Renesas RA Family

Device Lifecycle Management for RA8 MCUs

Introduction

Renesas RA8 Series Device Lifecycle Management (DLM) is unique to the RA MCU Family. Compared with the RA Cortex-M33 MCU Device Lifecycle Management System, in the RA8 MCU Series, Device Lifecycle Management has been separated from authenticated debug. This application project provides a comprehensive guideline on how to use the RA8 DLM system.

For the Cortex-M33 DLM systems, the debug capability and serial programming capability are defined by the device lifecycle states. Please reference application project R11AN0496 for the Cortex-M33 DLM usage.

Required Resources

The following resources are referenced throughout this application note:

Development Tools and Software

- Renesas Flash Programmer (RFP) v3.15 or later
  https://www.renesas.com/us/en/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html
- Renesas Security Key Management Tool v1.06 or later
  https://www.renesas.com/software-tool/security-key-management-tool
- Gpg4win
  http://www.gpg4win.org/

Hardware

- EK-RA8M1, Evaluation Kit for RA8M1 MCU Group (renesas.com/ra/ek-ra8m1)
- 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 Flash Programmer (RFP). In addition, the application note assumes that you have some knowledge of RA Family MCU security features. See chapter *Security Features* in the *Renesas RA8M1 Group MCU User's Manual: Hardware* for background knowledge. It is also recommended that the user read the Security Key Management Tool (SKMT) manual prior to proceeding with this application note.

The intended audience are 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 RA8 MCU Series

The Renesas RA8 DLM system is designed to provide IP protection during the entire product lifecycle of the MCU. This system covers the needs of Device Lifecycle Management during MCU releases, product development, product deployment, field updates, and potential end-of-product failure analysis.

1.1 RA8 MCU Device Lifecycle Management Overview

The key concepts for the Device Lifecycle Management on RA8 MCUs are the Device Lifecycle States, the Authentication Levels (AL2, AL1, AL0), and the Protection Levels (PL2, PL1, PL0).

The MCU's debug access level upon power-up is defined by the Protection Level, which is stored in non-volatile memory. The initial debug Authentication Level is determined by the Protection Level. A Protection Level change will change the initial Authentication Level upon the next power-up.

The debug Authentication Level (AL) is stored in volatile memory. Upon a device power-on reset, the Authentication Level is initialized to the Protection Level. An Authentication Level change will apply only when the device is powered. The Authentication Level determines the OEM state's allowed debug and serial programming access, as shown in Table 1.

Table 1 Authentication Levels

Authentication Level	Debug, Read/Erase/Program over serial programming
AL2	Secure and Non-Secure
AL1	Non-Secure Only
AL0	None

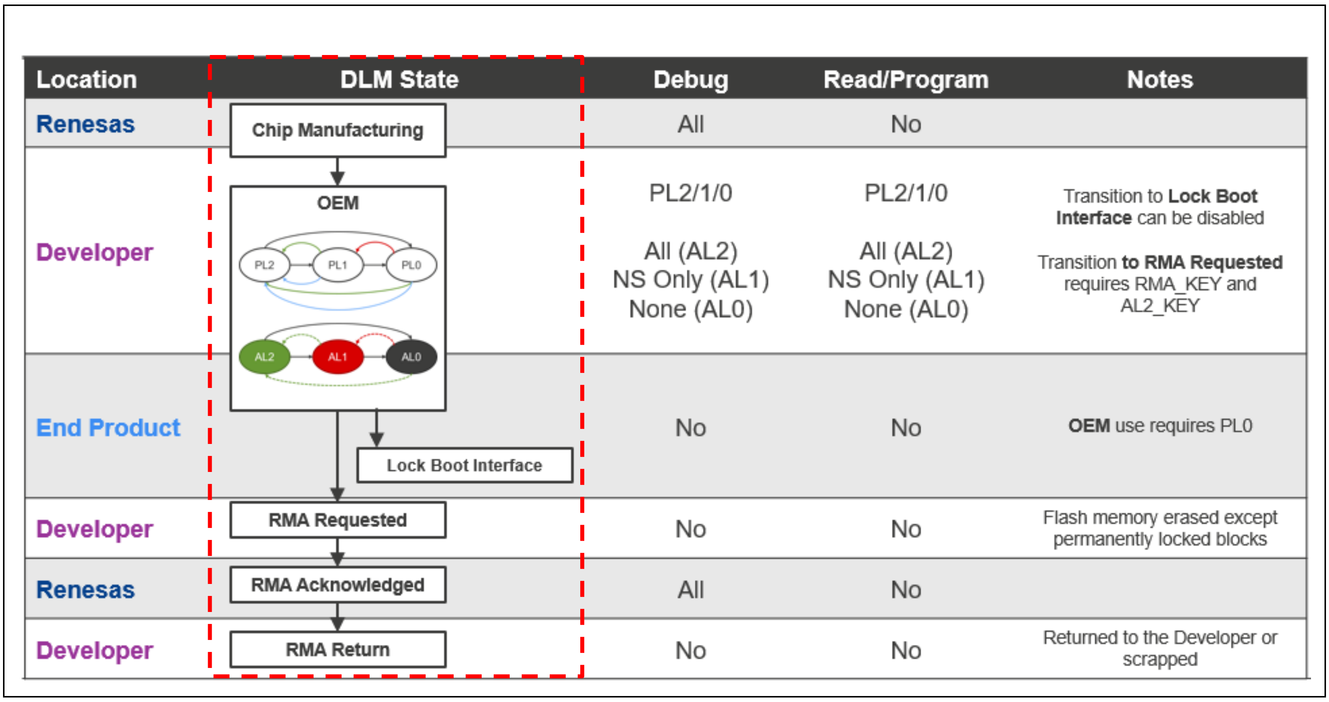
1.2 DLM States and Authenticated Debug

Table 2 is the summary of the available DLM states and the various capabilities in each state.

Table 2 Device Lifecycle state definition and capabilities in each state.

Lifecycle	Definition	Protection level	Debug function	Serial programming	Renesas test mode
CM	"**C**hip **M**anufacturing" The customer may receive the device in this state. If so, the customer needs to move the state to OEM prior to application development.	PL2	Available in the secure and nonsecure debug	Available Cannot access code/data flash area	Not available
OEM	"**O**riginal **E**quipment **M**anufacturer" The customer may receive the device in OEM state with PL2.	PL2 or PL1 or PL0	Depend on the authentication level (AL)		Not available
LCK_BOOT	"**L**o**CK**ed **BOOT** interface" The debug interface and the serial programming interfaces are permanently disabled.	PL0	Not available	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.	PL0	Not available	Available Cannot access code/data flash area	Not available
RMA_ACK	"**R**eturn **M**aterial **A**uthorization **ACK**nowledged" Failure analysis in Renesas	PL2	Available in the secure and non-secure debug	Available Cannot access code/data flash area	Available
RMA_RET	"Return Material Authorization RETurn" The device is back to the customer. The the device does not boot.	PL0	Not available	Not available	Not available

Figure 1 provides additional information on the available state transitions in OEM state.



Location	DLM State	Debug	Read/Program	Notes
Renesas	Chip Manufacturing	All	No	
Developer	OEM	PL2/1/0 All (AL2) NS Only (AL1) None (AL0)	PL2/1/0 All (AL2) NS Only (AL1) None (AL0)	Transition to **Lock Boot Interface** can be disabled Transition **to RMA Requested** requires RMA_KEY and AL2_KEY
End Product	Lock Boot Interface	No	No	**OEM** use requires PL0
Developer	RMA Requested	No	No	Flash memory erased except permanently locked blocks
Renesas	RMA Acknowledged	All	No	
Developer	RMA Return	No	No	Returned to the Developer or scrapped

Figure 1. RA8 MCU Device Lifecycles

Note that the Protection Level (PL) and Authentication Level (AL) can be changed only in OEM DLM State. If OEM state is used in the End Product, it is recommended to use PL0.

When transitioning to RMA_ACK, the contents of the flash memory, except permanently locked blocks or registers, are erased. Renesas can read the contents of the permanently locked blocks or registers during failure analysis. Transition to RMA_REQ is not possible if the AL2_KEY is disabled.

1.3 Device Lifecycle Management Keys

Authenticated Authentication Level transitions are possible using AL Keys. These AL Keys are injected through secure key injection during specific AL states to allow authenticated regression back to that state. The authentication is performed via a challenge-response mechanism, protecting against eavesdropping and replay attacks. These DLM Keys are 128-bit keys that can be injected into the unmapped flash area of the MCUs. The following is a description of these keys.

- AL1_KEY – The Authentication Level 1 Key can be injected in the OEM DLM state at AL2 or AL1. It is used to transition from AL0 to AL1 when in the OEM DLM state. AL1_KEY can be disabled permanently when in the OEM DLM state at either AL2 or AL1 by the boot firmware parameter setting command.
- AL2_KEY – The Authentication Level 2 Key can be injected in the OEM DLM state at AL2. It is used to transition from AL0 or AL1 to AL2 when in the OEM DLM state. AL2_KEY can be disabled permanently when in the OEM DLM state at AL2 by the boot firmware parameter setting command.
- RMA_KEY – The Return Material Authorization Key can be injected in the OEM DLM state at AL2. It is used to send the MCU to Renesas for failure analysis through the Return Material Authorization process. This key can be used in any of the OEM states to transition to the RMA_REQ state if AL2_KEY is not disabled.
- RMA_ACK_KEY – The Return Material Authorization Acknowledgment Key is a Renesas-specific key and is not used by the customers. This key is used to transition the MCU from the RMA_REQ to the RMA_ACK state.

1.4 Changing the Protection Levels and Authentication Levels

Protection Level and Authentication Level can be changed only via the factory boot firmware interfaces (UART, USB, SWD/JTAG) when in OEM DLM State, as shown in Figure 1.

1.4.1 Protection Level Transitions

Transition to a lower privileges (lower number) PL is always allowed. The device initialization operation can initialize the MCU to PL2. Transition to a higher PL requires the MCU to be at the same or higher AL level.

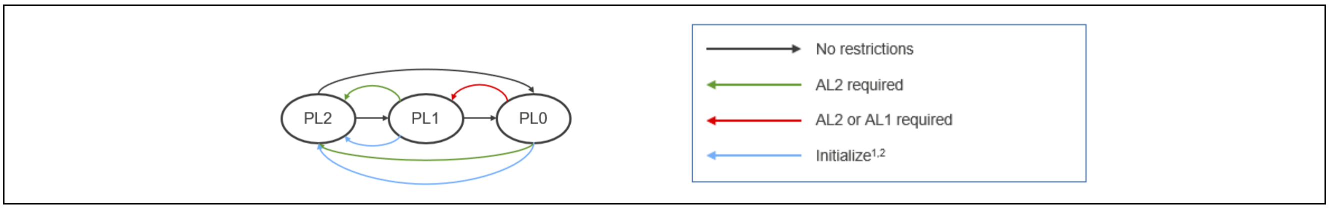


Figure 2. Protection Level Transitions

1.4.2 Authentication Level Keys and Authentication Level Transitions

Transition to lower privileges (lower number) AL is always permitted with no need for authentication. Transition to a higher-privileged Authentication Level requires authentication using the appropriate Authentication Level Key. The boot firmware only supports changing the Authentication Level from lower privilege to higher privilege. To return the Authentication Level to the lower privilege, the user should ensure the Protection Level is the desired setting, and then the user can reset the MCU to achieve the desired AL state.

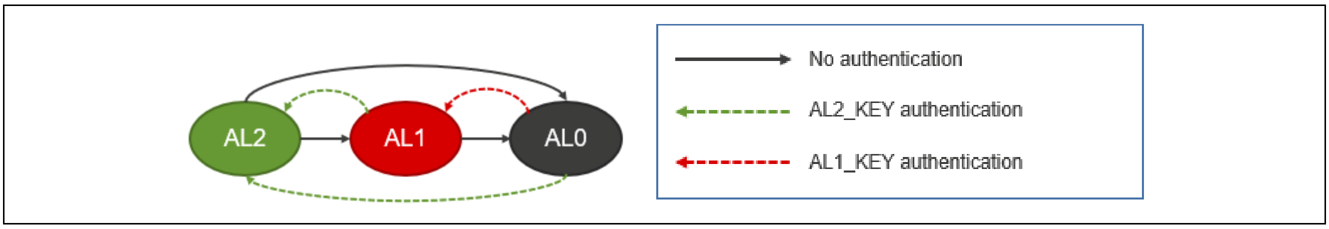


Figure 3. Authentication Level Transitions

As shown in Figure 1, the PL and AL states are applicable within the OEM Device Lifecycle State. For conciseness and alignment with development tool terminology, we will use “OEM_PLx” to represent the OEM DLM state with the indicated Protection Level.

2. RA8 Device Lifecycle Management Use Cases

This section discusses some of the use cases for Device Lifecycle Management for RA8 MCUs with TrustZone Support.

2.1 Immutable Root of Trust Provisioning

A Root of Trust (RoT) consists of data and services that provide a unique identity to a product. As such, it is vital to protect the code and data that are used to implement the Root of Trust. Arm TrustZone provides an excellent mechanism for isolating the Root of Trust from the rest of the application.

An important point to note is that the Root of Trust is typically immutable. This is done by using the permanent block protect feature of the MCU. Since permanent block protection disables the **Initialize** command, some DLM capabilities will not be available in this use case.

2.1.1 Development Phase

For MCUs, the Root of Trust is often implemented by the same development team that implements the entire application (that is, the Combined Development Model). During development, the Root of Trust can be programmed into the secure region with the device in the PL2 state. The PL state should then be changed to the PL1 state for application development and finally to PL0 for system testing. However, since all parties are trusted, DLM key usage during development is probably unnecessary. If a problem is found with either the Root of Trust or the application, the MCU can simply be initialized to erase all flash memory and return the DLM state to the PL2 state.

Note: Do not permanently lock flash blocks at the development phase to make the Root of Trust immutable. This will disable the **Initialize** command, and the MCU will not be able to be reprogrammed. For testing purposes, use the temporary block protect mechanism.

2.1.2 Production Phase

Production programming with a TrustZone-enabled application is often a two-step process. For the Root of Trust use case, the Root of Trust (in the secure region) will likely be provisioned separately from the application layer (in the non-secure region) to enable the injection of infrastructure keys.

The following shows the DLM options for a deployed MCU for the Root of Trust use case. For this use case, it is assumed that the RoT (Root of Trust) has been made immutable, rendering the **Initialize** command unavailable. Note that deploying the product in the PL0 state with no AL keys is not useful since the device can no longer be completely erased.

Table 3. RoT Use Case, Possible Production DLM/PL States

DLM/PL States	Initialize Command	AL Keys	Capabilities after Programming
OEM_PL0	Unavailable	AL2_KEY AL1_KEY	Limited boot mode interface access (1). Authenticated debug in AL1 and AL2 states. PL can be regressed to AL1 or AL2 after authentication.
LCK_BOOT	Unavailable	None	No access

Note: 1. MCU Unique ID, MCU type (for example, R7FA8D1BHECBD), and factory boot firmware version available.

If the end-product firmware will not be debugged if there is a problem with the end-product, the LCK_BOOT state is recommended. If the end-product firmware may be debugged, the OEM state with PL0 is required with an injected AL2_KEY and optionally an injected AL1_KEY.

2.2 IP Protection

The IP Protection use case consists of valuable IP provided in a pre-programmed MCU to another party, who then develops the end application. IP protection is critical, but it is also helpful to retain the ability to reprogram the IP to avoid MCU scrappage.

2.2.1 Development – Valuable IP

The valuable IP will be placed in the secure region, with the MCU in the OEM state with PL2. Upon delivery to the second party, the DLM state will remain OEM, and the Protection Level will be moved to PL1. There are some additional options available, as described in the following table.

Table 4. IP Protection Use Case, Possible IP Delivery PL States

DLM/PL States	Initialize Command	AL Keys	Capabilities after Programming
OEM_PL1	Available	None	Valuable IP is protected, but it can be erased by a second party, including accidental erasure. Cannot return to PL2 state to debug Valuable IP with the application.
OEM_PL1	Available	AL2_KEY	Valuable IP is protected, but it can be erased by a second party, including accidental erasure. Can perform authenticated transitions to the PL2 state to debug Valuable IP with the application.
OEM_PL1	Disabled	None	Valuable IP is protected, and it cannot be erased by the second party. Cannot return to PL2 state to debug Valuable IP with the application.
OEM_PL1	Disabled	AL2_KEY	Valuable IP is protected, and it cannot be erased by the second party. Can perform authenticated transitions to the AL2 state to debug Valuable IP with the application.

2.2.2 Development – Application

The second party will receive the device in the PL1 state. They can then develop and debug application code that uses the IP that was placed in the secure region, but they cannot view it.

To test the full application in a production configuration (i.e., the PL0 state), it is recommended to inject an AL1_KEY Key, for the following reasons:

- If the pre-programmed MCU was delivered with the **Initialize** command disabled, there is no way to return from the PL0 state to the PL1 state without using an authenticated transition with the AL1_KEY.
- If the pre-programmed MCU was delivered with the **Initialize** command enabled, using the **Initialize** command will erase both secure and nonsecure regions, including the purchased IP.

