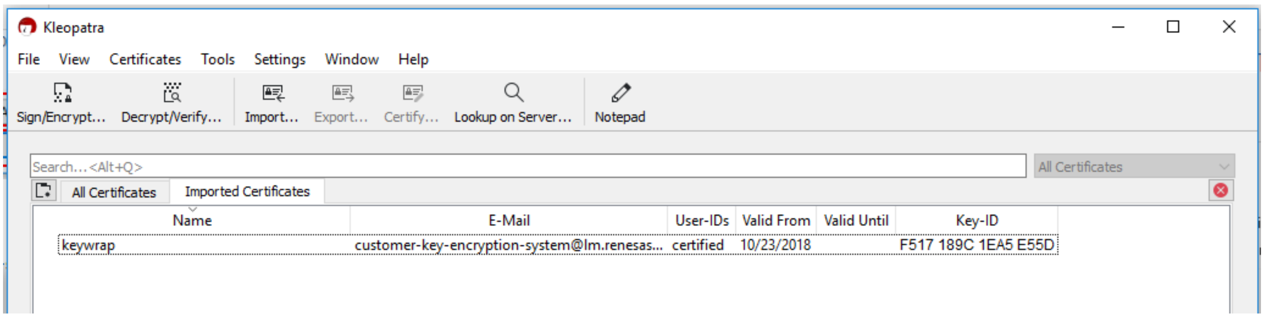


Figure 35. Import Renesas Public Key into Kleopatra

Select `keywrap-pub.key` saved from the previous step to import into Kleopatra.

The following item will come up in the Imported Certificates in Kleopatra and the Renesas public key is ready to be used.



Name	E-Mail	User-IDs	Valid From	Valid Until	Key-ID
keywrap	customer-key-encryption-system@lm.renesas...	certified	10/23/2018		F517 189C 1EA5 E55D

Figure 36. Renesas PGP Public Key

3.3 Creating the UFPK Using RFP and Wrapped UFPK Using DLM Sever

This section walks the user through the process for creating the Wrapped UFPK. This corresponds to the first steps in the DLM Key installation steps explained in section 3.

The DLM Keys can only be installed on the MCU in Wrapped format. As described in section 3.1, an UFPK and Wrapped UFPK need to be created first prior to the DLM Key creation.

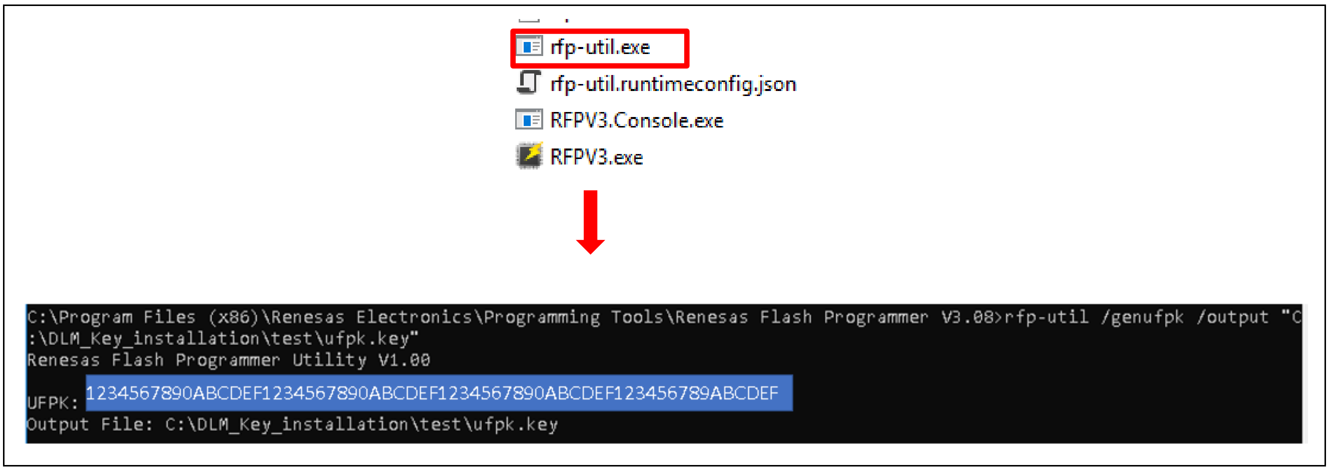
- The UFPK is wrapped by the Renesas DLM Server: https://dlm.renesas.com/. This webpage communicates with the user through HTTPS and PGP, thus offering security for protecting user data in transit.
- The DLM Key is wrapped when RFP installs the DLM Keys in the MCU.

3.3.1 Create User Factory Programming Key (UFPK)

We can use the `rfp-util.exe` included with RFP installation folder to generate UFPK. Open a command line window and activate the `rfp-util.exe` with an example shown as follows.

Example Command Line Input:

```
rfp-util /genufpk /output "C:\DLM_Key_Installation\test\ufpk.key"
```



```
C:\Program Files (x86)\Renesas Electronics\Programming Tools\Renesas Flash Programmer V3.08>rfp-util /genufpk /output "C
:\DLM_Key_installation\test\ufpk.key"
Renesas Flash Programmer Utility V1.00
UFPK: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF
Output File: C:\DLM_Key_installation\test\ufpk.key
```

Figure 37. Generate User Factory Programming Key

3.3.2 Encrypt UFPK with Renesas PGP Public Key

Select **Sign/Encrypt** from Kleopatra and select to encrypt this key using the Renesas PGP public key and sign with customer PGP key.

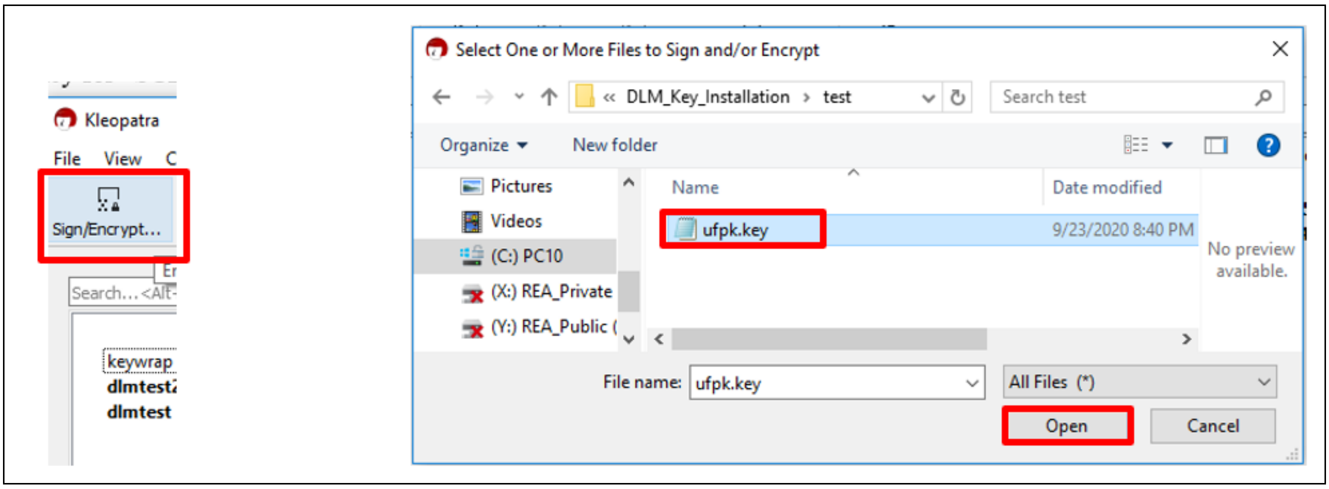
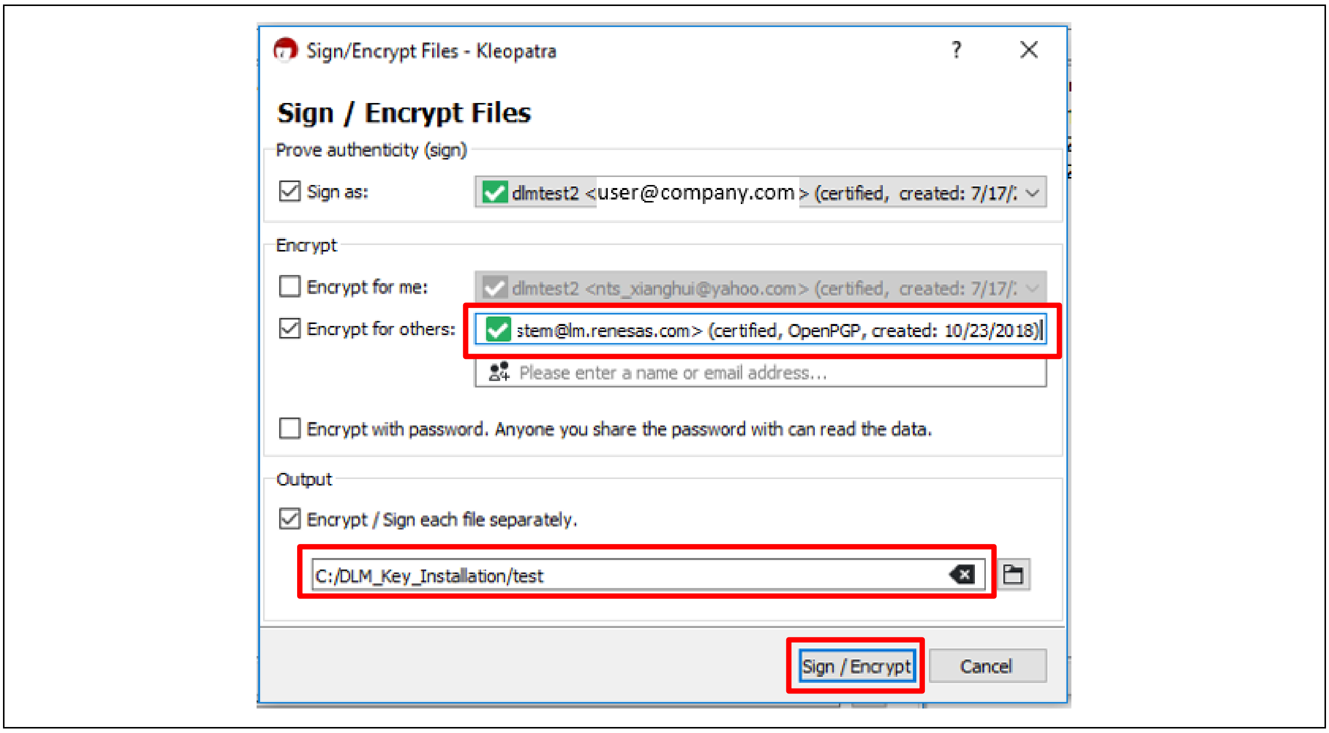


Figure 38. Select the UFPK Key to be Encrypted

Choose **Encrypt for others** and select the Renesas PGP Public key**.** Click **Sign/Encrypt**.



Sign/Encrypt Files - Kleopatra

Sign / Encrypt Files

Prove authenticity (sign)

☑ Sign as: dlmtest2 <user@company.com> (certified, created: 7/17/:

Encrypt

☐ Encrypt for me: dlmtest2 <nts_xianghui@yahoo.com> (certified, created: 7/17/:

☑ Encrypt for others: stem@lm.renesas.com> (certified, OpenPGP, created: 10/23/2018)

Please enter a name or email address...

☐ Encrypt with password. Anyone you share the password with can read the data.

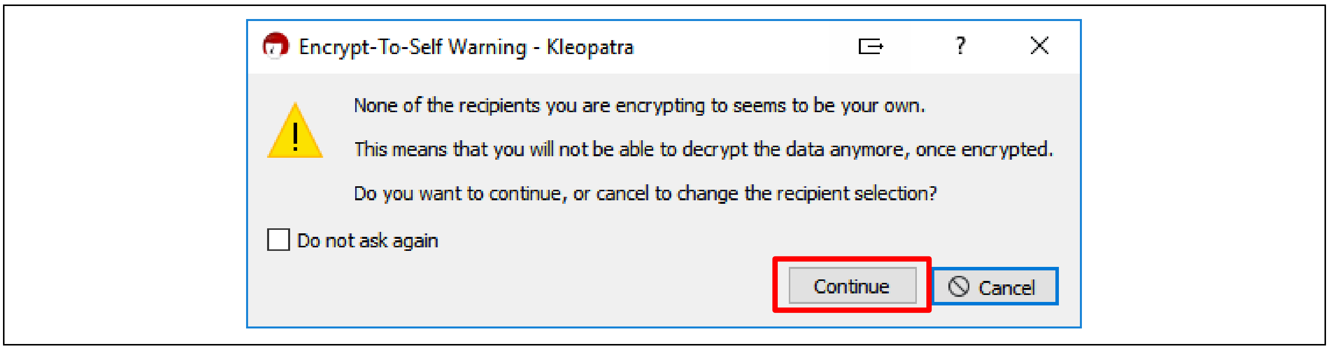
Output

☑ Encrypt / Sign each file separately.

C:/DLM_Key_Installation/test

Figure 39. Use Renesas Public Key to Encrypt UFPK

You will get an Encrypt-To-Self Warning that you cannot decrypt the data. Press **Continue**.



Encrypt-To-Self Warning - Kleopatra

None of the recipients you are encrypting to seems to be your own.

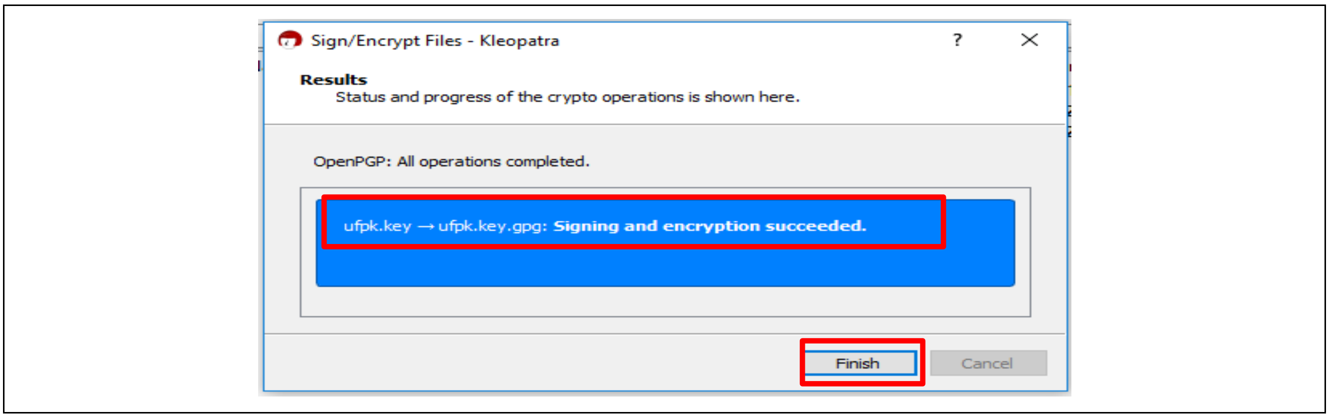
This means that you will not be able to decrypt the data anymore, once encrypted.

Do you want to continue, or cancel to change the recipient selection?

☐ Do not ask again

Figure 40. Confirm Encryption Option

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



Sign/Encrypt Files - Kleopatra

Results

Status and progress of the crypto operations is shown here.

OpenPGP: All operations completed.

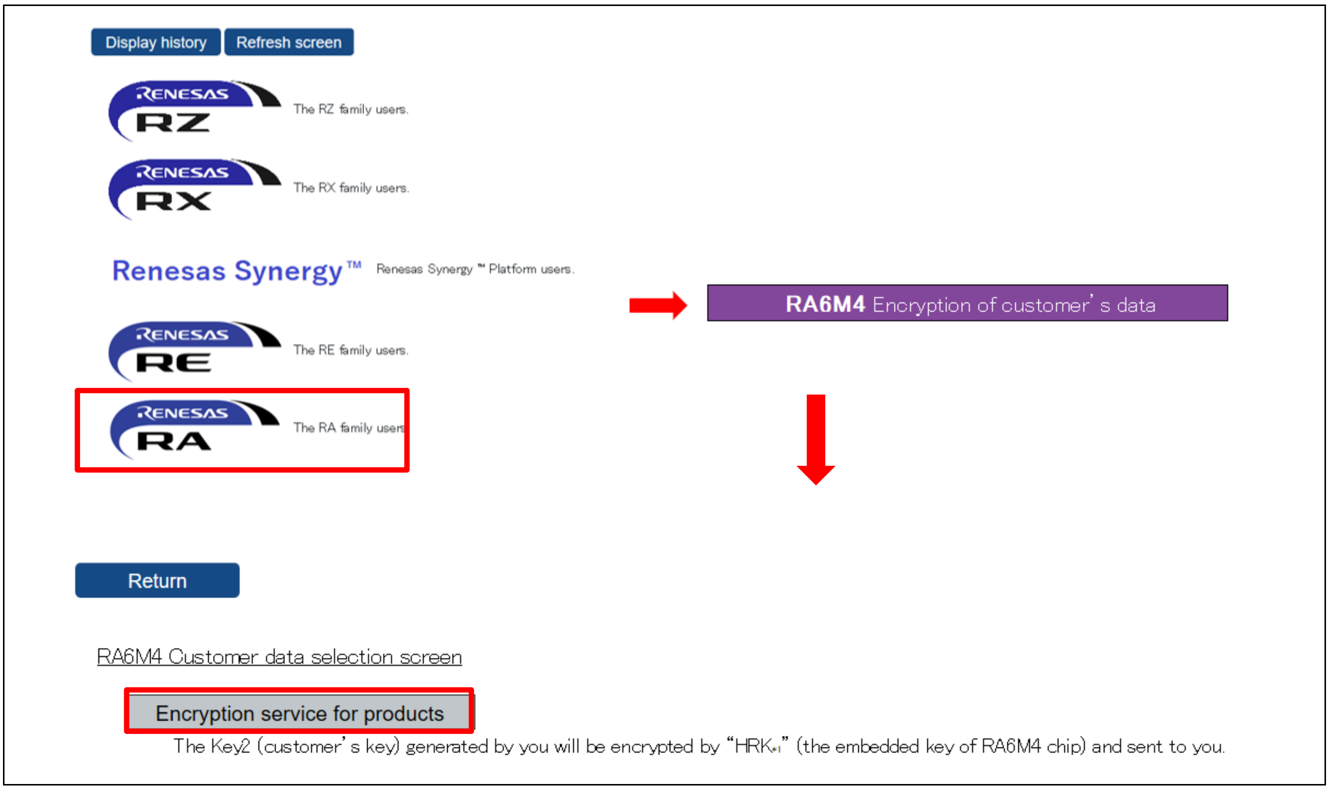
ufpk.key → ufpk.key.gpg: **Signing and encryption succeeded.**

Figure 41. Encrypt the UFPK Key with Renesas Public Key

3.3.3 Send UFPK Key to Renesas DLM Server

Now we can send the UFPK encrypted with Renesas Public Key to the Renesas DLM server where it will be decrypted by the Renesas Private Key and Generate the Wrapped UFPK (W-UFPK) by Renesas DLM sever.