2.2.3 Production – Final Product

Since the usage of the AL2_KEY and AL1_KEY and the configuration of the **Initialize** command are all independent, there are eight variations for delivering the end product. Table 5 summarizes the options. Note that some of the options depend on the state in which the pre-programmed MCU is delivered.

Table 5. IP Protection Use Case, Possible Production DLM States

DLM/PL States	Initialize Command	DLM Keys	Capabilities after Programming
OEM_PL0	Disabled	AL2_KEY AL1_KEY	Limited boot mode interface access (1). Authenticated debug in AL1 and AL2 states. PL can be regressed to AL1 or AL2 after authentication.
OEM_PL0	Disabled	AL2_KEY	Limited boot mode interface access (1). Authenticated debug in AL2 state. PL can be regressed to AL2 after authentication.
OEM_PL0	Disabled	AL1_KEY	Limited boot mode interface access (1). Authenticated debug in AL1 state. PL can be regressed to AL1 after authentication.
OEM_PL0	Disabled	None	Limited boot mode interface access (1).
OEM_PL0	Enabled	AL2_KEY AL1_KEY	Limited boot mode interface access (1). Authenticated debug in AL1 and AL2 states. PL can be regressed to AL1 or AL2 after authentication. Device can be **Initialized.**
OEM_PL0	Enabled	AL2_KEY	Limited boot mode interface access (1). Authenticated debug in AL2 state. PL can be regressed to AL2 after authentication. Device can be **Initialized.**
OEM_PL0	Enabled	AL1_KEY	Limited boot mode interface access (1). Authenticated debug in AL1 state. PL can be regressed to AL1 after authentication. Device can be **Initialized.**
OEM_PL0	Enabled	None	Limited boot mode interface access (1). Device can be **Initialized.**
LCK_BOOT	Unavailable	None	No access

Note: 1. MCU Unique ID, MCU type (for example, R7FA8D1BHECBD), and factory boot firmware version available.

2.3 Non-TrustZone Usage

Arm TrustZone usage on an RA Family MCU is optional. The Flat Project Development Model supports the use case of creating an application without TrustZone awareness. In Non-TrustZone applications, the AL keys are still useful for the ability to provide authenticated re-enabling of the debug and boot mode programming interfaces.

2.3.1 Development Phase

It is not necessary to use AL keys during development. In a Flat Project, all development is done in the PL2 state. To test the application as it will be delivered in a production state, change the state to PL0. To reprogram the device, issue the **Initialize** command.

Note: Do not permanently lock any flash blocks at the development phase. This will disable the **Initialize** command, and the MCU will not be able to be reprogrammed. For testing purposes, use the temporary block protect mechanism.

2.3.2 Production Phase

If it is desired to enable authenticated debugging, the AL2_KEY can be used for this purpose and must be injected. AL2_KEY can be used to transition the PL state from PL0 to PL2. The DLM state must remain in the OEM state.

If the end-product may need to be reprogrammed but retention of the existing code flash is not required, the PL0 state can be used without the AL2_KEY providing that the DLM state remains in the OEM state and the Initialize command is not disabled.

Table 6 summarizes the options.

Table 6. Non-TrustZone Use Case, Possible Production DLM States

DLM/PL State	Initialize Command	AL Keys	Capabilities after Programming
OEM_PL0	Disabled or Unavailable	AL2_KEY	Limited boot mode interface access (1). Authenticated debug in AL2 state. PL can be regressed to AL2 after authentication.
OEM_PL0	Enabled	AL2_KEY	Limited boot mode interface access (1). Authenticated debug in AL2 state. PL can be regressed to AL2 after authentication. Device can be **Initialized.**
OEM_PL0	Disabled or Unavailable	None	Limited boot mode interface access (1).
OEM_PL0	Enabled	None	Limited boot mode interface access (1). Device can be **Initialized.**
LCK_BOOT	Unavailable	None	No access

Note: 1. MCU Unique ID, MCU type (e.g., R7FA8D1BHECBD), and factory boot firmware version available.

2.4 Renesas RMA

To submit the MCU for failure analysis by Renesas through the RMA process, an RMA_KEY must be injected, and the device must be in OEM state prior to performing a transition to the RMA_REQ state. Transition from OEM_PL2 to RMA_REQ is not possible if AL2_KEY is disabled. This transition will erase all code and data flash that has not been permanently locked, and the MCU can be delivered to Renesas for failure analysis.

This use case can be combined with all of the above use cases.

3. Non-Authenticated State Transitions

Non-authenticated DLM, PL, and AL state transitions do not require the use of DLM/AL keys. These transitions reflect a typical lifecycle flow from development through to production and deployment, with decreasing levels of access with each transition.

3.1 Non-Authenticated State Transition Using the Renesas Device Partition Manager

The Renesas Device Partition Manager (RDPM) is a utility integrated into the e^2 studio IDE for Device Lifecycle State management during development. The RDPM can perform the following functions:

- Query the current device lifecycle state
- Query the device TrustZone boundary setup
- **Initialize** the device to the OEM_PL2 state (unless any flash blocks have been permanently locked)
- Set up TrustZone boundary

In the e^2 studio IDE, open the **Renesas Device Partition Manager**.

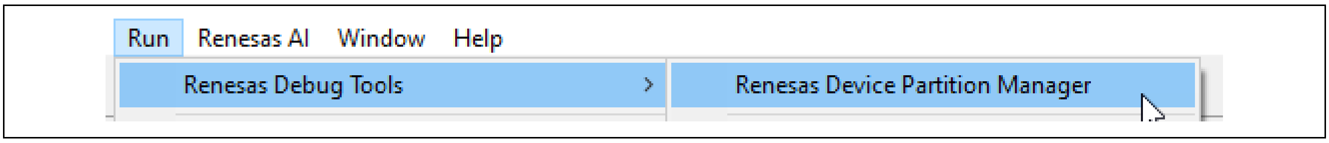
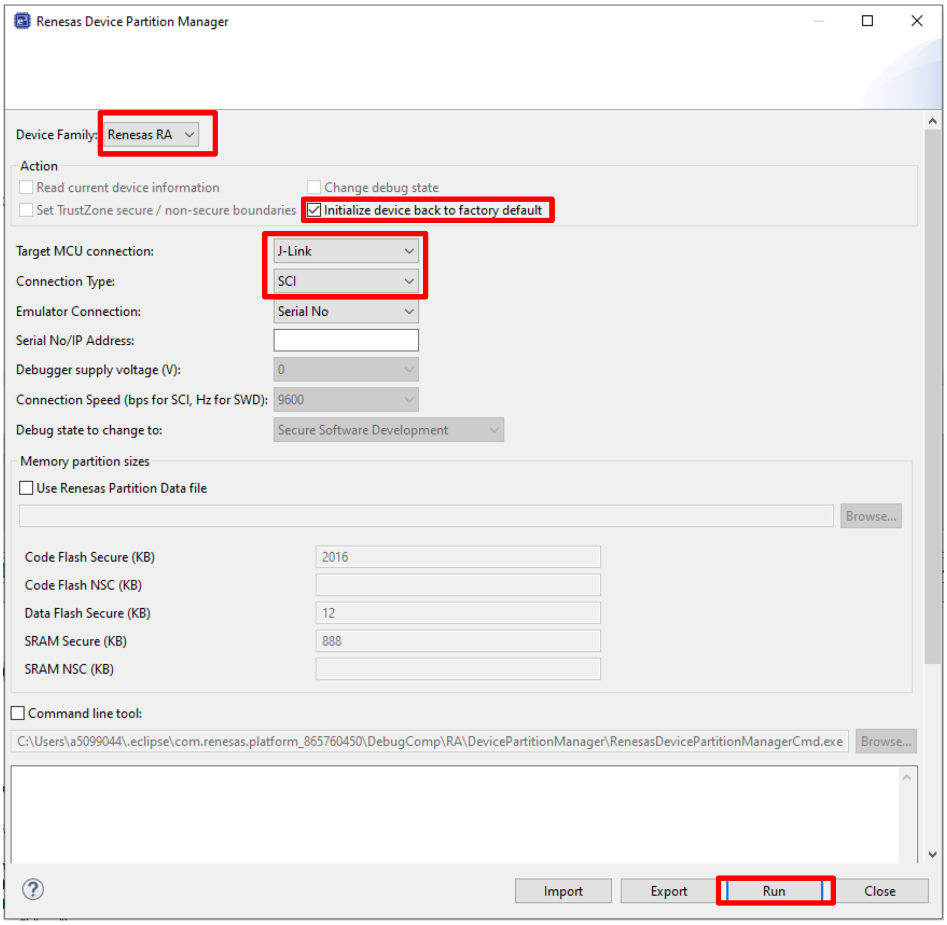


Figure 4. Open the Renesas Device Partition Manager

Figure 5 shows how to **Initialize** the device back to factory default. Choose the connection method and then click **Run**.



Device Family: Renesas RA

Action

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

Target MCU connection: J-Link

Connection Type: SCI

Emulator Connection: Serial No

Serial No/IP Address:

Debugger supply voltage (V): 0

Connection Speed (bps for SCI, Hz for SWD): 9600

Debug state to change to: Secure Software Development

Memory partition sizes

□ Use Renesas Partition Data file

Code Flash Secure (KB) 2016

Code Flash NSC (KB)

Data Flash Secure (KB) 12

SRAM Secure (KB) 888

SRAM NSC (KB)

□ Command line tool:

C:\Users\a5099044\.eclipse\com.renesas.platform_865760450\DebugComp\RA\DevicePartitionManager\RenesasDevicePartitionManagerCmd.exe

Figure 5. Initialize RA MCU Using Renesas Device Partition Manager

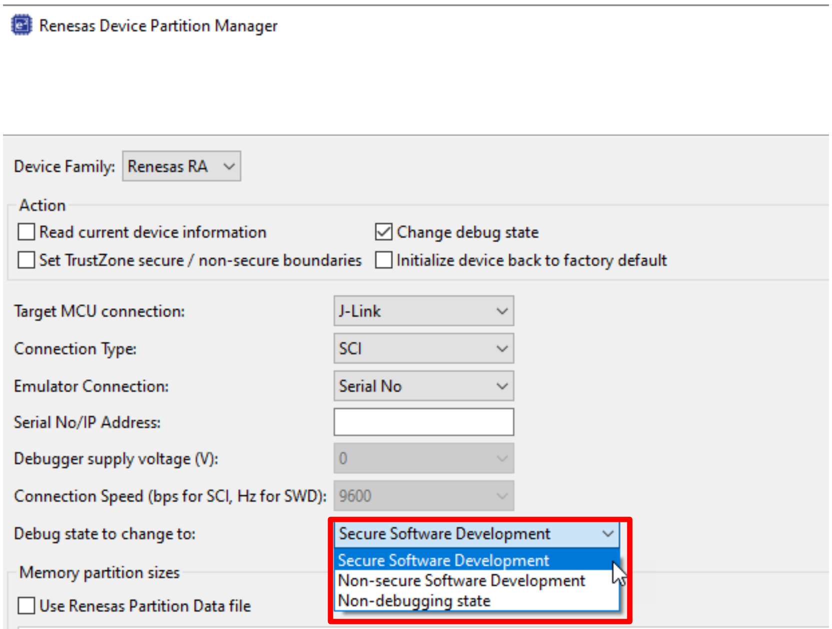
During product development, users typically use the RDPM to perform Protection Level advancements (CM to OEM_PL2, OEM_PL2 to OEM_PL1, OEM_PL1 to OEM_PL0) and to **Initialize** the device (erase all the device memories and set the Protection Level to OEM_PL2). The corresponding AL level will be achieved with the PL levels when using the RFP.

Limitations with the Renesas Device Partition Manager

- It does not support authenticated transitions.
- It supports transitions to limited device lifecycle states.
- It requires the e^2 studio IDE to operate.

For RA8 MCUs, when using the Renesa Device Partition Manager, the following are the matching states:

- Secure Software Development -> OEM_PL2
- Non-secure Software Development -> OEM_PL1
- Non-debugging State -> OEM_PL0



Renesas Device Partition Manager

Device Family: Renesas RA

Action

□ Read current device information ☑ Change debug state

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

Target MCU connection: J-Link

Connection Type: SCI

Emulator Connection: Serial No

Serial No/IP Address:

Debugger supply voltage (V): 0

Connection Speed (bps for SCI, Hz for SWD): 9600

Debug state to change to: Secure Software Development

Memory partition sizes

□ Use Renesas Partition Data file

Figure 6. Advance Device Lifecycle States Using Renesas Device Partition Manager

3.2 Non-Authenticated State Transitions Using the Renesas Flash Programmer

The Renesas Flash Programmer provides end-to-end production flow support. RFP provides the following functionalities.

- Supports all authenticated and non-authenticated state transitions.
- Supports DLM/AL key injection.
- Supports disabling the **Initialize** command. This may be desired if the device is deployed in the OEM_PL0 state and there is a requirement to avoid accidental flash content erasure. However, once the **Initialize** command is disabled, it cannot be re-enabled.
- Supports disabling of AL2 and AL1 authentication. This may be desired if the device is deployed in OEM_PL0 state and there is a requirement to disable debugging of the system. However, once the AL2 and AL1 authentication is disabled, it cannot be re-enabled.
- Supports disabling of entry to the LCK_BOOT transition. This may be desired if the device is deployed in PL0 state and there is a requirement to be able to recover the device in the future for other usage. However, once the LCK_BOOT transition is disabled, it cannot be re-enabled.

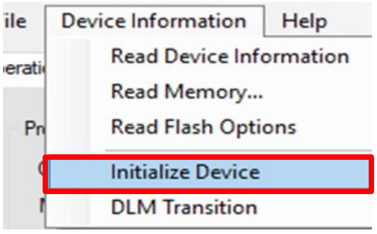
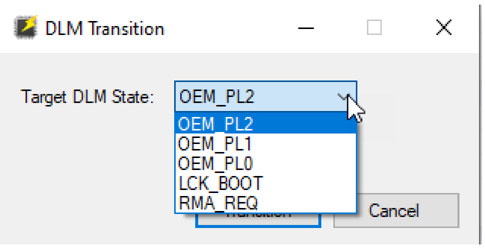


Figure 7. Using the Initialize Device Command

Non-authenticated transitions are described in Figure 2 and Figure 3can be performed as shown in Figure 8. The corresponding AL level will be achieved with the PL levels when using the RFP.

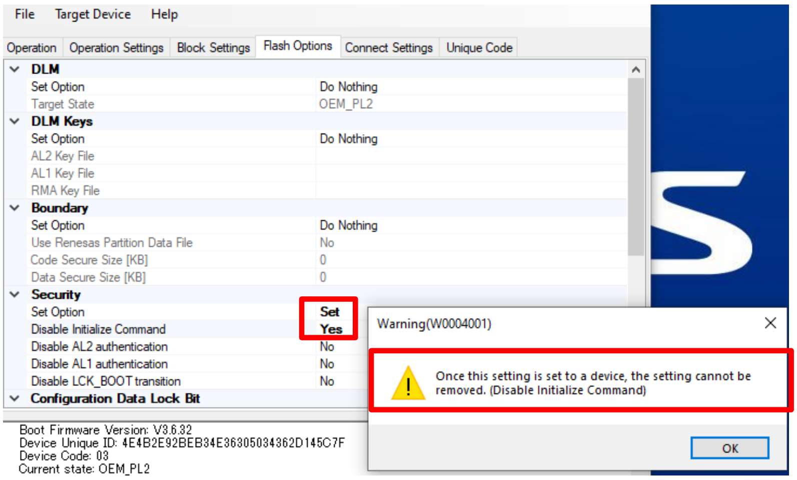

DLM Transition

Target DLM State: OEM_PL2

OEM_PL2
OEM_PL1
OEM_PL0
LCK_BOOT
RMA_REQ

Figure 8. Available Lifecycle State Transitions Supported by RFP

The **Initialize** command can be disabled as shown below. This operation cannot be reversed.



DLM	
Set Option	Do Nothing
Target State	OEM_PL2
DLM Keys	
Set Option	Do Nothing
AL2 Key File	
AL1 Key File	
RMA Key File	
Boundary	
Set Option	Do Nothing
Use Renesas Partition Data File	No
Code Secure Size [KB]	0
Data Secure Size [KB]	0
Security	
Set Option	**Set**
Disable Initialize Command	**Yes**
Disable AL2 authentication	No
Disable AL1 authentication	No
Disable LCK_BOOT transition	No
Configuration Data Lock Bit	

```
Boot Firmware Version: V3.6.32
Device Unique ID: 4E4B2E92BEB34E36305034362D145C7F
Device Code: 03
Current state: OEM_PL2
```

Warning(W0004001)

Once this setting is set to a device, the setting cannot be removed. (Disable Initialize Command)

Figure 9. Disable the Initialize Command

Similar warnings will also be printed if **Disable AL2 and/or AL1 authentication** or **Disable LCK_BOOT transition** option is selected.

4. DLM Authentication Key Creation and Injection

This section guides the user in generating an Authentication Key and injecting the AL key into the MCU. Generating and injecting the RMA_KEY uses a similar procedure.