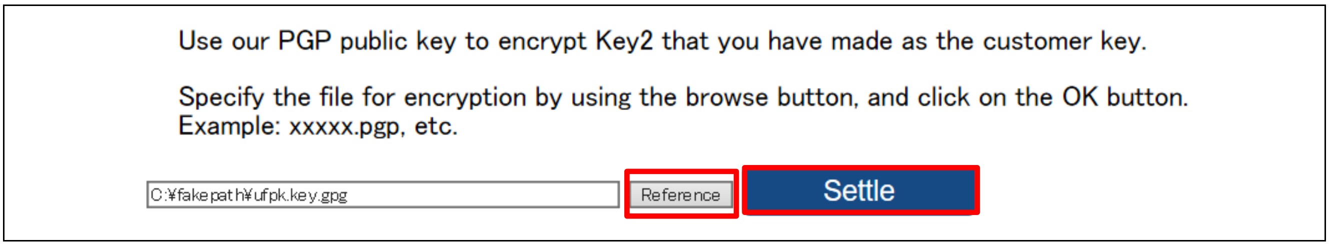
From the DLM sever user interface, select the RA Family series and choose **RA6M4 Encryption of customer** > **Encryption service for products** as shown below.



The Key2 (customer' s key) generated by you will be encrypted by "HRK*1" (the embedded key of RA6M4 chip) and sent to you.

Figure 42. Select RA Device

Next, click **Reference** and select the .pgp file generated from section 3.3.2.



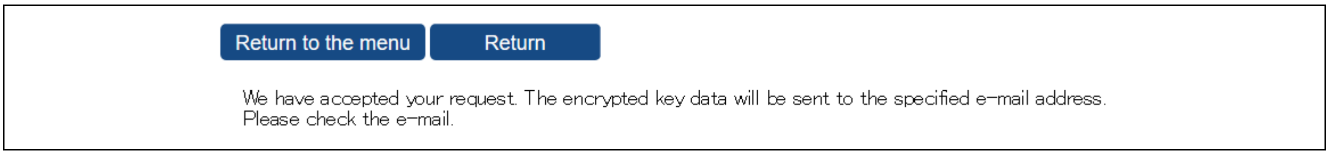
Use our PGP public key to encrypt Key2 that you have made as the customer key.

Specify the file for encryption by using the browse button, and click on the OK button.
Example: xxxxx.pgp, etc.

C:¥fakepath¥ufpk.key.gpg

Figure 43. Send Encrypted UFPK to DLM Server

Click **Settle** to see the following message.

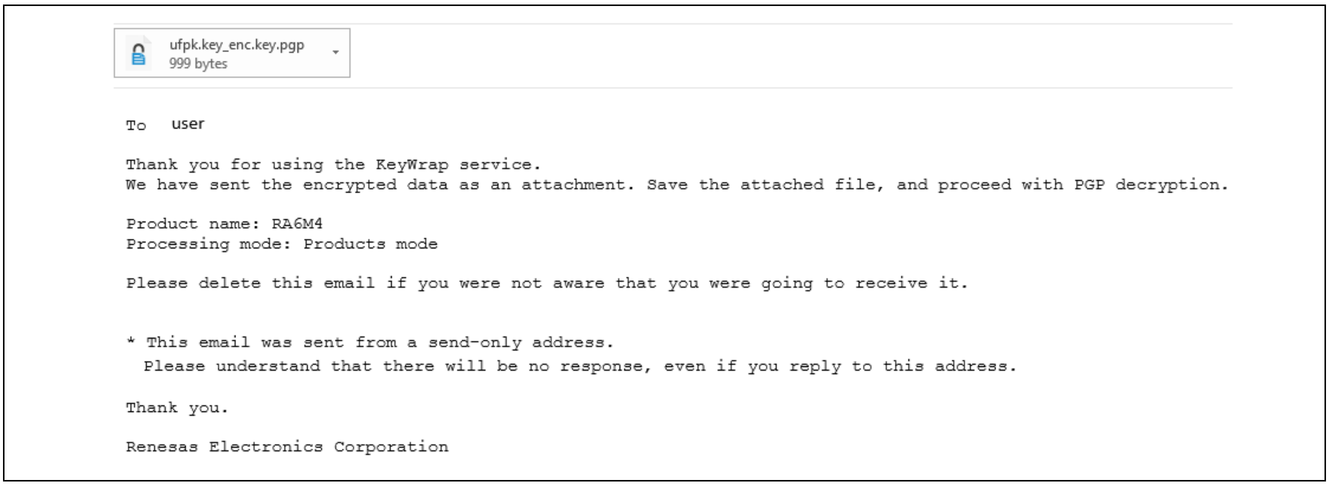


We have accepted your request. The encrypted key data will be sent to the specified e-mail address.
Please check the e-mail.

Figure 44. Message from DLM Sever

3.3.4 Receive the Wrapped UFPK Key Encrypted with Customer PGP Public Key

The Wrapped UFPK Key encrypted with customer PGP Public key should arrive in your email in couple of minutes in most cases.


ufpk.key_enc.key.pgp
999 bytes

```
To  user

Thank you for using the KeyWrap service.
We have sent the encrypted data as an attachment. Save the attached file, and proceed with PGP decryption.

Product name: RA6M4
Processing mode: Products mode

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.

Thank you.

Renesas Electronics Corporation
```

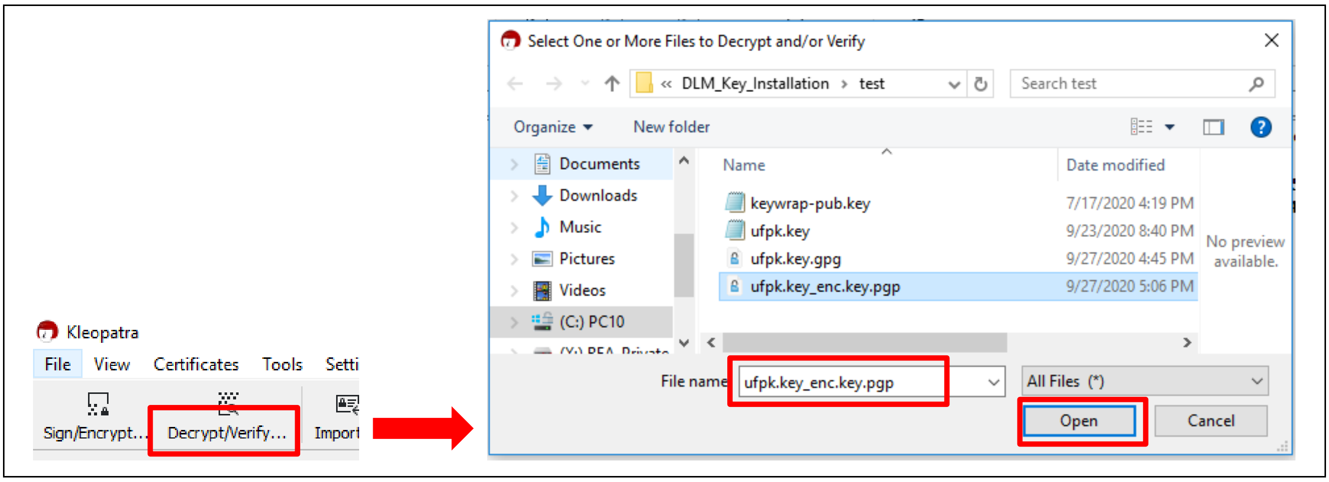
Figure 45. Receive the Wrapped DLM Key Encrypted by Renesas PGP Private Key

Download the Wrapped UFPK Key (which is encrypted by the Renesas private key) for use in the next step.

3.3.5 Decrypted the Encrypted Wrapped UFPK Key using Customer PGP Private Key

The Wrapped UFPK received from the email is encrypted with the customer public key. The customer needs to decrypt this key using the customer private key to acquire the Wrapped UFPK key, which can then be installed on the MCU using the RFP program.

With the Kleopatra program, click **Decrypt/Verify** and select the Wrapped UFPK received from section 3.3.4.