4.1 Tools Used in the Authentication Key Injection Process

Three tools are used in the process of creating and injecting the DLM/AL key.

- Gpg4win - This tool is used to generate and use PGP keys, which are used to establish a secure communication channel between the developer and the Renesas Key Wrap server. Gpg4win is used here for demonstration, but any PGP management tool can be used.
- Renesas Security Key Management Tool (SKMT) - This tool generates the various key files needed for DLM/AL key injection.
- Renesas Flash Programmer (RFP) - This tool is used to inject DLM/AL keys.

4.2 Authentication Key Injection Overview

There are three high-level steps required to inject DLM/AL keys into the MCU.

1. Create an arbitrary 256-bit User Factory Programming Key (UFPK) and obtain a Wrapped UFPK (W-UFPK). The UFPK is used to encrypt the DLM/AL key, to ensure that no plaintext keys are

exposed during the injection process. The UFPK must be wrapped by the Renesas Key Wrapping Service to obtain the W-UFPK.

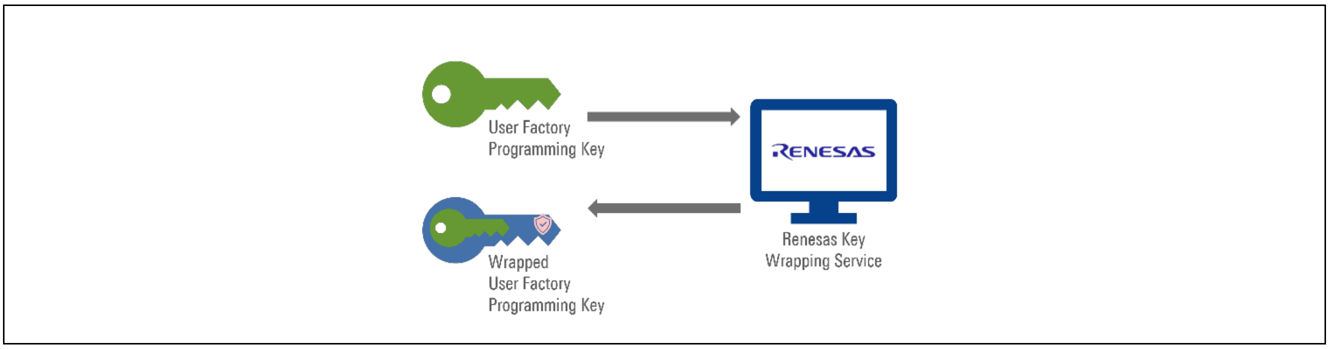


Figure 10. Wrap the User Factory Programming Key (UFPK)

2. Wrap the DLM/AL key using the UFPK.

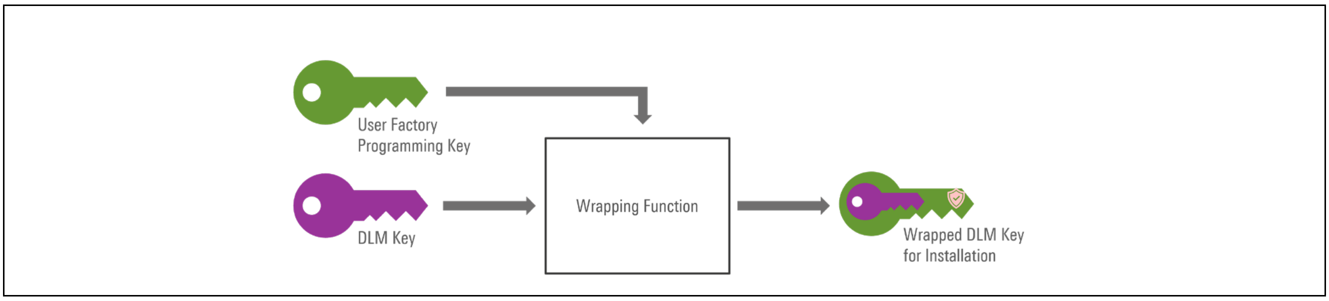


Figure 11. Wrap the DLM/AL Key Using UFPK

3. Inject the DLM/AL key using the serial programming interface by providing the W-UFPK and the wrapped DLM/AL key.

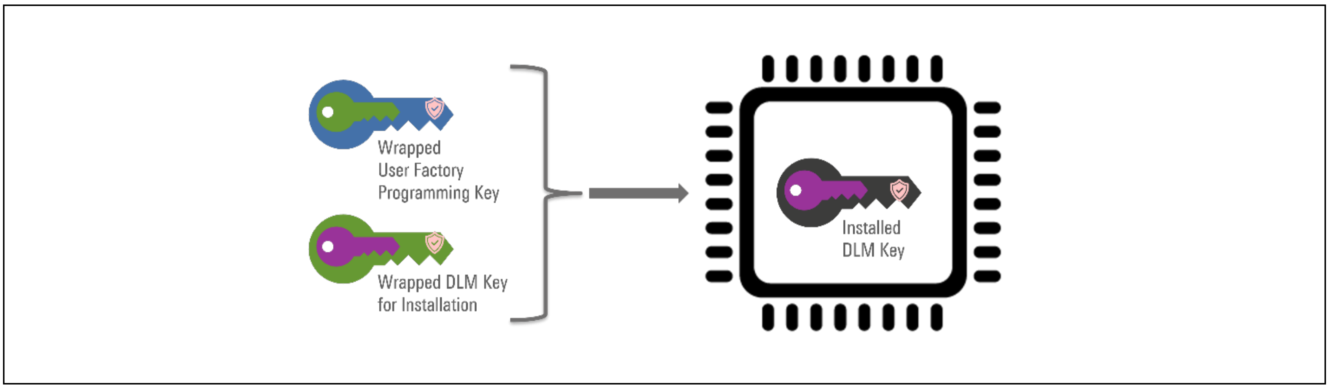


Figure 12. Created Wrapped DLM/AL Keys

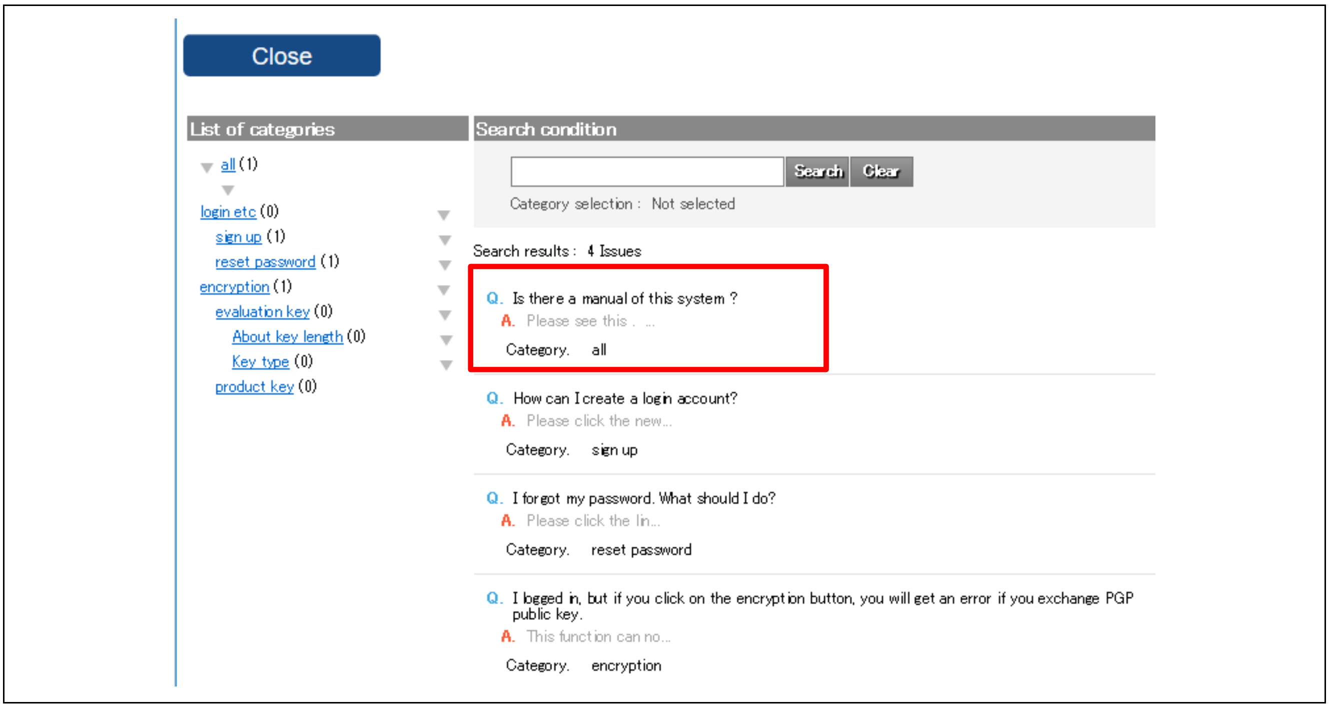
4.3 Establish PGP-Encrypted Communication with the Renesas DLM Server

The DLM/AL key material is exchanged with the Renesas DLM server using PGP encryption. This requires an exchange of public keys. This is a one-time process.

4.3.1 Overview of Device Lifecycle Management (DLM) Server

Using a web browser, enter the URL https://dlm.renesas.com/ to access the Renesas DLM server.

The *Key Wrap Service User's Manual* can be accessed by clicking on the FAQ link on the right of the screen. The link to the *DLM server User's Manual* is in the answer to the FAQ question, "Is there a manual of this system?" as shown in **Figure 13**.



Close

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

Search Clear

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 lin...
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 13. DLM Server FAQ and User's Manual

The information communicated between the user and the DLM server needs to be encrypted by OpenPGP. **Figure 14** shows an overview of the operational flow for the Key Wrapping service with OpenPGP encryption as a security measure to protect the DLM Keys in transit.

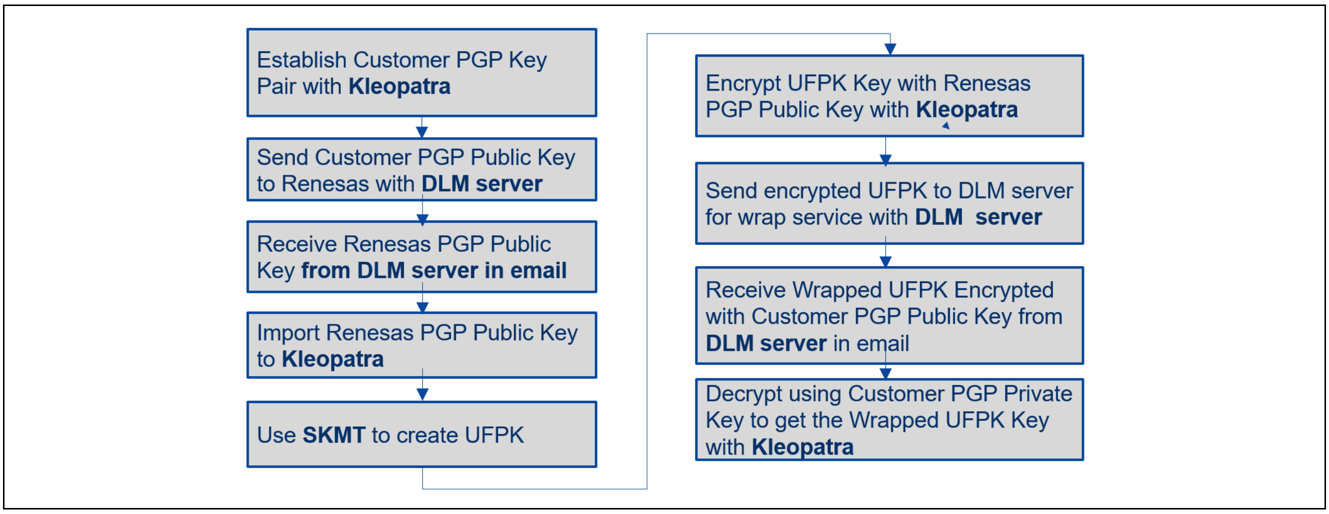

Establish Customer PGP Key Pair with Kleopatra
Send Customer PGP Public Key to Renesas with DLM server
Receive Renesas PGP Public Key from DLM server in email
Import Renesas PGP Public Key to Kleopatra
Use SKMT to create UFPK
Encrypt UFPK Key with Renesas PGP Public Key with Kleopatra
Send encrypted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

Figure 14. Overview of DLM Key Wrapping Service Using PGP

4.3.2 Create a PGP Key Pair

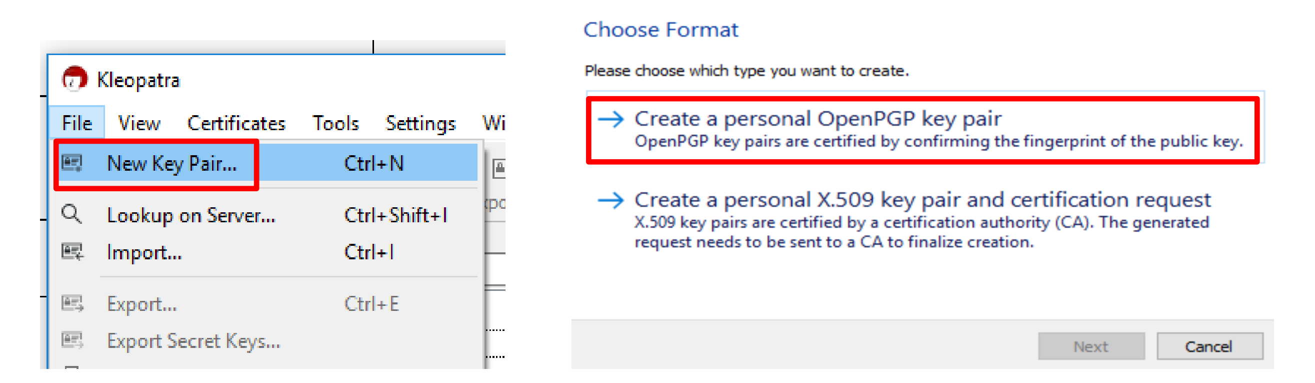
If you already have a PGP key pair, you can skip this step. Otherwise, use the following steps to create an OpenPGP key pair.

This Application Note uses Gpg4win, the official GnuPG distribution for Windows®, for the PGP key generation, encryption, and decryption service. Note that the generated private key needs to be managed securely and protected from exposure.

Gpg4Win can be downloaded from this URL: http://www.gpg4win.org/

Kleopatra is the certificate manager and unified crypto GUI provided in Gpg4Win. The screenshots included in this application note are based on Gpg4win-4.0.0. There may be minor graphic interface updates with later versions; however, the functionality used in this application note should persist.

Launch Kleopatra. Click File->New Key Pair and Create a personal OpenPGP key pair.



Choose Format

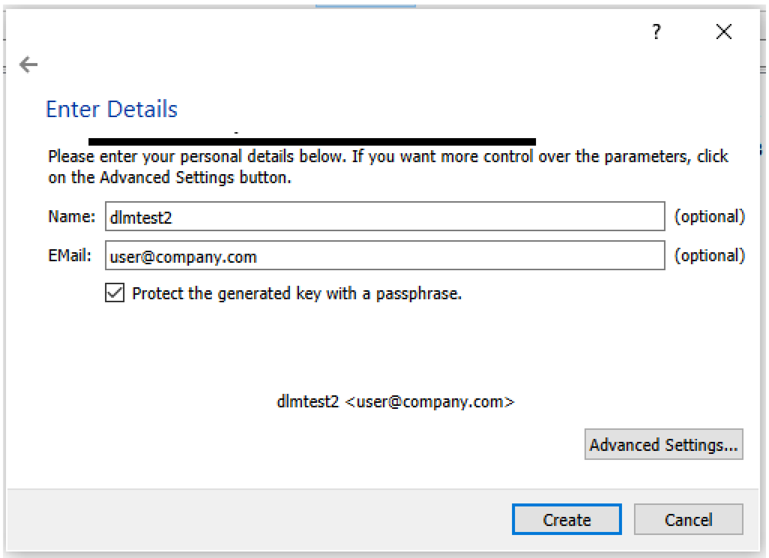
Please choose which type you want to create.

→ 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 15. Generate PGP Key Pair

Enter your details as shown. Check Protect the generated key with a passphrase.



Enter Details

Please enter your personal details below. If you want more control over the parameters, click on the Advanced Settings button.

Name: dlmtest2 (optional)

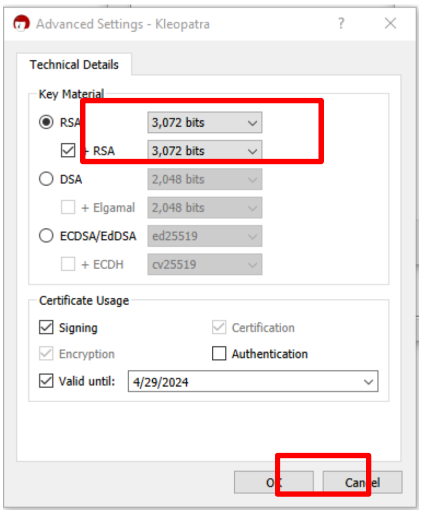
EMail: user@company.com (optional)

☑ Protect the generated key with a passphrase.

dlmtest2 <user@company.com>

Figure 16. Provide Credentials to Create the PGP Key Pair

Click **Advanced Settings** and select **RSA** and **3,072 bits**. Click **OK** and then click **Create**.