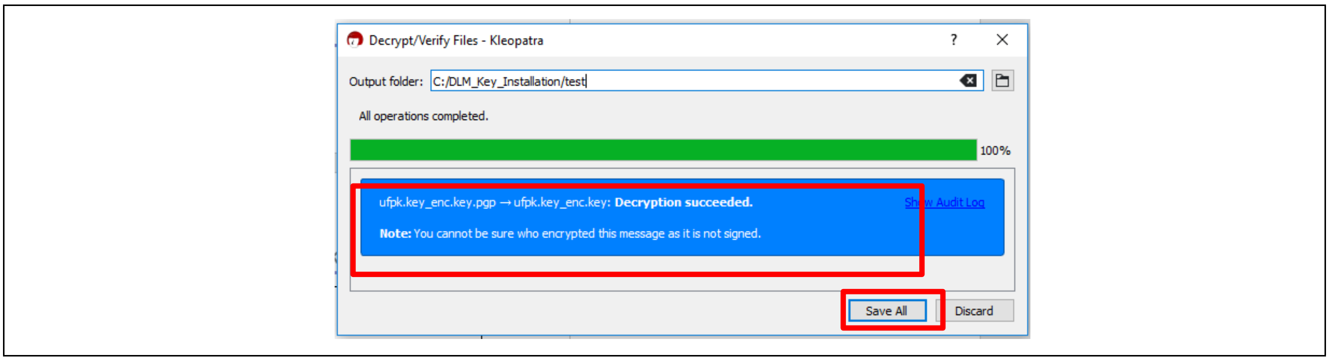

Kleopatra
File View Certificates Tools Setti
Sign/Encrypt... Decrypt/Verify... Import
Select One or More Files to Decrypt and/or Verify
« DLM_Key_Installation › test
Search test
Organize ▾ New folder
Documents
Downloads
Music
Pictures
Videos
(C:) PC10
No preview available.
Open
Cancel

Name	Date modified
keywrap-pub.key	7/17/2020 4:19 PM
ufpk.key	9/23/2020 8:40 PM
ufpk.key.gpg	9/27/2020 4:45 PM
ufpk.key_enc.key.pgp	9/27/2020 5:06 PM

File name: ufpk.key_enc.key.pgp
All Files (*)

Figure 46. Decrypt with Customer PGP Private Key

Next follow the prompt to provide the customer PGP key passphrase.



Output folder: C:/DLM_Key_Installation/test

All operations completed.

ufpk.key_enc.key.pgp → ufpk.key_enc.key: **Decryption succeeded.**

Note: You cannot be sure who encrypted this message as it is not signed.

Figure 47. Decrypt with Customer PGP Private Key

3.4 Generate the Encrypted DLM Key using UFPK and WUFPK

As shown in Figure 16, in order to create the Wrapped DLM Keys, the DLM Key needs to be encrypted by the UFPK and both the encrypted DLM Key and W-UFPK need to be sent to the MCU via the serial programming interface in order to create the Wrapped DLM Keys.

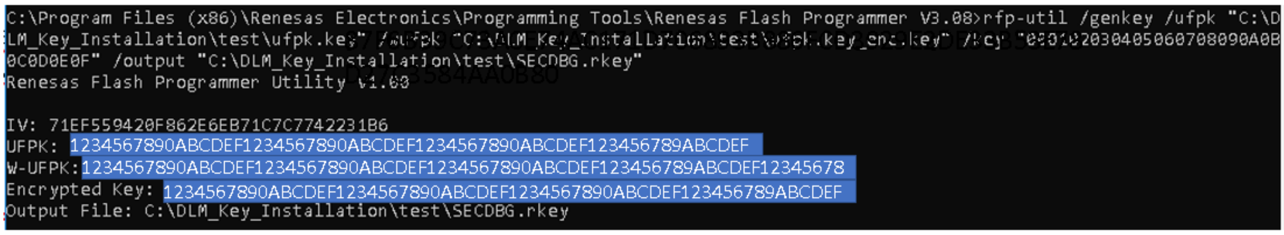
RFP command line tool `rfp-util.exe` can be used to generate the key bundle information to install to the MCU using RFP. The user should use `rfp-util.exe` with UFPK and W-UFPK as input to generate the key bundle to install with RFP. The following is an example command line input for generating the SECDBG_KEY to install on RA6M4 MCU.

Example Command Line Input:

```
rfp-util /genkey /ufpk “C:\DLM_Key_Installation\test\ufpk.key” /wufpk
“C:\DLM_Key_Installation\test\ufpk.key_enc.key” /key
“000102030405060708090A0B0C0D0E0F” /output
“C:\DLM_Key_Installation\test\SECDBG.rkey”
```

Note that **`000102030405060708090A0B0C0D0E0F`** is the plaintext DLM key data used to regress the MCU device lifecycle from NSECSD to SSD if `SECDBG.rkey` is installed on the MCU in SSD state. The customer can choose their own authentication data to use for their application. It is important to keep this information secure and not allow it to leak.

The following is an example of the generation of SECDBG key file to install to the MCU using RFP.



```
C:\Program Files (x86)\Renesas Electronics\Programming Tools\Renesas Flash Programmer V3.08>rfp-util /genkey /ufpk "C:\D
LM_Key_Installation\test\ufpk.key" /wufpk "C:\DLM_Key_Installation\test\ufpk.key_enc.key" /key "000102030405060708090A0B
0C0D0E0F" /output "C:\DLM_Key_Installation\test\SECDBG.rkey"
Renesas Flash Programmer Utility V1.00

IV: 71EF559420F862E6EB71C7C7742231B6
UFPK: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF
W-UFPK:1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF12345678
Encrypted Key: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF
Output File: C:\DLM_Key_Installation\test\SECDBG.rkey
```

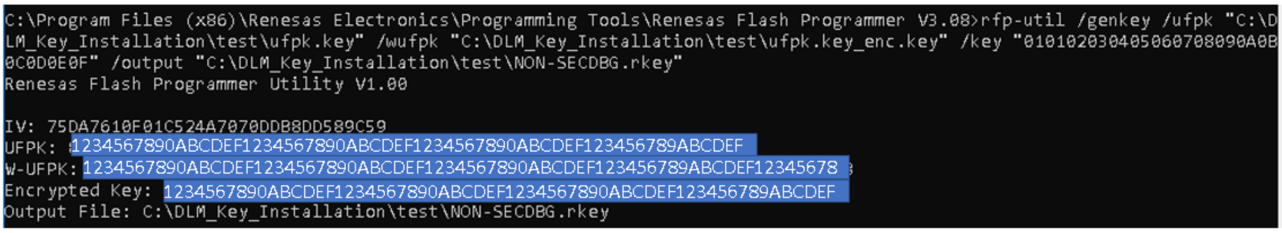
Figure 48. Use rfp-util.exe to Generate DLM Key SECDNG_KEY

Similarly, the user can generate the NONSECDBG_KEY using the following command line input:

```
rfp-util /genkey /ufpk “C:\DLM_Key_Installation\test\ufpk.key” /wufpk
“C:\DLM_Key_Installation\test\ufpk.key_enc.key” /key
“010102030405060708090A0B0C0D0E0F” /output “C:\DLM_Key_Installation\test\NON-
SECDBG.rkey”
```

Note that **`010102030405060708090A0B0C0D0E0F`** is the plaintext DLM key data that will be used to regress the MCU device lifecycle from DPL to NSECSD if `NONSECDBG.rkey` is installed on the MCU in NSECSD state. The customer can choose their own authentication data to use for their application. It is important to keep this information secure and not allow it to leak.

The following is an example of the generation of NONSECDBG key file to install to the MCU using RFP.



```
C:\Program Files (x86)\Renesas Electronics\Programming Tools\Renesas Flash Programmer V3.08>rfp-util /genkey /ufpk "C:\D
LM_Key_Installation\test\ufpk.key" /wufpk "C:\DLM_Key_Installation\test\ufpk.key_enc.key" /key "010102030405060708090A0B
0C0D0E0F" /output "C:\DLM_Key_Installation\test\NON-SECDBG.rkey"
Renesas Flash Programmer Utility V1.00

IV: 75DA7610F01C524A7070DDB8DD589C59
UFPK: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF
W-UFPK:1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF12345678
Encrypted Key: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF
Output File: C:\DLM_Key_Installation\test\NON-SECDBG.rkey
```

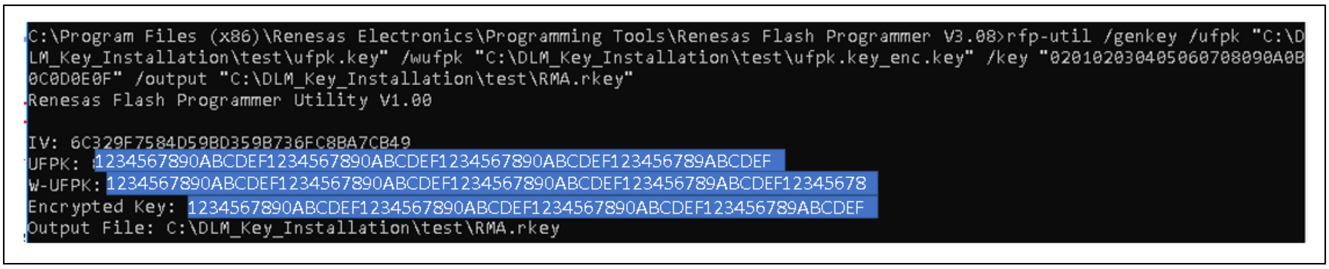
Figure 49. Use rfp-util.exe to Generate DLM Key NONSECDBG_KEY

Similarly, the user can generate the RMA_KEY using the following command line input:

```
rfp-util /genkey /ufpk “C:\DLM_Key_Installation\test\ufpk.key” /wufpk
“C:\DLM_Key_Installation\test\ufpk.key_enc.key” /key
“020102030405060708090A0B0C0D0E0F” /output
“C:\DLM_Key_Installation\test\RMA.rkey”
```

Note that **`0201020304050607080900A0B0C0D0E0F`** is the plaintext DLM key data that will be used to advance the MCU device lifecycle from DPL to RMA_REQ if `RMA.rkey` is installed on the MCU in SSD state. The customer can choose their own authentication data to use for their application. It is important to keep this information secure and not allow it to leak.