Key Material

● RSA 3,072 bits

☑ + RSA 3,072 bits

○ DSA 2,048 bits

□ + Elgamal 2,048 bits

○ ECDSA/EdDSA ed25519

□ + ECDH cv25519

Certificate Usage

☑ Signing ☑ Certification

☑ Encryption □ Authentication

☑ Valid until: 4/29/2024

Figure 17. Advanced Settings for the PGP Key pair

Provide a passphrase to protect the private key. Be sure to save your passphrase.

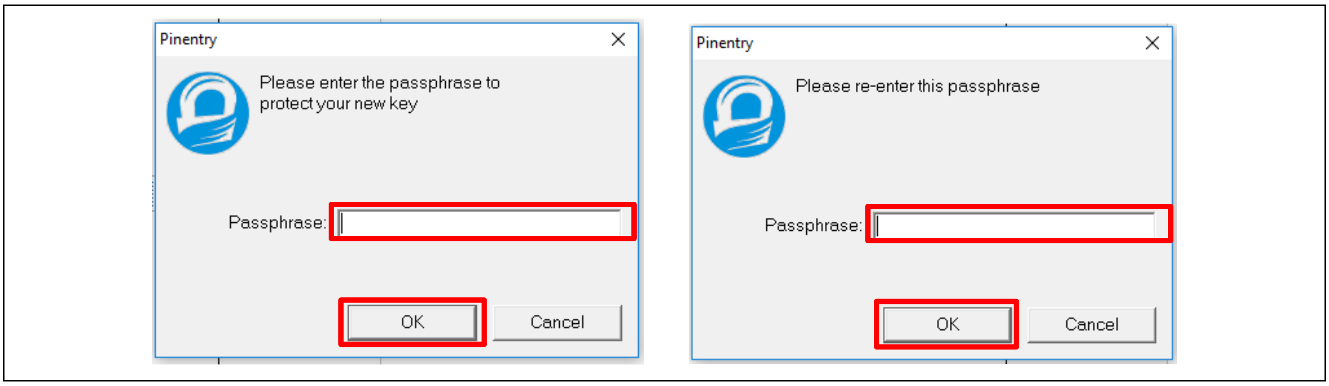
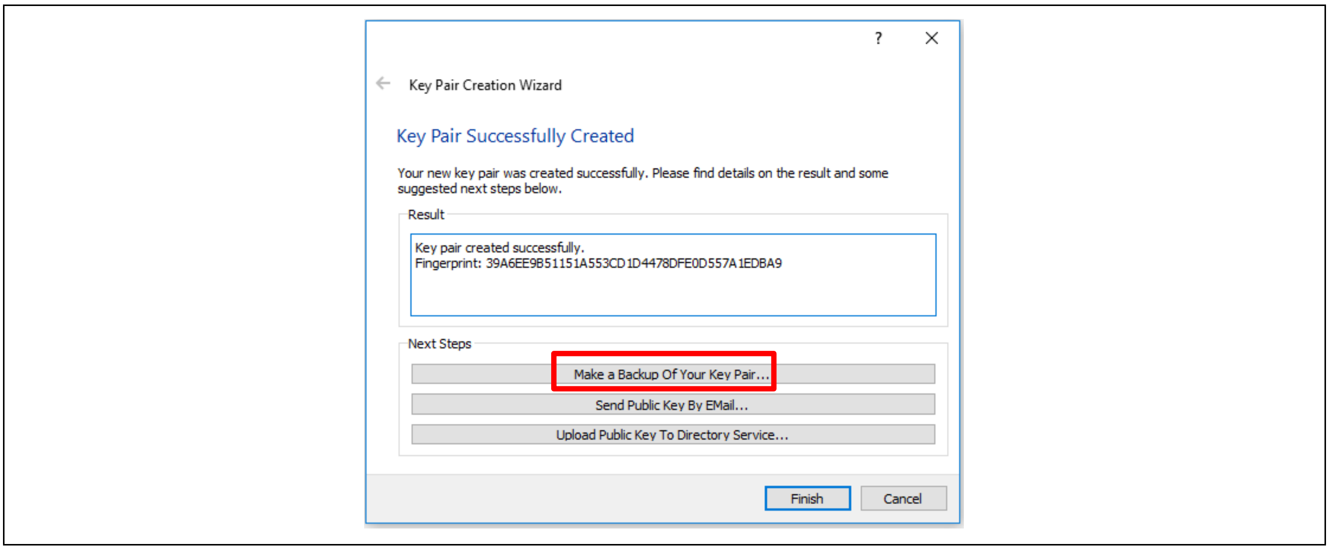

Pinentry
Please enter the passphrase to protect your new key
Passphrase:
OK
Cancel
Pinentry
Please re-enter this passphrase
Passphrase:
OK
Cancel

Figure 18. Provide passphrases

Observe that the PGP key pair is created successfully. It is recommended that you create a backup of your key pair.



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 19. Key Pair Successfully Created

Export your public key by right-clicking the name you just created and selecting **Export**, as shown below.

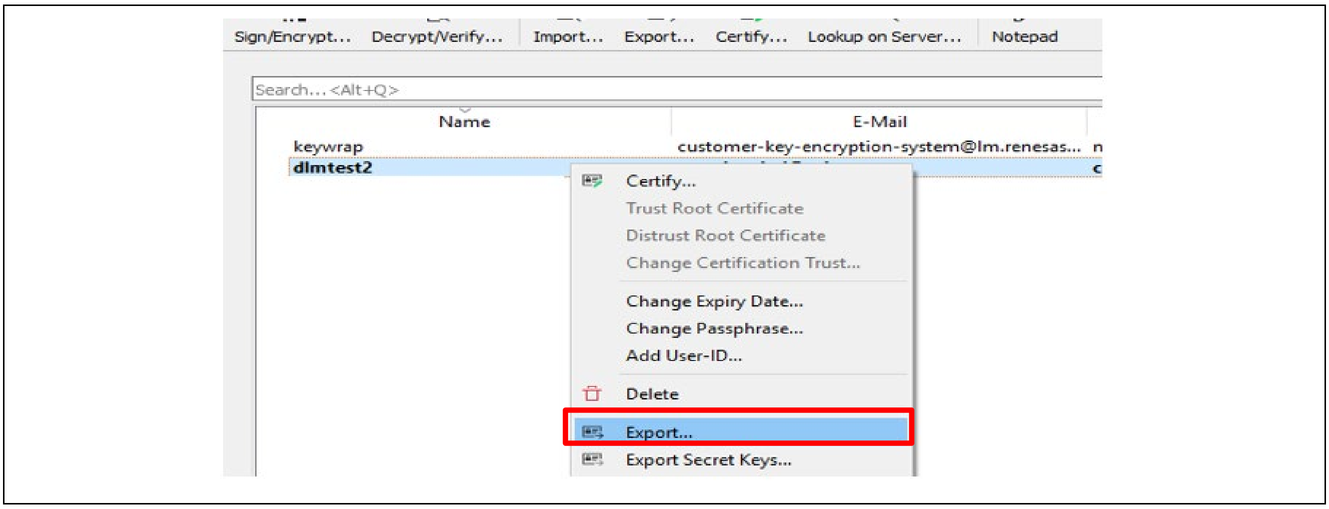
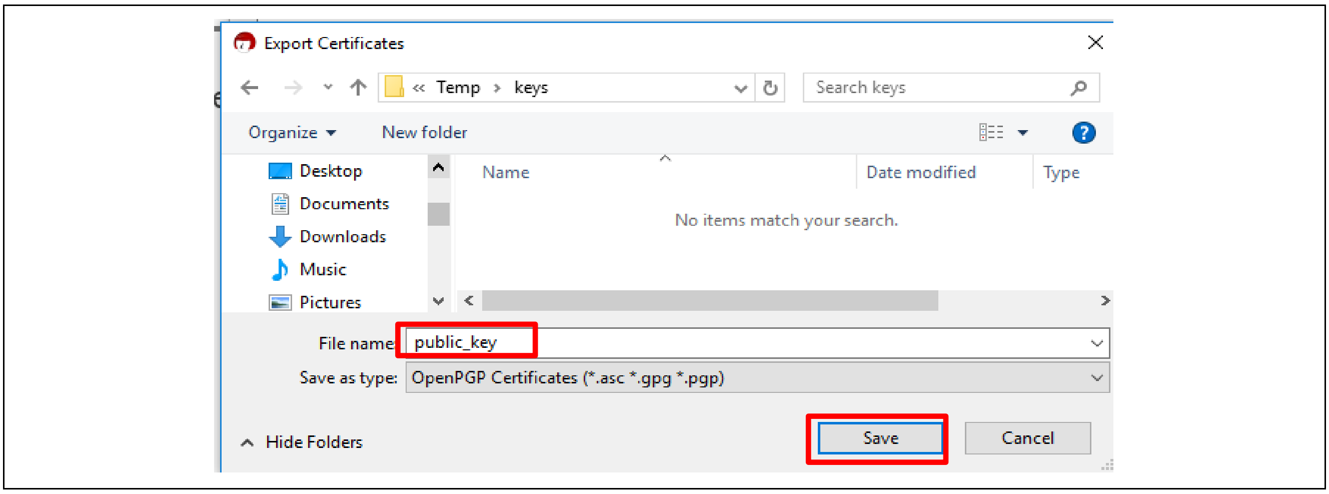

Sign/Encrypt... Decrypt/Verify... Import... Export... Certify... Lookup on Server... Notepad
Search... <Alt+Q>
Name
E-Mail
keywrap
customer-key-encryption-system@lm.renesas...
dlmtest2
Certify...
Trust Root Certificate
Distrust Root Certificate
Change Certification Trust...
Change Expiry Date...
Change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

Figure 20. Export the Public key

Save your public key to a file with `*.asc` extension, for example `public_key.asc`.



File name: public_key

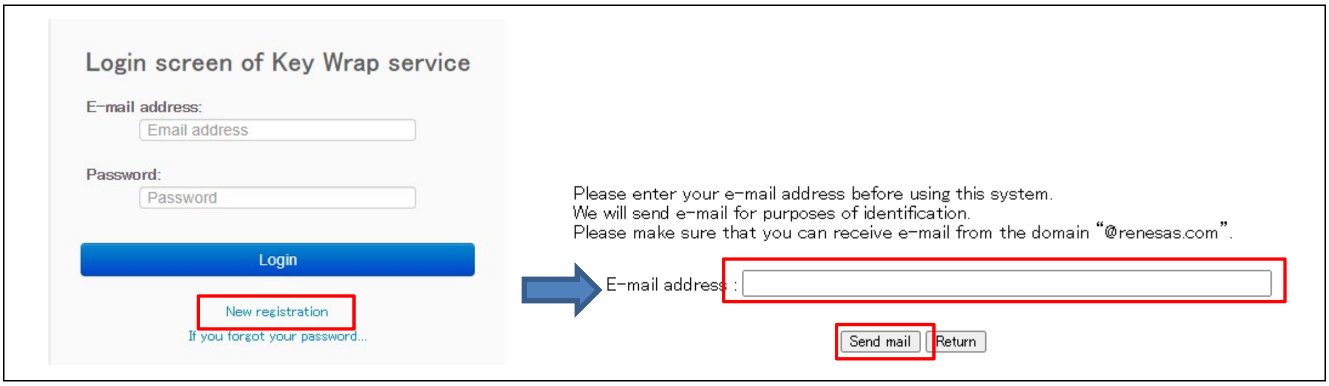
Save as type: OpenPGP Certificates (*.asc *.gpg *.pgp)

Figure 21. Save the Public Key

4.3.3 Registration with DLM Server

This section provides a brief walk-through of the DLM server registration steps. This is a one-time process for new users. You can also review the section “New registration” in the DLM user manual for further details on the new registration steps.

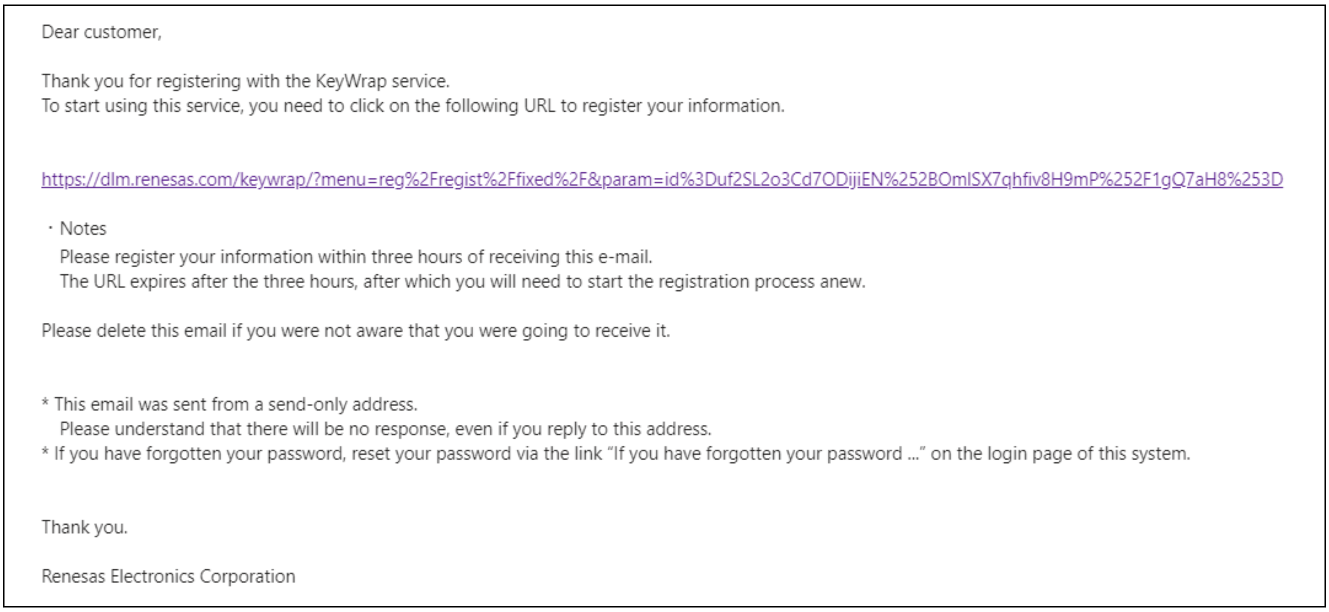
Open the URL https://dlm.renesas.com/ in a web browser and click **New registration**. Follow the prompt to provide an email address and click **Send mail**.



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 22. Provide Email Address for New Registration

You will receive the registration link in an email, as shown in **Figure 23**.



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%252BOmISX7qhfiv8H9mP%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.

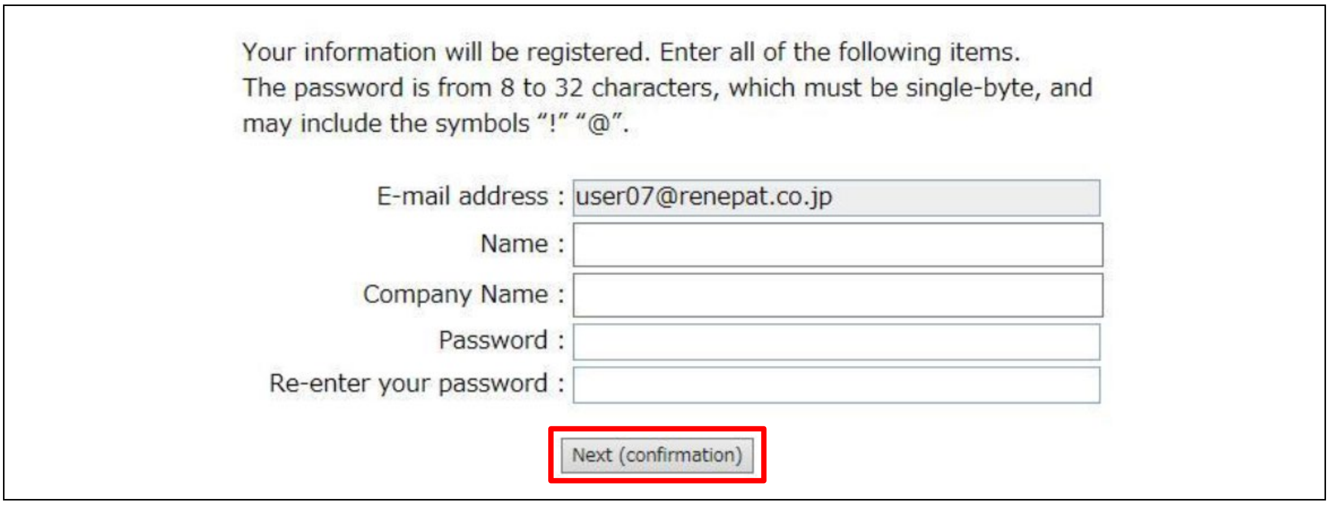
* This email was sent from a send-only address.
Please understand that there will be no response, even if you reply to this address.
* If you have forgotten your password, reset your password via the link "If you have forgotten your password ..." on the login page of this system.

Thank you.

Renesas Electronics Corporation

Figure 23. Email with Registration Link

Click on the URL in the confirmation email and provide the requested information. Click the **Next (confirmation)** button. After the confirmation screen is displayed, click the **Registration** button to complete the user registration.



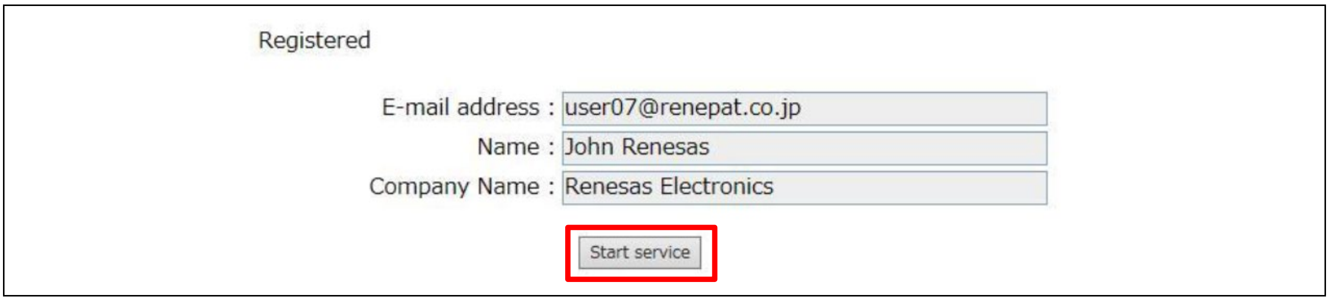
Your information will be registered. Enter all of the following items.
The password is from 8 to 32 characters, which must be single-byte, and may include the symbols "!" "@".

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

Figure 24. Register Customer Information

4.3.4 PGP Public Key Exchange

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

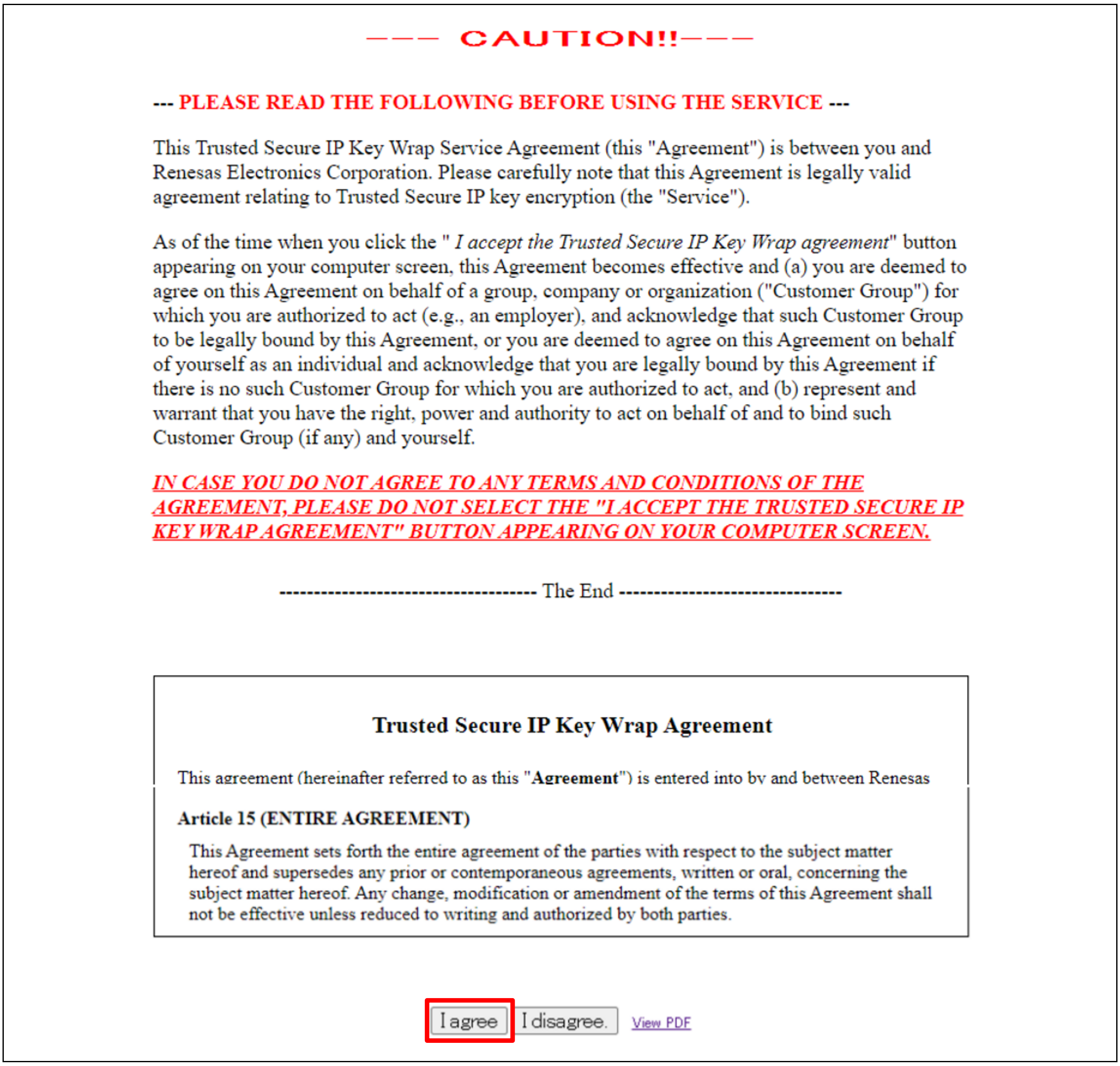


Registered

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

Figure 25. Start Using the DLM Service

Accept the **Trusted Secure IP Key Wrap Agreement** as shown below by clicking **Agree**. Note that this Agreement will come up whenever you log into the DLM server.



——— CAUTION!!———

--- PLEASE READ THE FOLLOWING BEFORE USING THE SERVICE ---

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

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

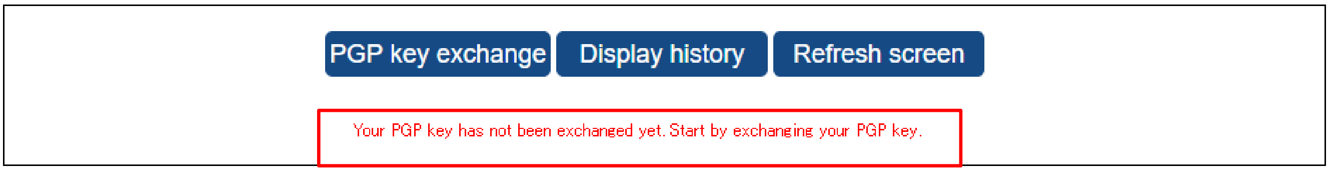
Article 15 (ENTIRE AGREEMENT)

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 26. Agree with the Key Wrap Agreement

Communication between the user and the DLM server uses PGP to encrypt all transferred data. Public PGP keys must be exchanged to enable these transfers.

If you try to transfer data to the key wrap service without exchanging PGP keys, you will receive the error message shown below.

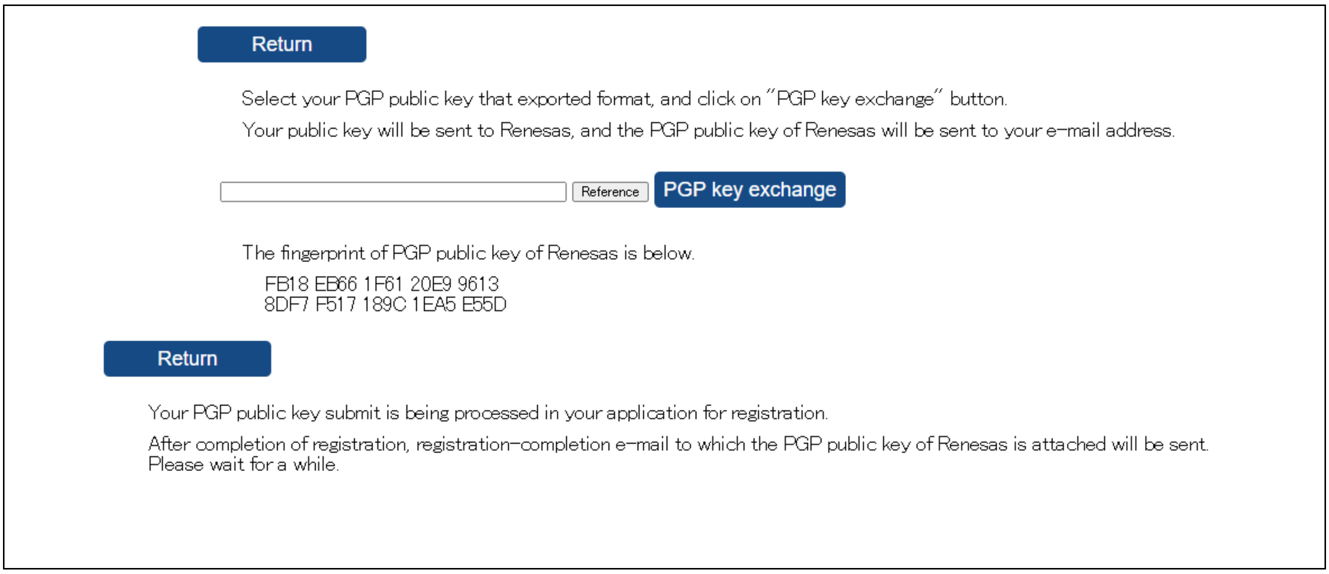


Your PGP key has not been exchanged yet. Start by exchanging your PGP key.

Figure 27. Request for PGP Key Exchange

To exchange PGP keys, click the **PGP key exchange** button. The dialog shown in **Figure 28** will open. Click **Reference** and select the public key that you created and exported earlier (section 4.3.2, in **Figure 21**).

Next, click **PGP key exchange** and wait to receive the Renesas PGP public key at the email address you provided during registration.



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

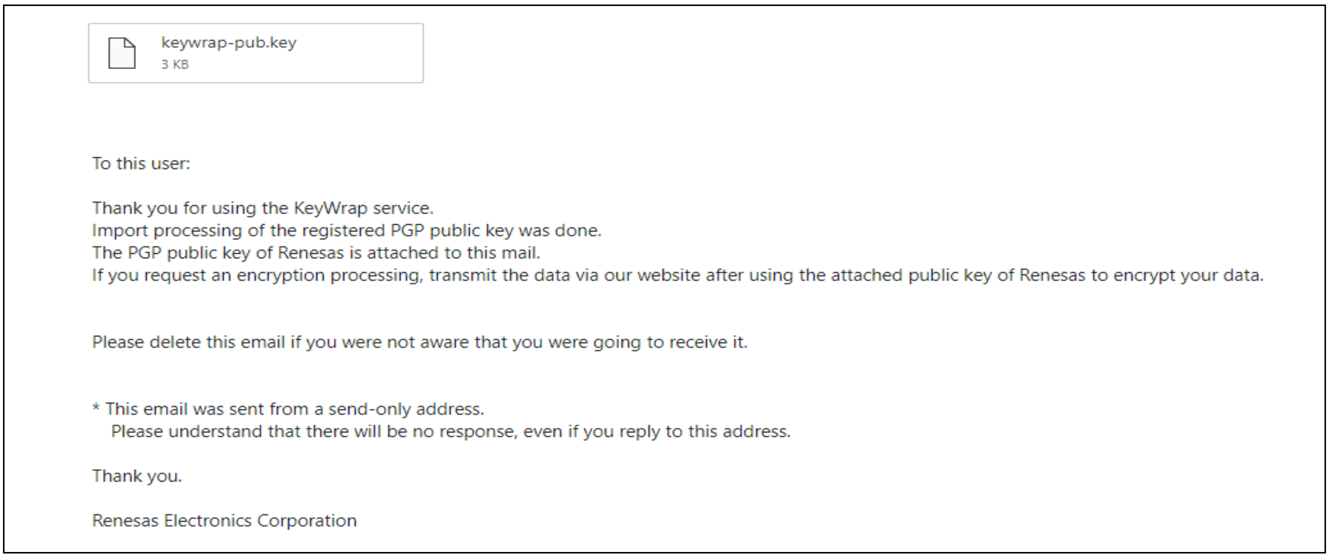
Your PGP public key submit is being processed in your application for registration.

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 28. Provide PGP Public Key to DLM Server

You will receive an email with the content as shown in **Figure 29**, including the Renesas public key. Save the Renesas PGP public key received (`keywrap-pub.key`). Note that the fingerprint of this key is provided on the PGP key exchange dialog, as shown in **Figure 28**.

PGP keys can be exchanged any number of times. If keys are exchanged multiple times, all previous PGP keys are discarded, and the key wrap service will use the latest PGP public key that has been successfully exchanged to encrypt the transferred data.



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 29. Receive the Renesas PGP Public Key

4.3.5 Import Renesas PGP Public Key into Kleopatra

Go back to the Kleopatra application and import the Renesas PGP Public key into Kleopatra, as shown in **Figure 30**.

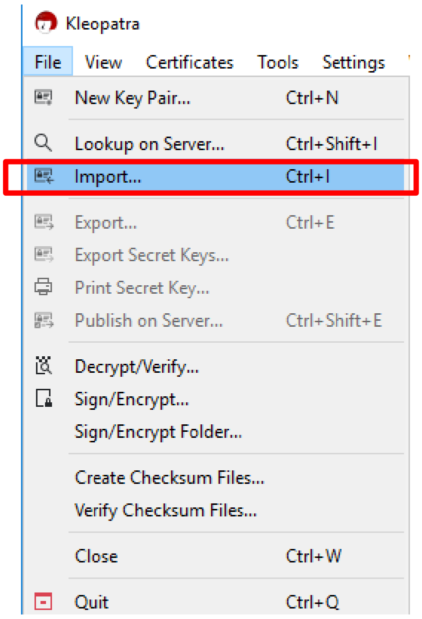
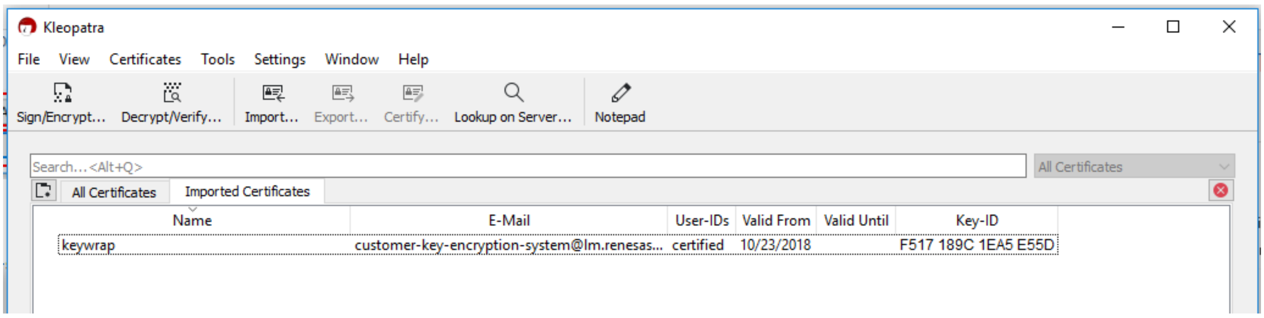


Figure 30. Import Renesas Public Key into Kleopatra

Click **File->Import**, and select the `keywrap-pub.key` file received from the key wrap service.

The following item will appear in the **Imported Certificates** in Kleopatra, and the Renesas public key is ready to be used.


Kleopatra
File View Certificates Tools Settings Window Help
Sign/Encrypt... Decrypt/Verify... Import... Export... Certify... Lookup on Server... Notepad
Search...<Alt+Q>
All Certificates
All Certificates Imported Certificates

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 31. Renesas PGP Public Key

4.4 Create the UFPK and W-UFPK

This section walks the user through the process of creating a UFPK and obtaining the W-UFPK.

4.4.1 Introduction to Renesas Security Key Management Tool

The Renesas Security Key Management Tool (SKMT) performs several functions during the DLM/AL key injection process. The SKMT is available from the following link:

https://www.renesas.com/software-tool/security-key-management-tool

Locate the **Downloads** area, download the latest Security Key Management Tool installer, and install it as required for your operating environment. This tool supports Windows and Linux.

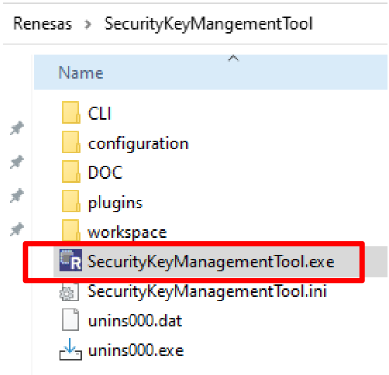
The User's Manual of this tool is located in the `\DOC` folder. We recommend that you read through the User's Manual before proceeding to the following section.