The following is an example of the generation of RMA key file to install to the MCU using RFP.



```
C:\Program Files (x86)\Renesas Electronics\Programming Tools\Renesas Flash Programmer V3.08>rfp-util /genkey /ufpk "C:\D
LM_Key_Installation\test\ufpk.key" /wufpk "C:\DLM_Key_Installation\test\ufpk.key_enc.key" /key "0201020304050607080900A0B
0C0D0E0F" /output "C:\DLM_Key_Installation\test\RMA.rkey"
Renesas Flash Programmer Utility V1.00

IV: 6C329F7584D59BD359B736FC8BA7CB49
UFPK: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF
W-UFPK: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF12345678
Encrypted Key: 1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF123456789ABCDEF
Output File: C:\DLM_Key_Installation\test\RMA.rkey
```

Figure 50. Use rfp-util.exe to Generate DLM Key RMA_KEY

The user can now proceed to use RFP to install the generated DLM Keys (`SECDBG.rkey`, `RMA.rkey`, and `NONSECDBG.rkey`).

3.5 DLM Key Installation

This section provides required configuration settings to perform DLM Key installation. For instructions on how to create an RFP project and establish connections to the target board, see the RFP User's Manual. This section provides key configuration settings for each of the state transitions.

For operations on unauthenticated transitions and **All erase** operations, see section 1.2.5.

Arm® TrustZone® technology enabled RA MCUs include a 256-bit Hardware Unique Key (HUK, see section 5.1), which is installed during the MCU's CM state. The HUK is unique per MCU and ensures the secure storage of the DLM keys. The DLM Keys will be installed wrapped by the HUK. This installation procedure ensures the DLM Keys are only usable on the MCU installed.

3.5.1 Install Secure Debug Key

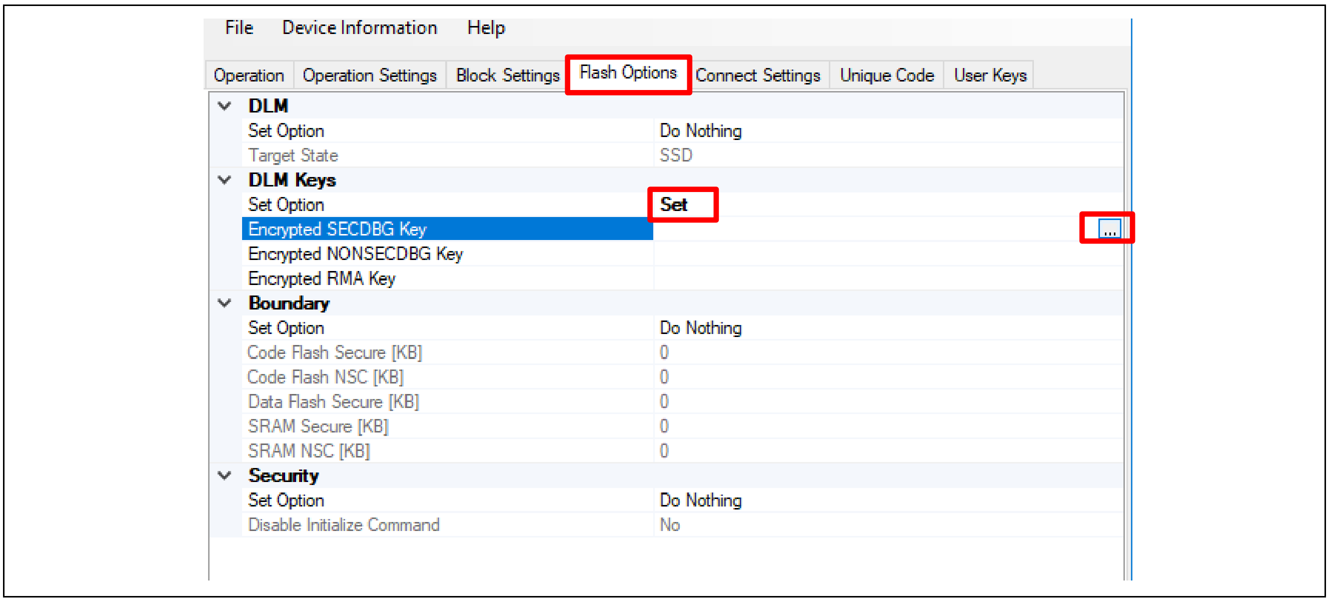
The customer can optionally install the Secure Debug Key (SECDBG_KEY) and Return Material Authorization (RAM_KEY) in SSD state. Ensure that the following two conditions are met prior to the key installation:

- The MCU is in SSD state
- SECDBG_KEY and RMA_KEY is generated following the steps in section 3.4.

To install the Secure Debug Key and the Return Material Authorization Key, first upload the DLM Keys to the RFP program. This section uses Secure Debug Key generated in the previous section as an example. The same procedure applies to the installation of Return Material Authorization Key installation.

Unzip `ra6m4_dlm_key_install.rfp.zip` to reveal `ra6m4_dlm_key_install.rfp.rpj`. Launch RFP and open project `ra6m4_dlm_key_install.rfp.rpj` and review the settings explained in this section.

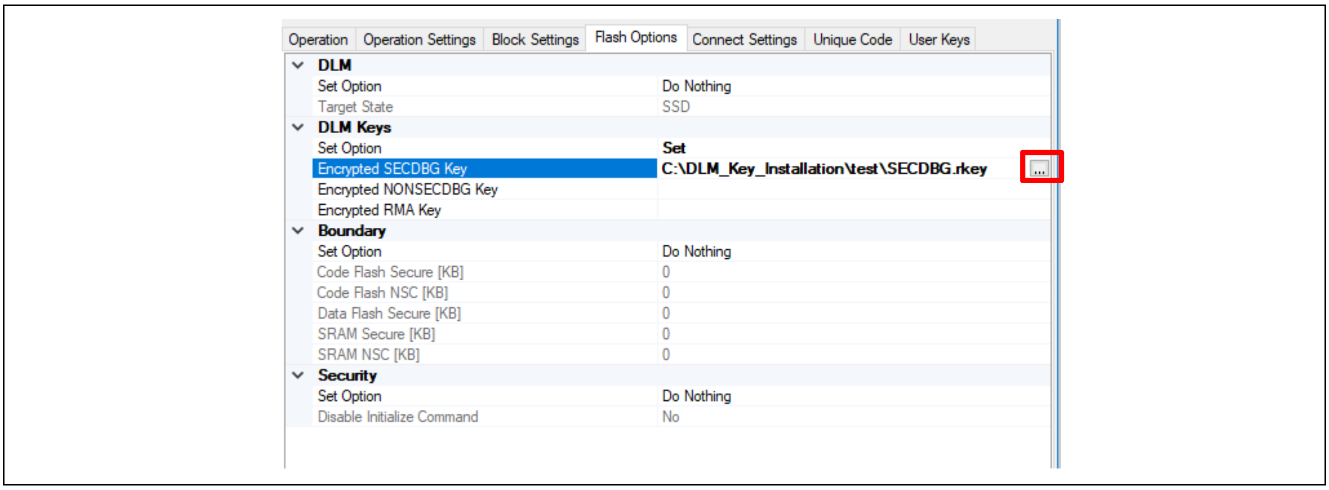
For the **Flash Options**, use the **Set Option** to set up **Set**. Now, highlight **Encrypted SECDBG Key** and then click the “**…**” on the right to select the corresponding SECDBG_KEY.



DLM	
Set Option	Do Nothing
Target State	SSD
DLM Keys	
Set Option	**Set**
Encrypted SECDBG Key	
Encrypted NONSECDBG Key	
Encrypted RMA Key	
Boundary	
Set Option	Do Nothing
Code Flash Secure [KB]	0
Code Flash NSC [KB]	0
Data Flash Secure [KB]	0
SRAM Secure [KB]	0
SRAM NSC [KB]	0
Security	
Set Option	Do Nothing
Disable Initialize Command	No

Figure 51. Select the SECDBG_Key to install

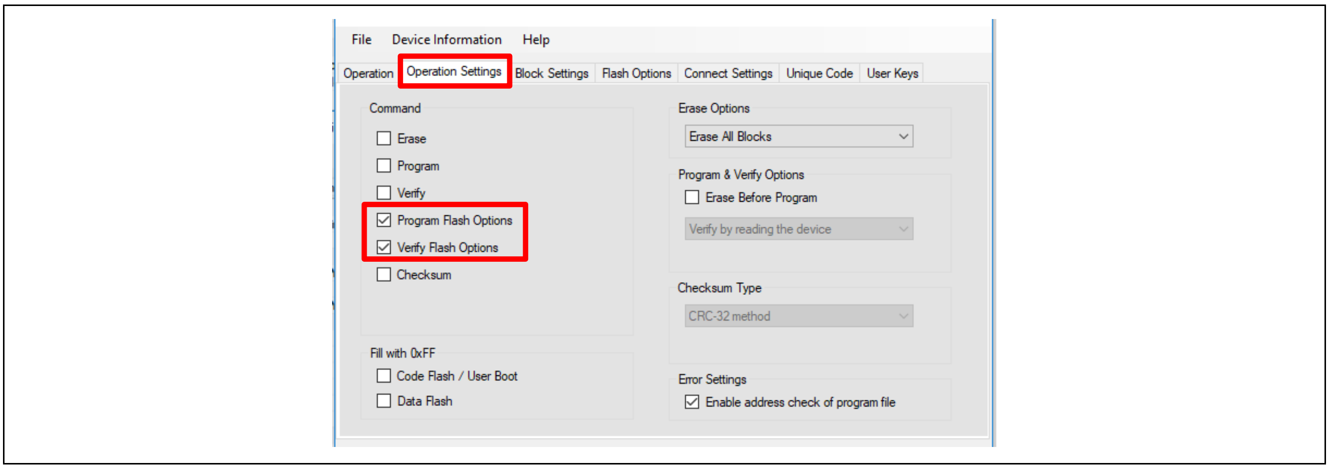
Select the `SECDBG.rkey` generated in section 3.4.