The SKMT provides two interfaces to users: a Command Line Interface (CLI) and a Graphical User Interface (GUI). The GUI interface is primarily intended for development usage. The CLI interface is primarily intended for production support or development efforts involving multiple keys due to its ability to create key file-generation scripts. This application note primarily uses the GUI tool to demonstrate the functionality. The SKMT User's Manual provides extensive examples of using the CLI for customers to reference.

4.4.2 Create a UFPK Key File

The Security Key Management Tool (SKMT) can be used in either Graphic User Interface (GUI) or Command Line Interface (CLI) mode to create a UFPK key file. This example uses the GUI interface.

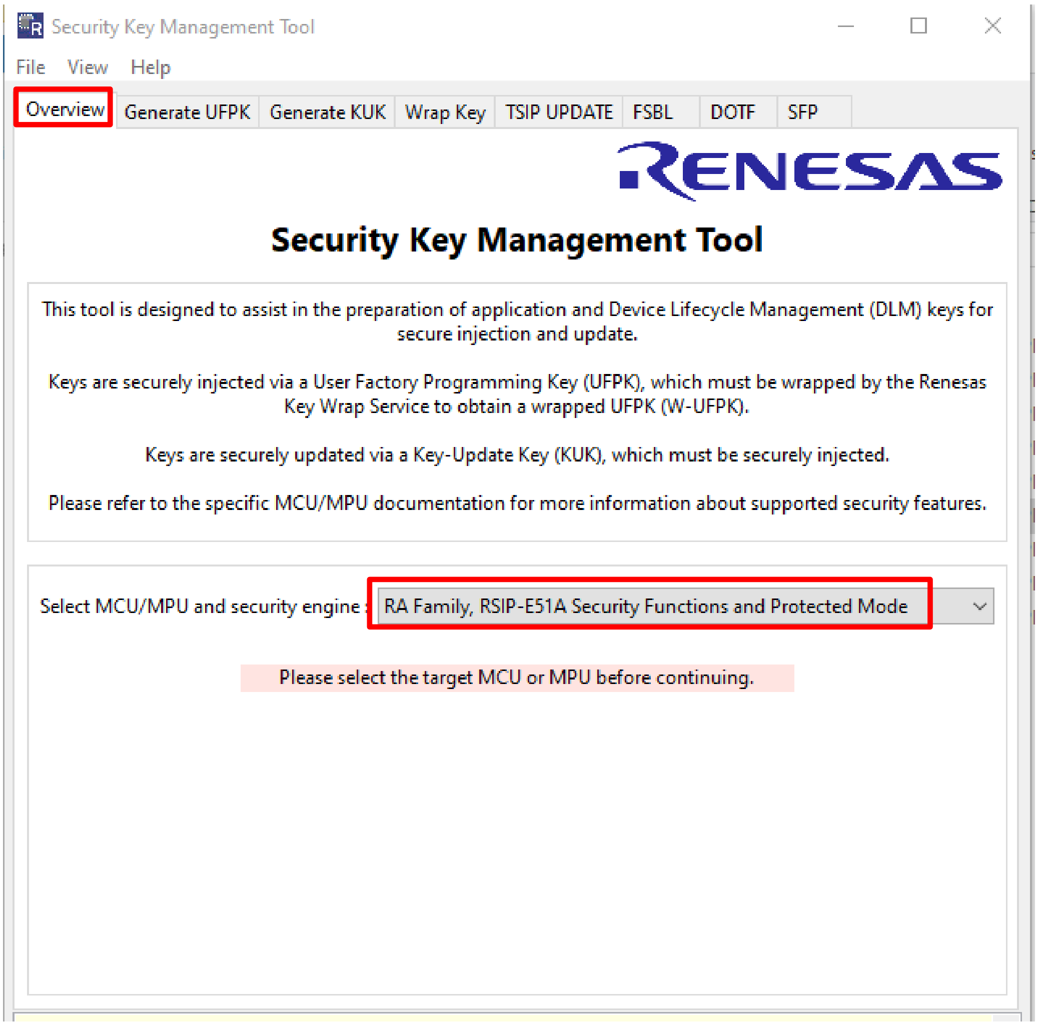
Launch the **Security Key Management Tool** executable.



Renesas > SecurityKeyMangementTool
Name
CLI
configuration
DOC
plugins
workspace
SecurityKeyManagementTool.exe
SecurityKeyManagementTool.ini
unins000.dat
unins000.exe

Figure 32. Launch SKMT GUI Interface

In the **Overview** window, select the microcontroller/microprocessor and security engine as **RA Family, RSIP-E51A Security Functions, and Protected Mode**.



Security Key Management Tool

This tool is designed to assist in the preparation of application and Device Lifecycle Management (DLM) keys for secure injection and update.

Keys are securely injected via a User Factory Programming Key (UFPK), which must be wrapped by the Renesas Key Wrap Service to obtain a wrapped UFPK (W-UFPK).

Keys are securely updated via a Key-Update Key (KUK), which must be securely injected.

Please refer to the specific MCU/MPU documentation for more information about supported security features.

Select MCU/MPU and security engine : RA Family, RSIP-E51A Security Functions and Protected Mode

Please select the target MCU or MPU before continuing.

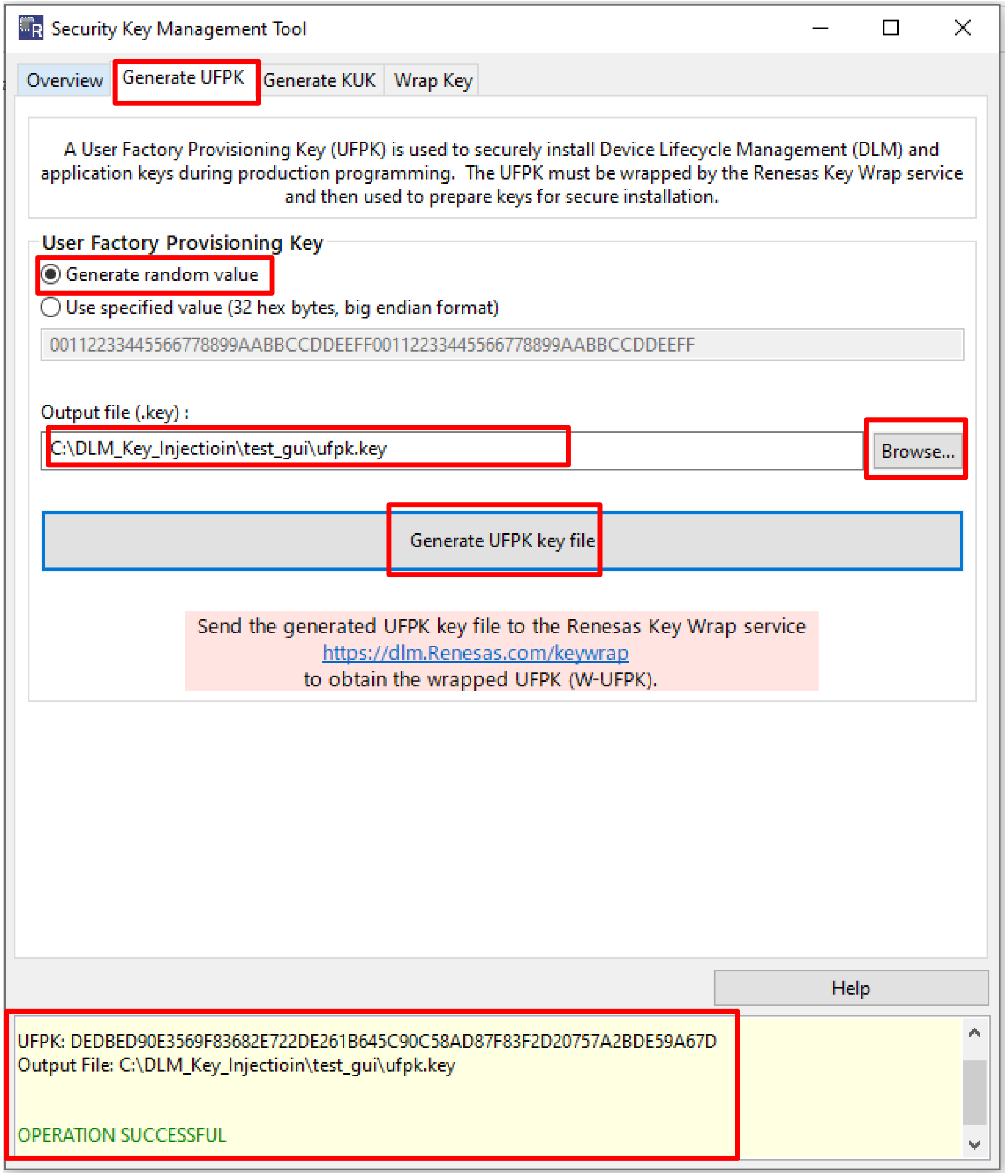
Figure 33. Select RA Family, RSIP-E51A Security Functions and Protected Mode

Next, click on the **Generate UFPK** tab.

As shown in Figure 34 and Figure 35, you can specify a desired value for the UFPK, or the tool can create a random value. Figure 34 shows the tool generating a random value by selecting **Generate random value**. Figure 35 shows specifying a specific value by selecting **User specified value** and entering the 32-byte key value in big-endian format.

Click the **Browse** button to enter a file name for the generated key file. Here, we have chosen `ufpk.key`.

Finally, click the button **Generate UFPK key file** to generate the key file `ufpk.key`.



A User Factory Provisioning Key (UFPK) is used to securely install Device Lifecycle Management (DLM) and application keys during production programming. The UFPK must be wrapped by the Renesas Key Wrap service and then used to prepare keys for secure installation.

User Factory Provisioning Key

● Generate random value

○ Use specified value (32 hex bytes, big endian format)

00112233445566778899AABBCCDDEEFF00112233445566778899AABBCCDDEEFF

Output file (.key) : C:\DLM_Key_Injectioin\test_gui\ufpk.key

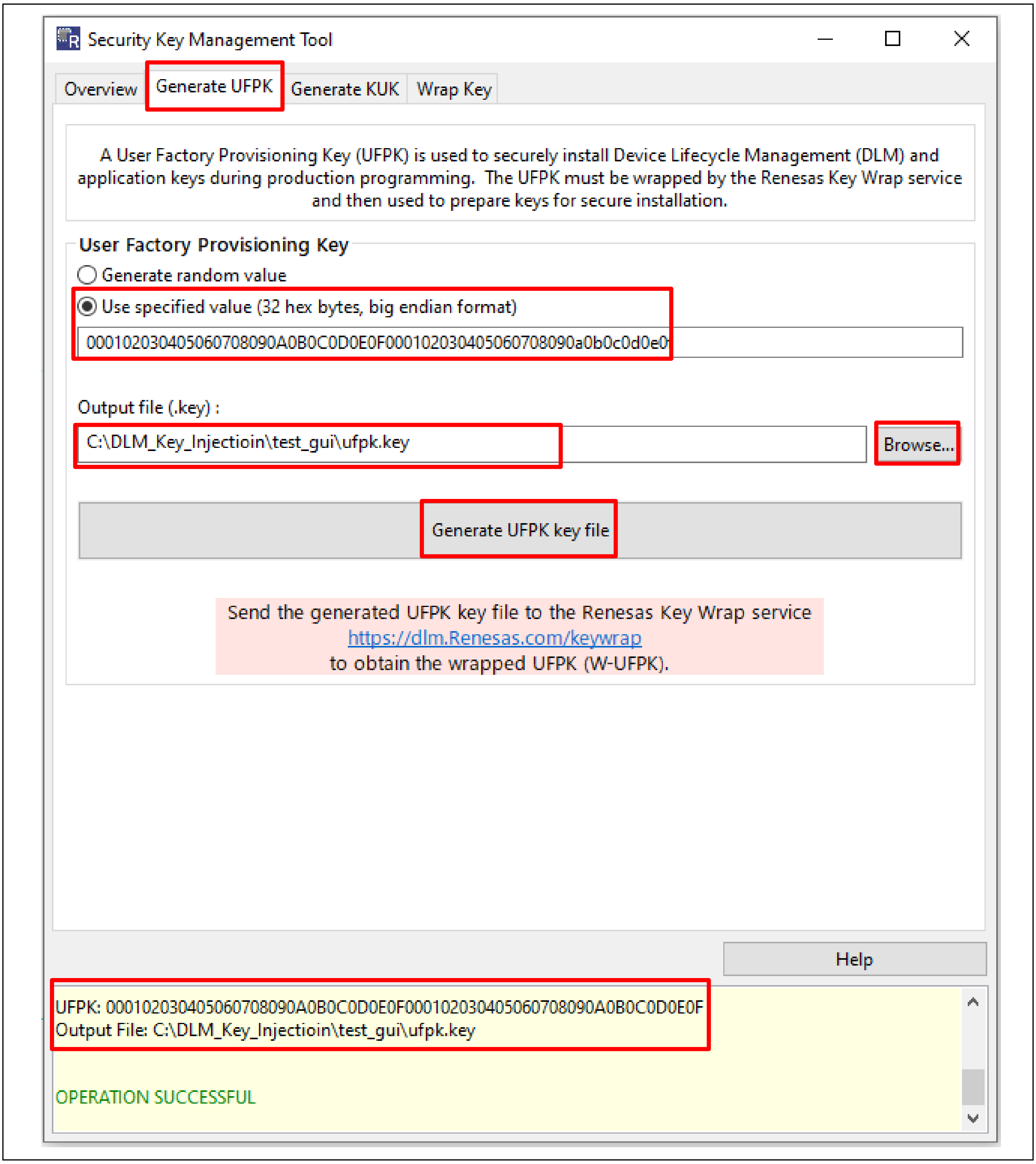
Send the generated UFPK key file to the Renesas Key Wrap service https://dlm.Renesas.com/keywrap to obtain the wrapped UFPK (W-UFPK).

```
UFPK: DEDBED90E3569F83682E722DE261B645C90C58AD87F83F2D20757A2BDE59A67D
Output File: C:\DLM_Key_Injectioin\test_gui\ufpk.key


OPERATION SUCCESSFUL
```

Figure 34. Generate a Random UFPK Using the GUI

In the next example, the following 32-byte key is used:

`000102030405060708090A0B0C0D0E0F000102030405060708090a0b0c0d0e0f`



A User Factory Provisioning Key (UFPK) is used to securely install Device Lifecycle Management (DLM) and application keys during production programming. The UFPK must be wrapped by the Renesas Key Wrap service and then used to prepare keys for secure installation.

User Factory Provisioning Key

○ Generate random value

● Use specified value (32 hex bytes, big endian format)

000102030405060708090A0B0C0D0E0F000102030405060708090a0b0c0d0e0f

Output file (.key) :

C:\DLM_Key_Injectioin\test_gui\ufpk.key

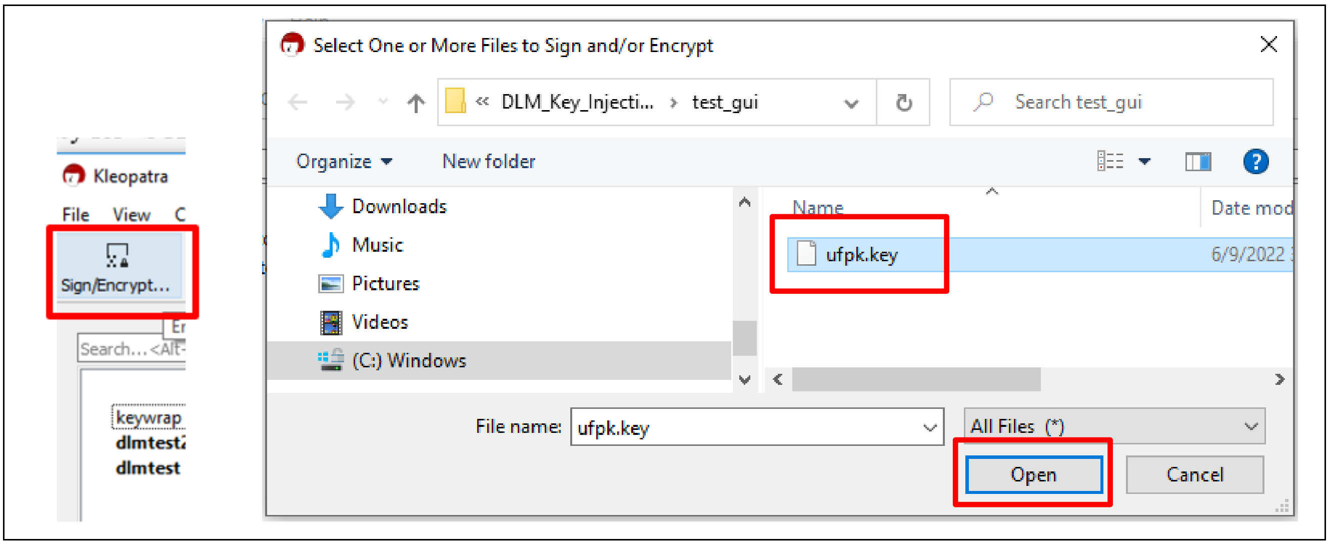
Send the generated UFPK key file to the Renesas Key Wrap service https://dlm.Renesas.com/keywrap to obtain the wrapped UFPK (W-UFPK).

```
UFPK: 000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F
Output File: C:\DLM_Key_Injectioin\test_gui\ufpk.key

OPERATION SUCCESSFUL
```

Figure 35. Create a Fixed UFPK Using the GUI

4.4.3 Encrypt UFPK with Renesas PGP Public Key

Select **Sign/Encrypt** from Kleopatra and select to encrypt the `ufpk.key` key file with the Renesas PGP public key.

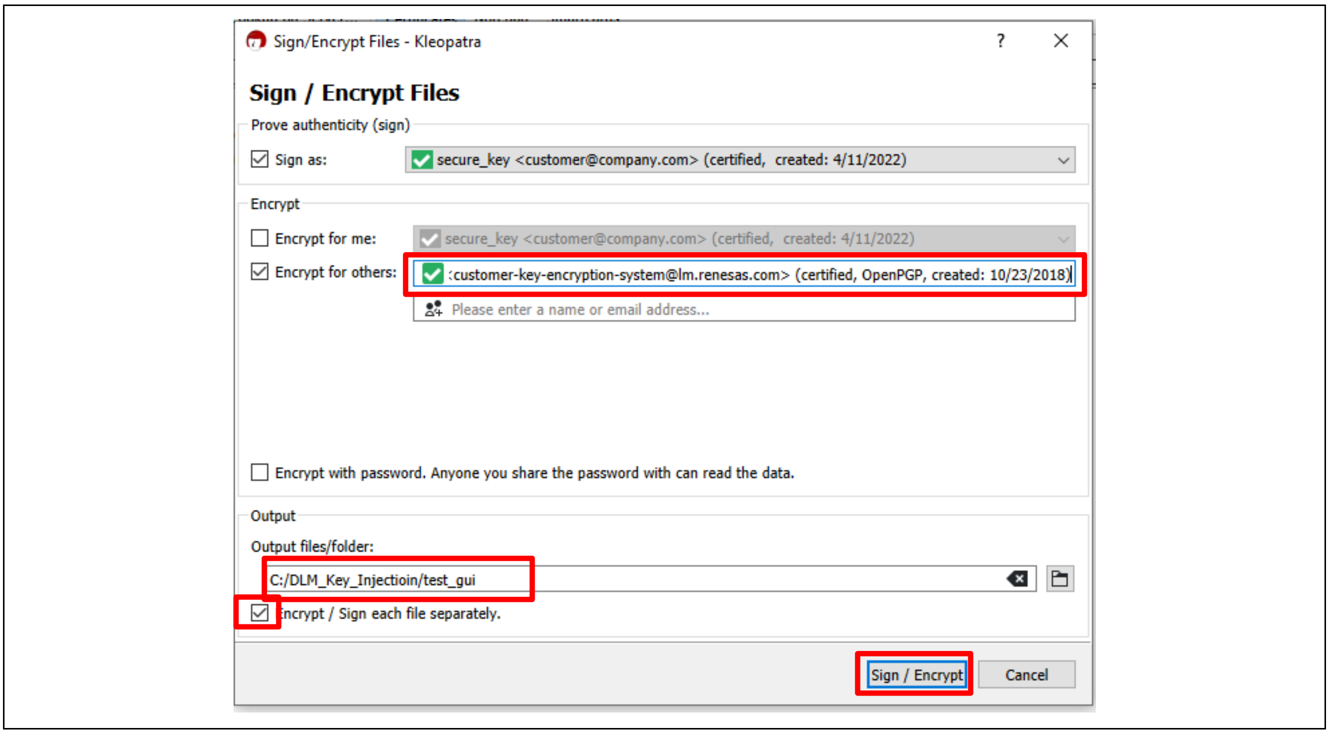


Select One or More Files to Sign and/or Encrypt

File name: ufpk.key

Figure 36. 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: secure_key <customer@company.com> (certified, created: 4/11/2022)

Encrypt

☐ Encrypt for me: secure_key <customer@company.com> (certified, created: 4/11/2022)

☑ Encrypt for others: :customer-key-encryption-system@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.

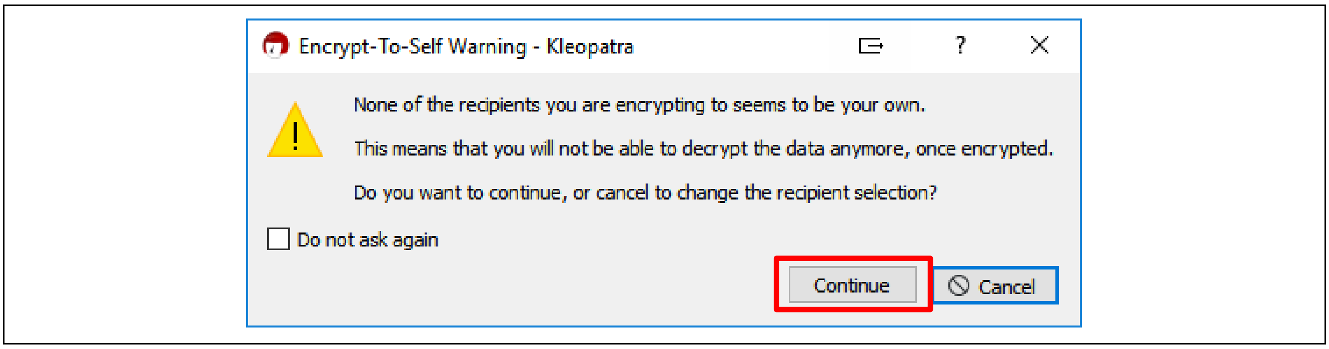
Output

Output files/folder: C:/DLM_Key_Injectioin/test_gui

☑ Encrypt / Sign each file separately.

Figure 37. Use Renesas Public Key to Encrypt UFPK

You may also want to select **Encrypt for me** so you have an encrypted version of the UFPK key file to archive. If you do not, you will get an Encrypt-To-Self Warning that you cannot decrypt the data. Press **Continue**.



Encrypt-To-Self Warning - Kleopatra

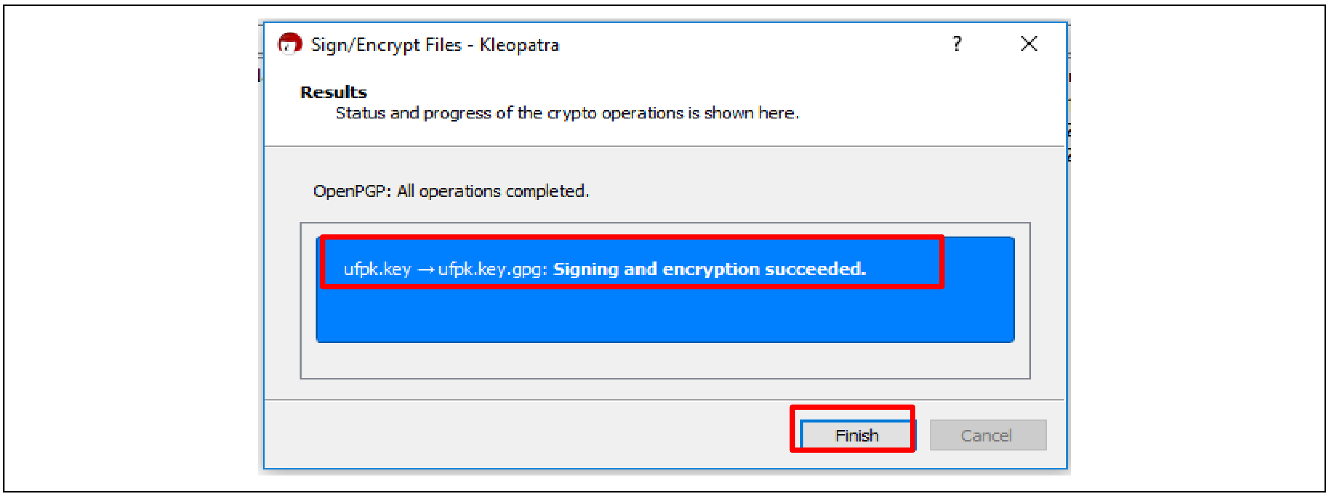
None of the recipients you are encrypting to seems to be your own.

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?

Figure 38. Confirm Encryption Option

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



Sign/Encrypt Files - Kleopatra

Results
Status and progress of the crypto operations is shown here.

OpenPGP: All operations completed.

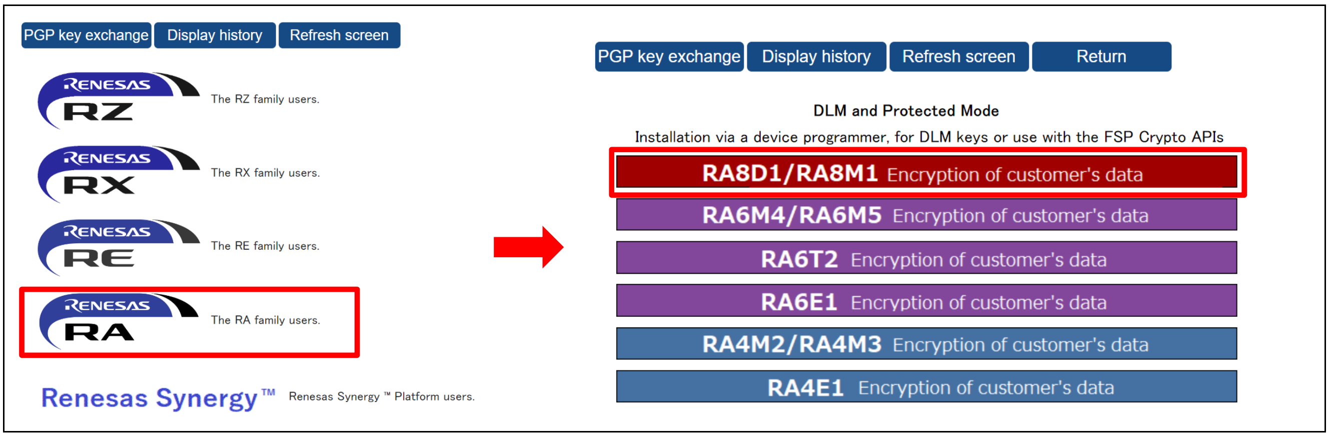
ufpk.key → ufpk.key.gpg: **Signing and encryption succeeded.**

Figure 39. Encrypt the UFPK Key with Renesas Public Key

4.4.4 Send UFPK to Renesas DLM Server

Now, we can send the encrypted UFPK to the Renesas DLM Server, which will be decrypted by the Renesas private key and used to generate the Wrapped UFPK (W-UFPK).

From the DLM Server user interface, select the RA Family series and choose **RA8D1/RA8M1 Encryption of customer's data** as shown below.



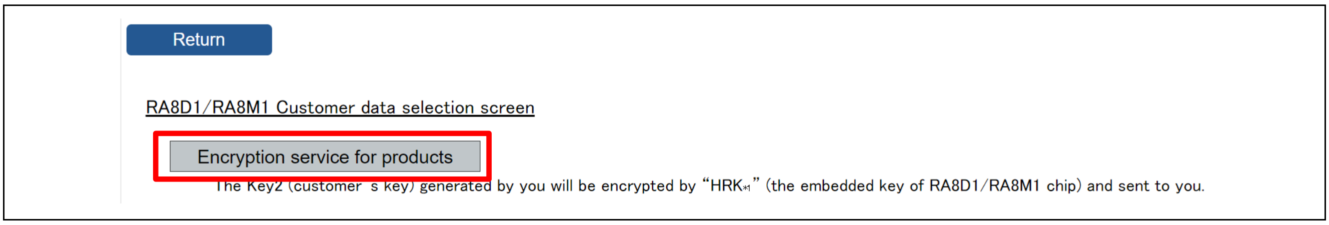
DLM and Protected Mode

Installation via a device programmer, for DLM keys or use with the FSP Crypto APIs

RA8D1/RA8M1 Encryption of customer's data
RA6M4/RA6M5 Encryption of customer's data
RA6T2 Encryption of customer's data
RA6E1 Encryption of customer's data
RA4M2/RA4M3 Encryption of customer's data
RA4E1 Encryption of customer's data

Figure 40. Select RA Device DLM and Protected Mode

At the next screen, select **Encryption service for products**

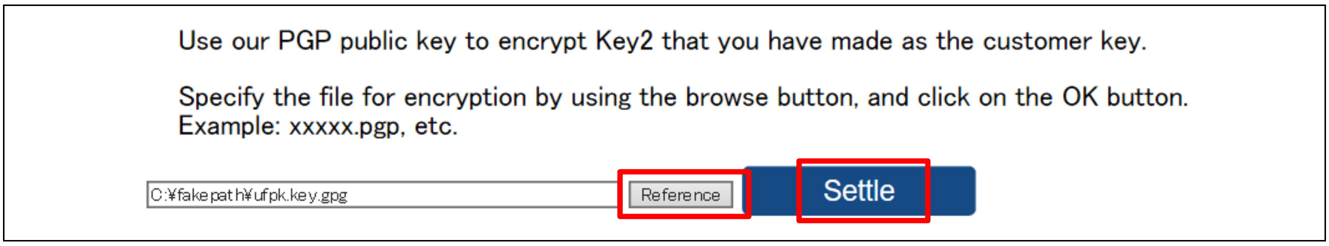


RA8D1/RA8M1 Customer data selection screen

The Key2 (customer s key) generated by you will be encrypted by "HRK※1" (the embedded key of RA8D1/RA8M1 chip) and sent to you.

Figure 41. Select Encryption service for products

Next, click **Reference** and select the `.gpg` file generated from Figure 39.



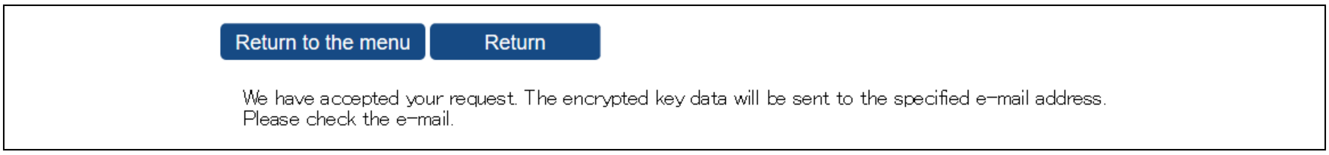
Use our PGP public key to encrypt Key2 that you have made as the customer key.

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 42. Send Encrypted UFPK to DLM Server

Click **Settle. T**he DLM server will display a message that the file has been received, and the W-UFPK will be sent to the registered email address.

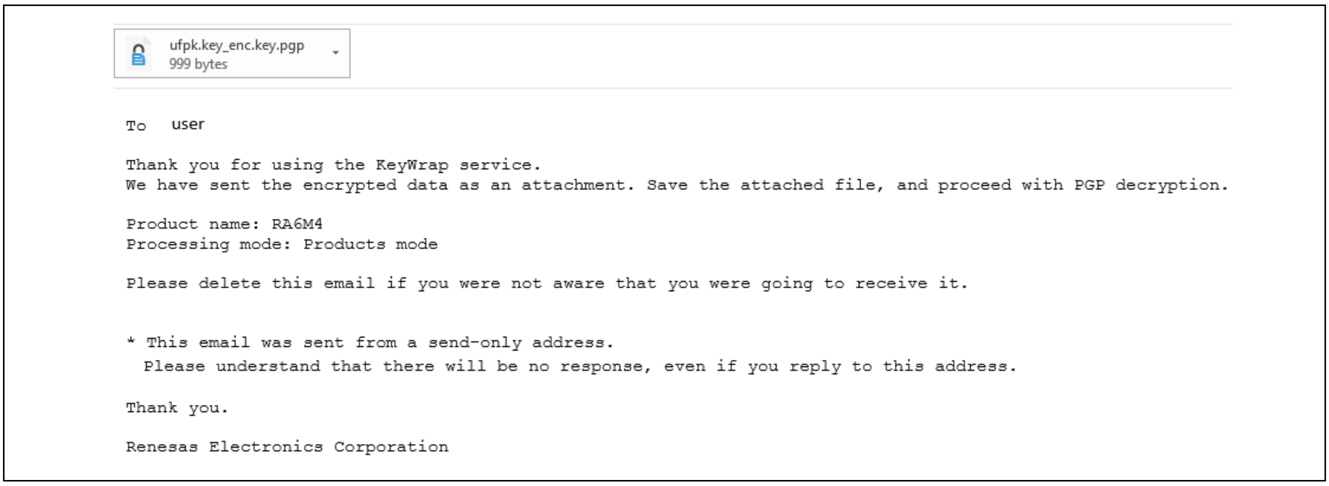


We have accepted your request. The encrypted key data will be sent to the specified e-mail address.
Please check the e-mail.