DLM	
Set Option	Do Nothing
Target State	SSD
DLM Keys	
Set Option	**Set**
Encrypted SECDBG Key	**C:\DLM_Key_Installation\test\SECDBG.rkey**
Encrypted NONSECDBG Key	
Encrypted RMA Key	
Boundary	
Set Option	Do Nothing
Code Flash Secure [KB]	0
Code Flash NSC [KB]	0
Data Flash Secure [KB]	0
SRAM Secure [KB]	0
SRAM NSC [KB]	0
Security	
Set Option	Do Nothing
Disable Initialize Command	No

Figure 52. Select the SECDBG_KEY to Install

Under the **Operation Settings** tab, select **Program Flash Options** and **Verify Flash Options**.



Command
- □ Erase
- □ Program
- □ Verify
- ☑ Program Flash Options
- ☑ Verify Flash Options
- □ Checksum

Fill with 0xFF
- □ Code Flash / User Boot
- □ Data Flash

Erase Options: Erase All Blocks

Program & Verify Options
- □ Erase Before Program
- Verify by reading the device

Checksum Type: CRC-32 method

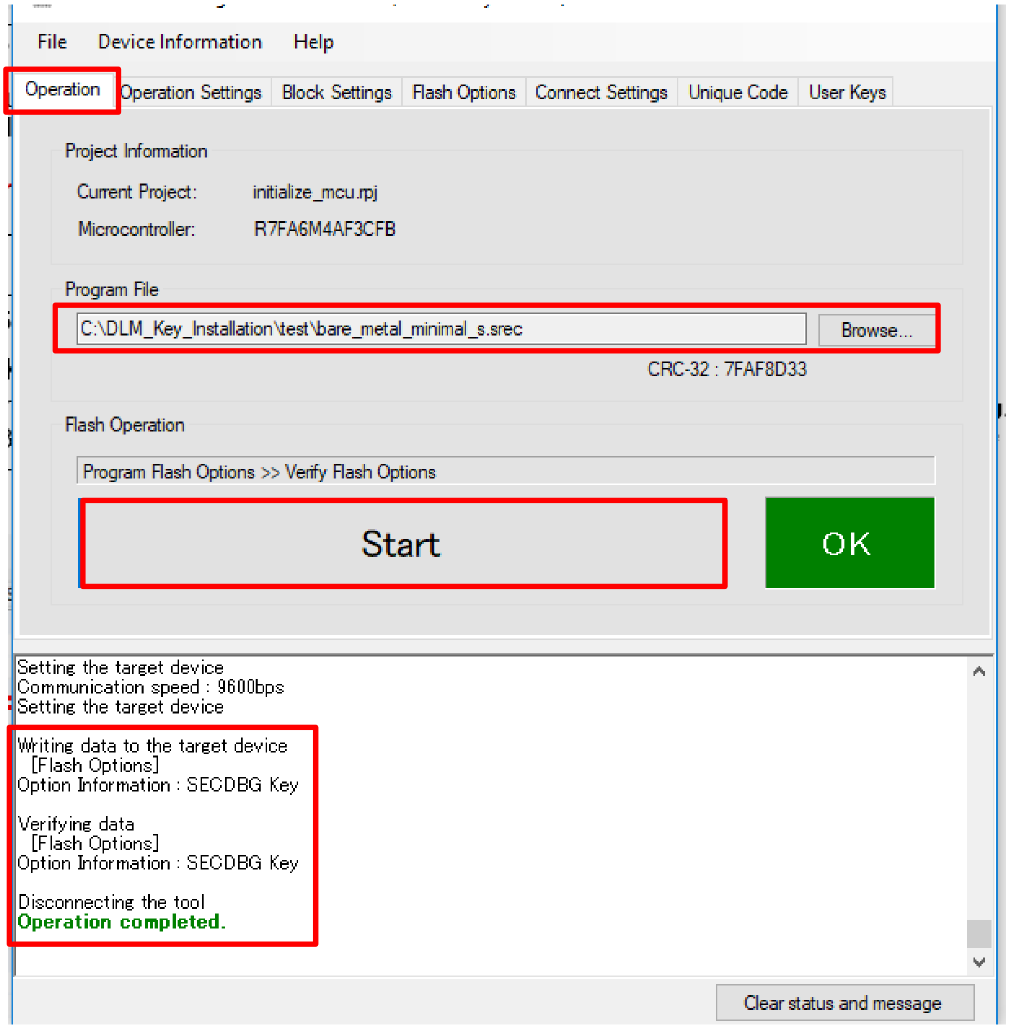
Error Settings
- ☑ Enable address check of program file

Figure 53. Select Program and Verify Flash Option Setting

RFP operation for setting the DLM keys can be done at the same time as programming a valid binary file selected in the **Operation** tab. Downloading with a binary is optional.

Navigate to the **Operation** tab, select the secure application binary intended and click **Start** to program the settings.

Unzip `test.zip` to reveal `bare_metal_minimal_s.srec` and `bare_metal_minimal_ns.srec`. For testing purpose, these binaries are included with this application note for the user's convenience.



Current Project: initialize_mcu.rpj

Microcontroller: R7FA6M4AF3CFB

Program File: C:\DLM_Key_Installation\test\bare_metal_minimal_s.srec

CRC-32 : 7FAF8D33

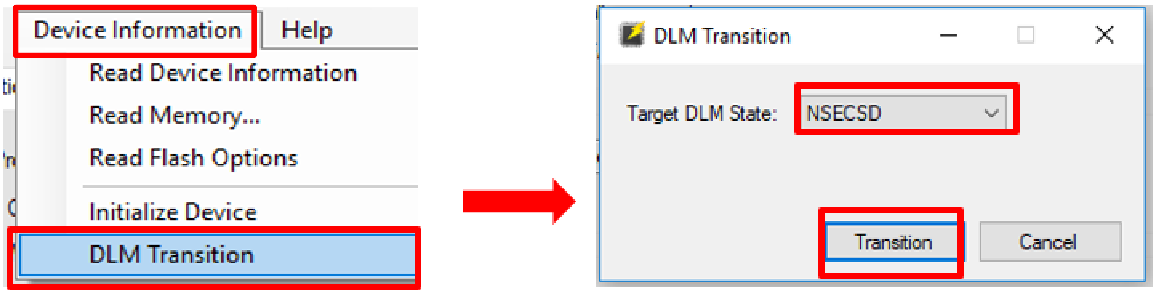
Flash Operation: Program Flash Options >> Verify Flash Options

```
Setting the target device
Communication speed : 9600bps
Setting the target device

Writing data to the target device
  [Flash Options]
Option Information : SECDBG Key

Verifying data
  [Flash Options]
Option Information : SECDBG Key

Disconnecting the tool
Operation completed.
```

Figure 54. Install the SECDBG_KEY

3.5.2 Install the Non-Secure Debug Key

Prior to installing the Non-Secure Debug Key, transition the MCU Device Lifecycle State to NSECSD.

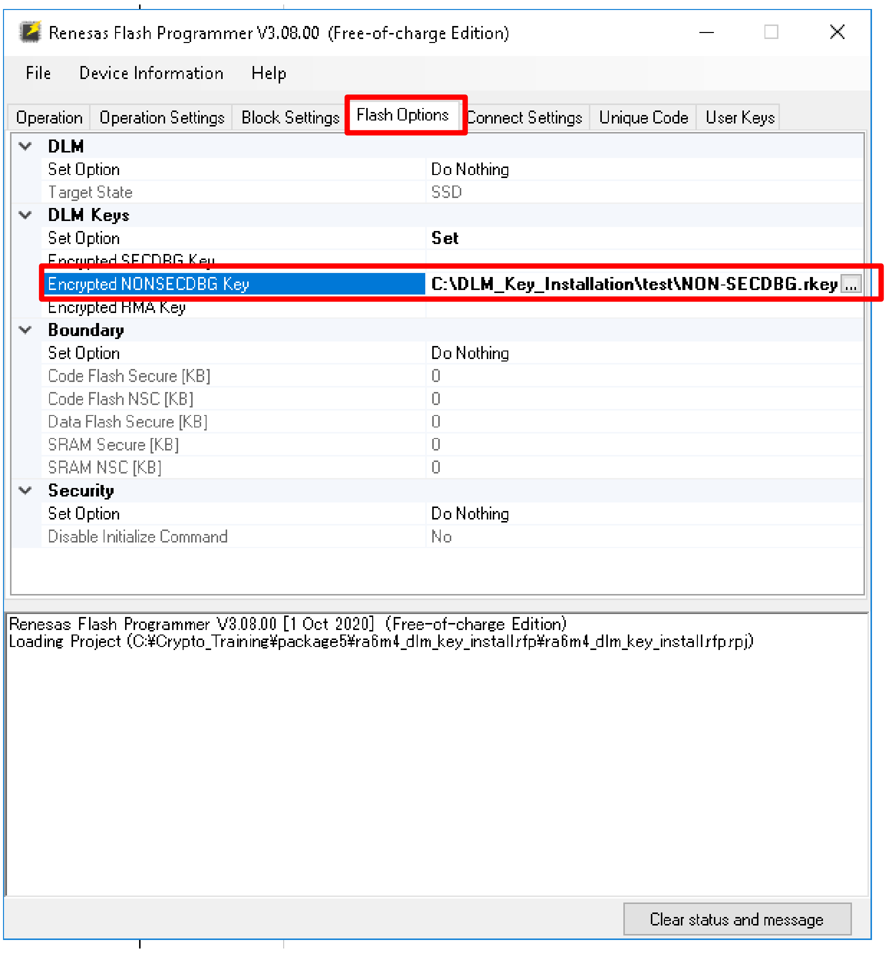


Target DLM State: NSECSD

Figure 55. Transition the MCU Device Lifecycle State to NSECSD

Next, follow steps similar to section 3.5.1 to install the Non-Secure Debug Key. In this example, we can use the NONSECDBG_KEY (`NONSECDBG.rkey`) generated with Figure 49 to illustrate the operation.

Note that the user needs to delete the SECDBG key file entry and add the NONSECDBG key file entry as shown below.



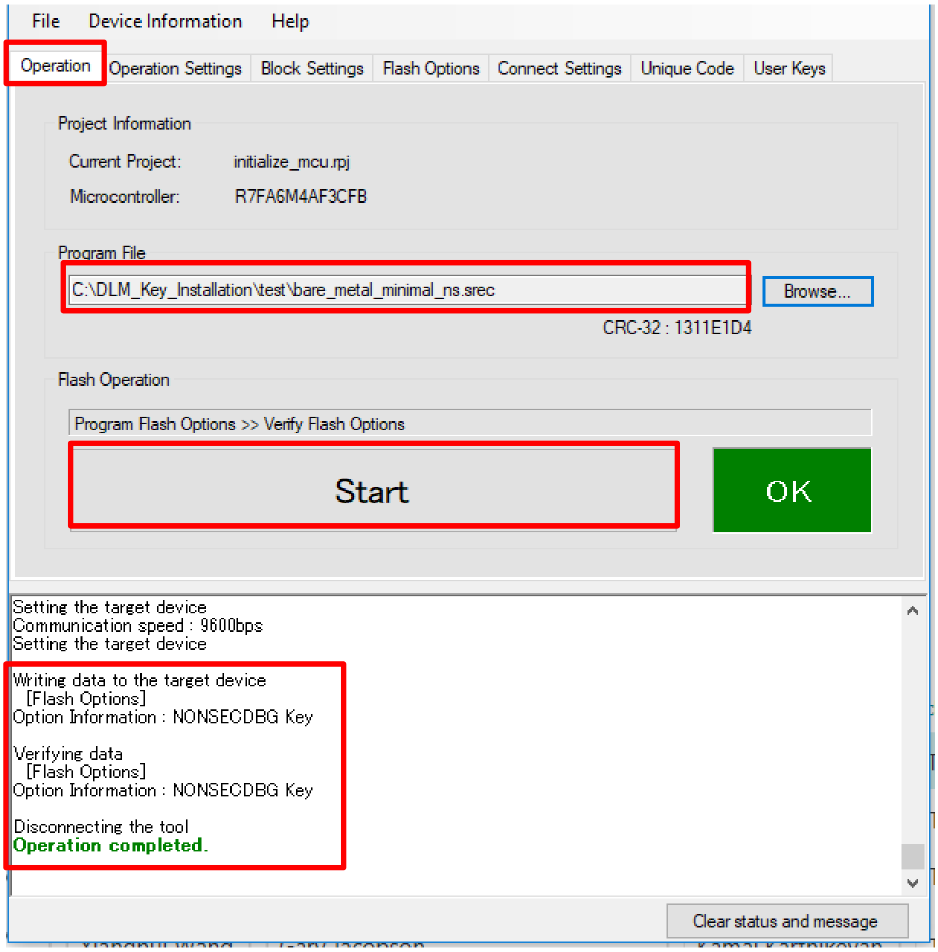
DLM	
Set Option	Do Nothing
Target State	SSD
DLM Keys	
Set Option	**Set**
Encrypted SECDBG Key	
Encrypted NONSECDBG Key	**C:\DLM_Key_Installation\test\NON-SECDBG.rkey**
Encrypted HMA Key	
Boundary	
Set Option	Do Nothing
Code Flash Secure [KB]	0
Code Flash NSC [KB]	0
Data Flash Secure [KB]	0
SRAM Secure [KB]	0
SRAM NSC [KB]	0
Security	
Set Option	Do Nothing
Disable Initialize Command	No

```
Renesas Flash Programmer V3.08.00 [1 Oct 2020] (Free-of-charge Edition)
Loading Project (C:¥Crypto_Training¥package5¥ra6m4_dlm_key_install.rfp¥ra6m4_dlm_key_install.rfp.rpj)
```

Figure 56. Select the NONSECDBG.rkey to Install

Similar to the installation of SECDBG_KEY, RFP operation on setting the DLM keys can be done at the same time as programming a valid non-secure binary file selected in the **Operation** tab. Installing the binary with DLM key installation is optional.

As explained previously, for testing purpose, the `bare_metal_minimal_ns.srec` binary is included with this application note for user's convenience.



Current Project: initialize_mcu.rpj

Microcontroller: R7FA6M4AF3CFB

C:\DLM_Key_Installation\test\bare_metal_minimal_ns.srec

CRC-32 : 1311E1D4

Program Flash Options >> Verify Flash Options

```
Setting the target device
Communication speed : 9600bps
Setting the target device

Writing data to the target device
  [Flash Options]
Option Information : NONSECDBG Key

Verifying data
  [Flash Options]
Option Information : NONSECDBG Key

Disconnecting the tool
Operation completed.
```

Figure 57. Install the NONSECDBG_KEY Using RFP

3.6 Authenticated DLM State Transitions

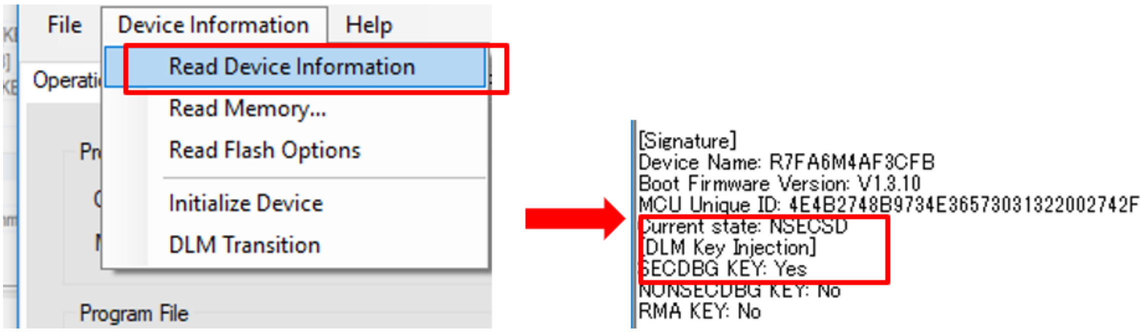
This section provides operational steps for authenticated DLM state transitions. The assumption is that SECDBG_KEY and NONSECDBG_KEY are already installed using the steps discussed previously.

Note that for practice purposes, the user can install RMA_KEY in SSD state. However, unless this is a product return, DO NOT transition to RMA_REQ state using the RMA_KEY. Once transitioned to RMA_REQ state, **All erase** operation does not work anymore. The MCU will be locked out of debugging and reprogram capability via the serial programming port.

In addition, transition to RAM_REQ should only happen in a secure environment.

3.6.1 Authenticated Transition from Non-Secure Debug State to Secure Debug State

Assume that the current Device Lifecycle State is NSECSD and that SECDBG_KEY has been previous installed. User can read out the **Device Information** to confirm this.

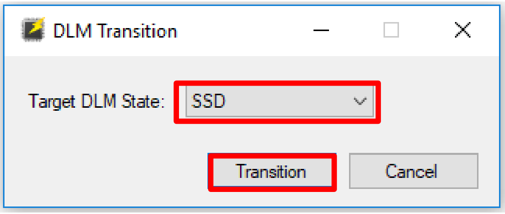


```
[Signature]
Device Name: R7FA6M4AF3CFB
Boot Firmware Version: V1.3.10
MCU Unique ID: 4E4B2748B9734E36573031322002742F
Current state: NSECSD
[DLM Key Injection]
SECDBG KEY: Yes
NONSECDBG KEY: No
RMA KEY: No
```

Figure 58. Confirm the MCU is in NSECSD and the SECDBG_KEY is Installed

Use the following steps to regress the Device Lifecycle State to SSD:

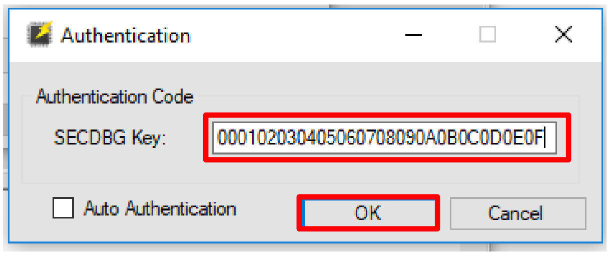
1. Select transition to **SSD** from the Device Information menu in RFP.



Target DLM State: SSD

Figure 59. Select Regress MCU Device Lifecycle State back to SSD

2. If the device is in NSECSD state and the SECDBG Key has been installed on the MCU, the following prompt will pop up. Follow the prompt to provide the SECDBG Key authentication data and then click **OK.** This would be **000102030405060708090A0B0C0D0E0F** if the `SECDBG.rkey` generated based on Figure 48, is installed in the MCU.

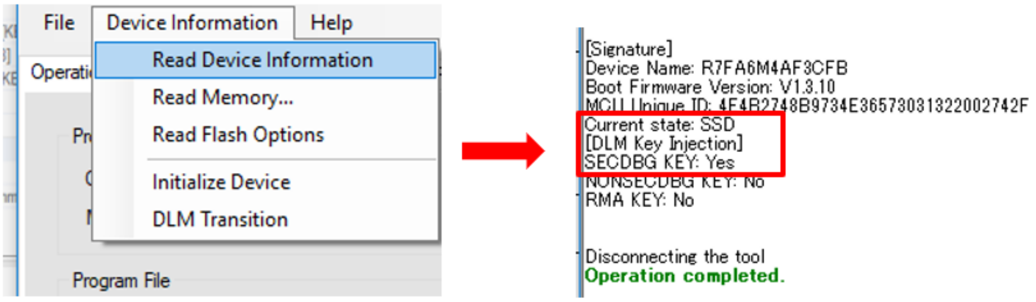


SECDBG Key: 000102030405060708090A0B0C0D0E0F

☐ Auto Authentication

Figure 60. Provide the Authentication Key for SECDBG

3. The user can now confirm that the Device Lifecycle State has transitioned back to SSD state.

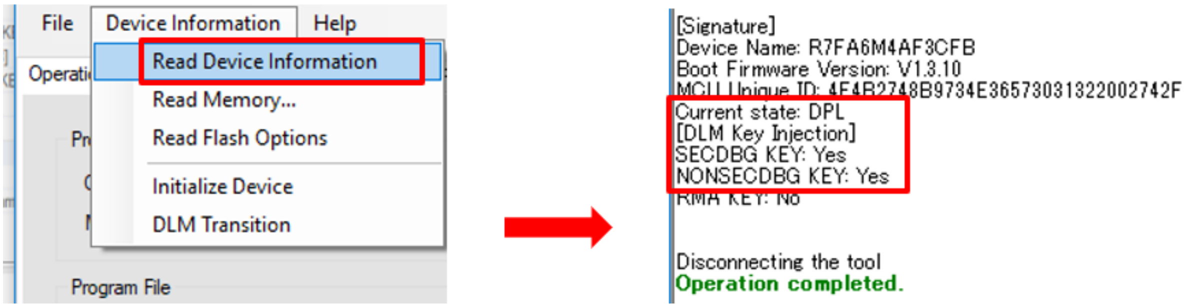


```
[Signature]
Device Name: R7FA6M4AF3CFB
Boot Firmware Version: V1.3.10
MCU Unique ID: 4F4B2748B9734E36573031322002742F
Current state: SSD
[DLM Key Injection]
SECDBG KEY: Yes
NONSECDBG KEY: No
RMA KEY: No

Disconnecting the tool
Operation completed.
```

Figure 61. Confirm Device Lifecycle Transitions Back to SSD

3.6.2 Authenticated Transition from Deployed State to Non-Secure Debug State

As explained in section 2.5, if the deployment state is DPL, it is possible to regress the MCU device lifecycle state from DPL to NSECSD. Again, the user can confirm the MCU Device Lifecycle State by reading out the **Device Information** as shown in the following graphic.

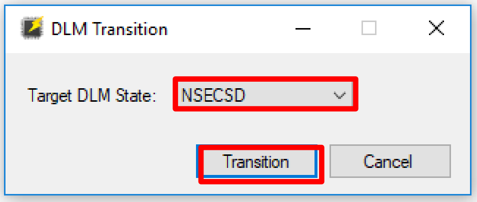


```
[Signature]
Device Name: R7FA6M4AF3CFB
Boot Firmware Version: V1.3.10
MCU Unique ID: 4F4B2748B9734E36573031322002742F
Current state: DPL
[DLM Key Injection]
SECDBG KEY: Yes
NONSECDBG KEY: Yes
RMA KEY: No

Disconnecting the tool
Operation completed.
```

Figure 62. Confirm the Device Lifecycle State and DLM Key status

Use the following steps to regress the Device Lifecycle State to NSECSD.

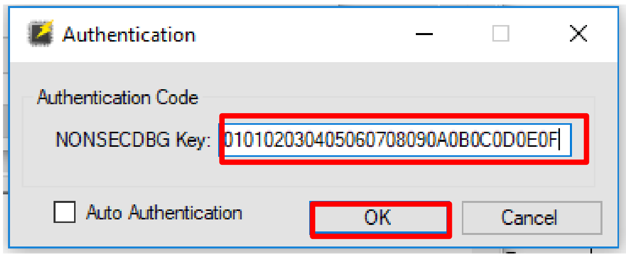
1. Select transition to NSECSD.



Target DLM State: NSECSD

Figure 63. Select Transition to NSECSD

2. If the device is in DPL state and the NONSECDBG_KEY has been installed on the MCU, the following prompt will pop up. Follow the prompt to provide the NONSECDBG Key and then click **OK**. This would be **010102030405060708090A0B0C0D0E0F** if the `NONSECDBG.rkey` generated based on Figure 49 is installed in the MCU.

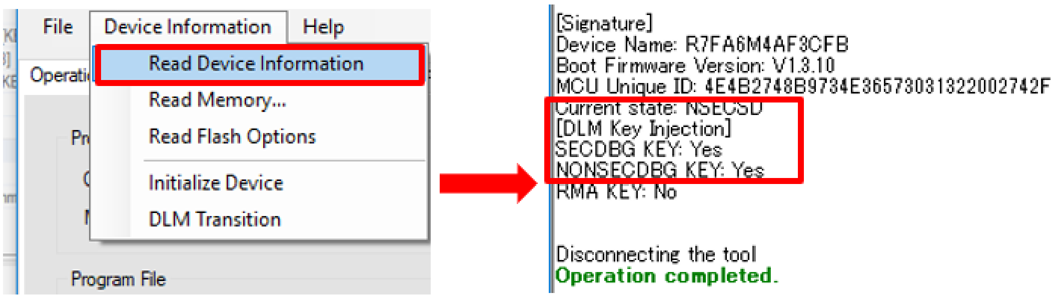


NONSECDBG Key: 010102030405060708090A0B0C0D0E0F

☐ Auto Authentication

Figure 64. Provide the authenticate key for NONSECDBG_KEY

3. The user can now confirm that the Device Lifecyle State is regressed back to NSECSD state.



```
[Signature]
Device Name: R7FA6M4AF3CFB
Boot Firmware Version: V1.3.10
MCU Unique ID: 4E4B2748B9734E36573031322002742F
Current state: NSECSD
[DLM Key Injection]
SECDBG KEY: Yes
NONSECDBG KEY: Yes
RMA KEY: No


Disconnecting the tool
Operation completed.
```

Figure 65. Confirm Device Lifecycle has Transitioned Back to NSECSD

4. References

Available on renesas.com:

- Renesas RA6M4 Group User’s Manual: Hardware
- Flexible Software Package (FSP) User’s Manual
- Securing Data at Rest Utilizing Renesas Security MPU

5. Appendix

5.1 Glossary

Term	Meaning
SCE9	Secure Crypto Engine 9 is a hardware unit which resides on Renesas Arm® Cortex®-M33 MCU
Device Certificate	Certificate uniquely identifying an individual device. It is digitally signed, asserting that the certificate comes from a known source and has not been modified, and that the device is trusted.

Term	Meaning
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)
SCE	Secure Crypto Engine – A module in the MCU that provides for efficient, low-power cryptographic acceleration, TRNG (True Random Number Generation), and creation and isolation of cryptographic keys.
PKI	Public Key Infrastructure – A set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates, which are typically used to manage secure identity via public key cryptography.
Key Pair	Asymmetric keys are generated in pairs – a public and private key. The private key is held in secret by only one party and can be used to assert that party’s identity. The public key is freely distributed and is uniquely associated with the private key.
Secure Code	A function or group of functions that resides in a secure region of internal flash, as defined and enforced by the MPUs. These secure functions can access both secure data and non-secure data regions.
Non-Secure code	A function or group of functions that resides in a non-secure region of internal flash. These non-secure codes cannot access the secure region. They can only access the non-secure region.
HUK	Hardware Unique Key. This is a unique key stored inside the RA Family MCU.
Challenge String	Randomly generated string at the host application. This string is used by the host application to validate the ownership of the private key by the target.
Unique ID	An identification value, unique to each individual RA Family MCU, that is stored inside the MCU. The unique ID is used by the SCE when it wraps a key.
Challenge Response String	The response to the challenge string. The Challenge Response String is the signature of the challenge data as created by signing the Challenge String with the receiver's private key.

Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

EK-RA6M4 Resources	renesas.com/ra/ek-ra6m4
RA Product Information	renesas.com/ra
RA Product Support Forum	renesas.com/ra/forum
RA Flexible Software Package	renesas.com/FSP
Renesas Support	renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.01.20	—	First release document
1.10	Dec.09.20	—	Updated importing Renesas Public Key to Kleopatra procedure
1.11	Apr.30.21	—	Improve wording.

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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 losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. 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 robots; 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; undersea 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.

(Rev.5.0-1 October 2020)

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.