Figure 43. Message from DLM Sever

4.4.5 Receive the Encrypted W-UFPK Key

The W-UFPK, encrypted with your PGP public key, should arrive in your email within a few minutes.



```
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
```

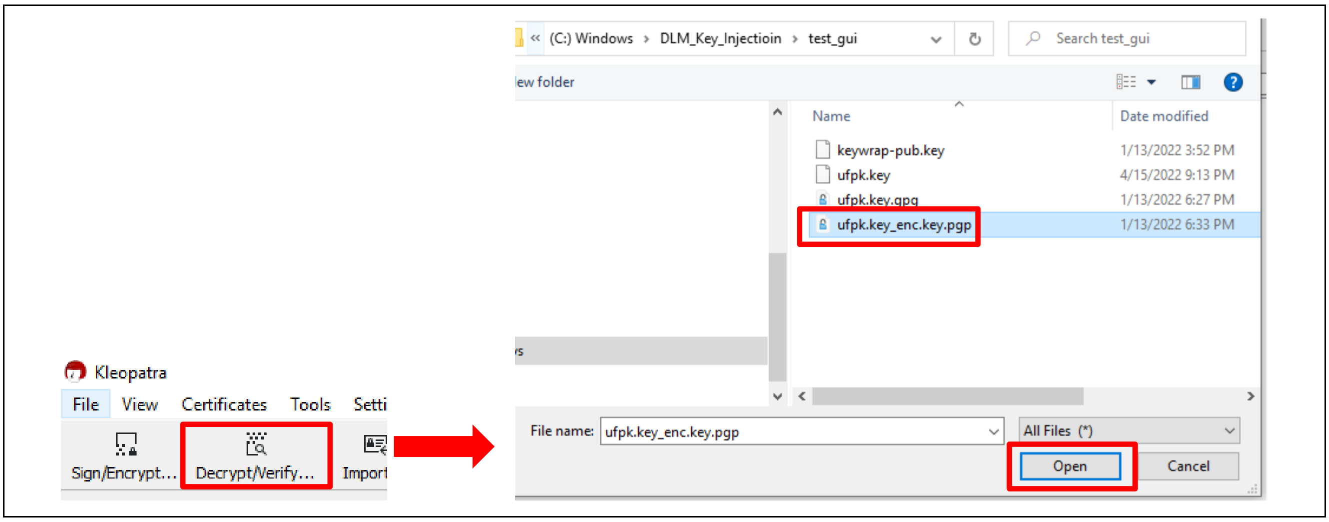
Figure 44. Receive the Wrapped DLM Key

Save this file for use in the next step.

4.4.6 Decrypt the Encrypted W-UFPK Key

The W-UFPK received from the email is encrypted with your public key. You must decrypt the W-UFPK key file using your private key.

In Kleopatra, click **Decrypt/Verify** and select the encrypted W-UFPK key file.



(C:) Windows › DLM_Key_Injectioin › test_gui

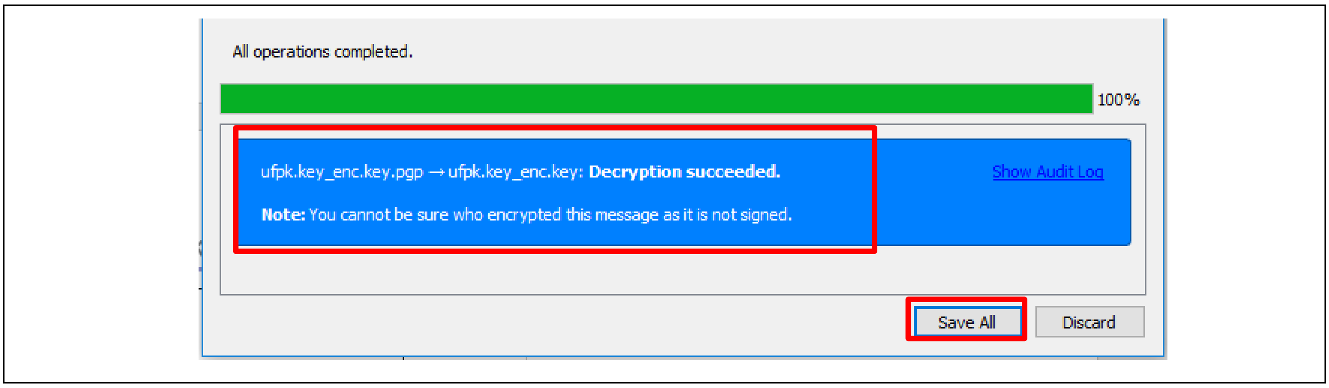
Name	Date modified
keywrap-pub.key	1/13/2022 3:52 PM
ufpk.key	4/15/2022 9:13 PM
ufpk.key.gpg	1/13/2022 6:27 PM
ufpk.key_enc.key.pgp	1/13/2022 6:33 PM

File name: ufpk.key_enc.key.pgp

All Files (*)

Figure 45. Encrypted W-UFPK Key File

If prompted, provide your PGP key passphrase. The decrypted file will be stored at the specified location.



All operations completed.

100%

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 46. Decrypt with PGP Private Key

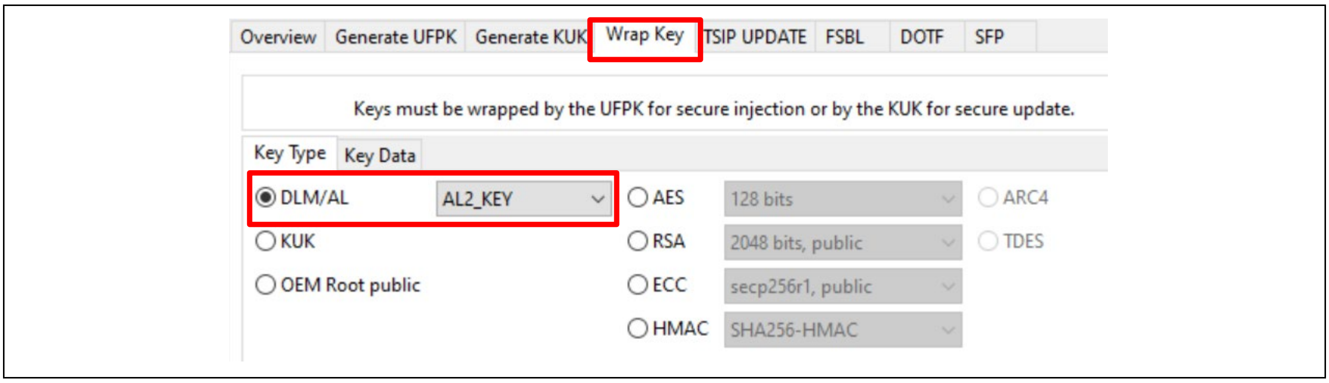
4.5 Create the Wrapped Authentication Keys using SKMT

To inject a DLM/AL key, the key needs to be wrapped by the UFPK, and both the wrapped DLM/AL Key and the W-UFPK need to be sent to the MCU via the serial programming interface. The SKMT can create an RFP key injection file, which includes the wrapped DLM/AL key and W-UFPK.

This section shows how to create key injection files for the AL2_KEY and AL1_KEY using the GUI interface.

4.5.1 Wrap an AL2_KEY

Launch the SKMT GUI and select the **Wrap Key** tab. Then open the **Key Type** tab and select **DLM/AL**. First, we will inject the AL2_KEY.



Keys must be wrapped by the UFPK for secure injection or by the KUK for secure update.

● DLM/AL AL2_KEY

○ KUK

○ OEM Root public

○ AES 128 bits

○ RSA 2048 bits, public

○ ECC secp256r1, public

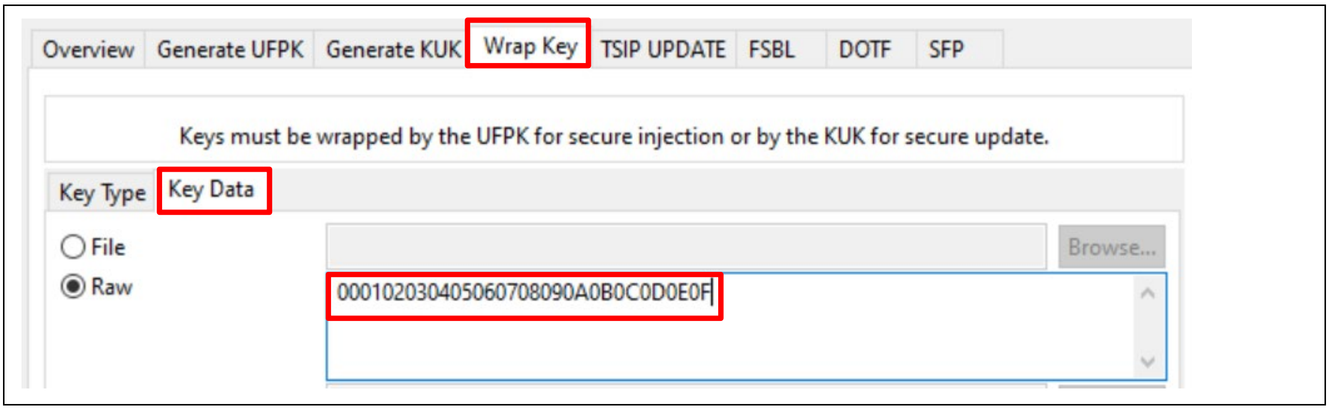
○ HMAC SHA256-HMAC

○ ARC4

○ TDES

Figure 47. Choose AL2_KEY as the Key Type

Select the **Key Data** tab. For this example, we will specify the key as raw data by selecting **Raw** and entering the key value (`000102030405060708090A0B0C0D0E0F`), but we could also specify a binary key file containing the key value.



Keys must be wrapped by the UFPK for secure injection or by the KUK for secure update.

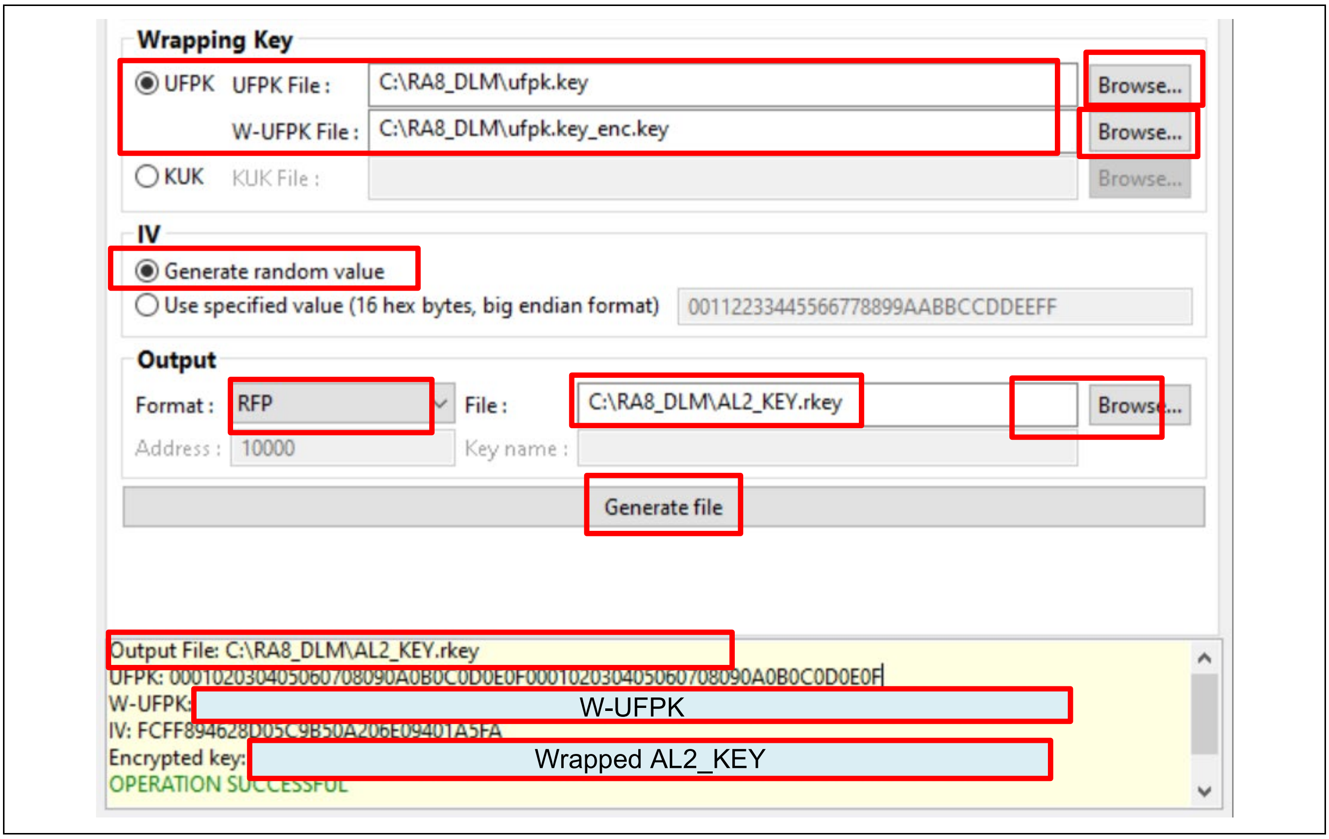
○ File

● Raw: 000102030405060708090A0B0C0D0E0F

Figure 48. Enter the AL2_KEY Data

In the **Wrapping Key** section, click the corresponding **Browse** buttons to select the **UFPK** and **W-UFPK** key pair created in the section 4.4.2 and 4.4.6. Choose **Generate random value** for the IV. In the **Output** section, select **RFP** and click the **Browse** button to enter the output file name.

Now click the **Generate File** button. The Renesas key file (`AL2_KEY.rkey)` will be generated.



Wrapping Key

● UFPK UFPK File : C:\RA8_DLM\ufpk.key

W-UFPK File : C:\RA8_DLM\ufpk.key_enc.key

○ KUK KUK File :

IV

● Generate random value

○ Use specified value (16 hex bytes, big endian format) 00112233445566778899AABBCCDDEEFF

Output

Format : RFP File : C:\RA8_DLM\AL2_KEY.rkey

Address : 10000 Key name :

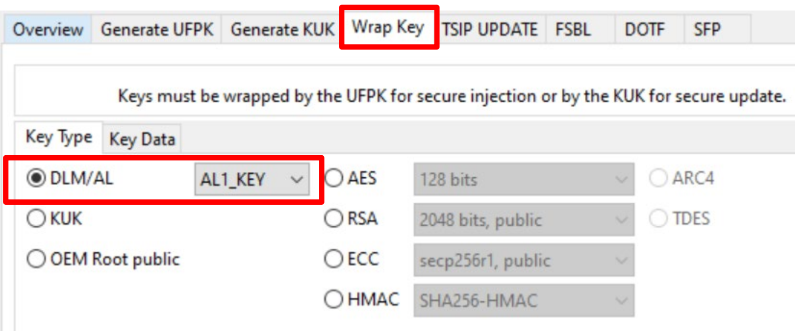
```
Output File: C:\RA8_DLM\AL2_KEY.rkey
UFPK: 000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F
W-UFPK: W-UFPK
IV: FCFF894628D05C9B50A206E09401A5FA
Encrypted key: Wrapped AL2_KEY
OPERATION SUCCESSFUL
```

Figure 49. Generate the RFP Injection File for the AL2_KEY

Note that the Wrapped UFPK is used in the AL2_KEY. When injecting the AL2_KEY to the MCU through RFP, the W-UFPK information is also included. However, the plaintext AL2_KEY and UFPK are NOT contained in the `*.rkey` file, enabling confidential transfer of the key injection file.

4.5.2 Wrap an AL1_KEY

The wrapping of the AL1_KEY is very similar to the wrapping of AL2_KEY. Select **AL1_KEY** in the **Key Type** tab.

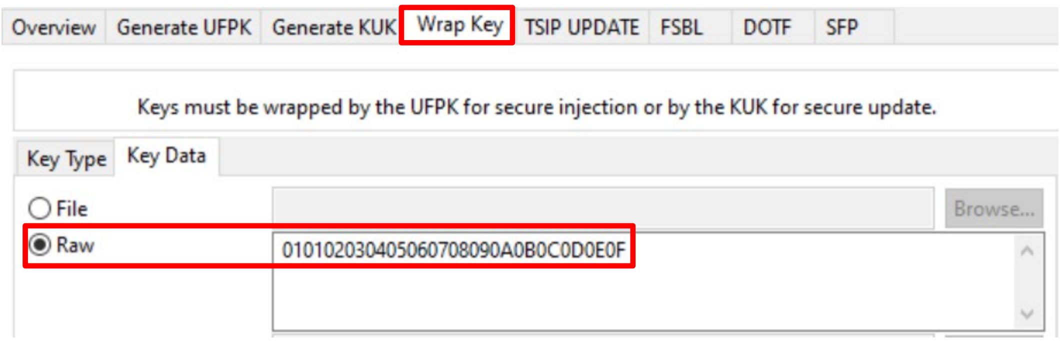


Keys must be wrapped by the UFPK for secure injection or by the KUK for secure update.

● DLM/AL AL1_KEY
○ KUK
○ OEM Root public
○ AES 128 bits
○ RSA 2048 bits, public
○ ECC secp256r1, public
○ HMAC SHA256-HMAC
○ ARC4
○ TDES

Figure 50. Choose AL1_KEY as the Key Type

Enter the data for the key value on the **Key Data** tab. Again, for this example, we will specify raw data for the key (`0101020304050607080 90A0B0C0D0E0F`).



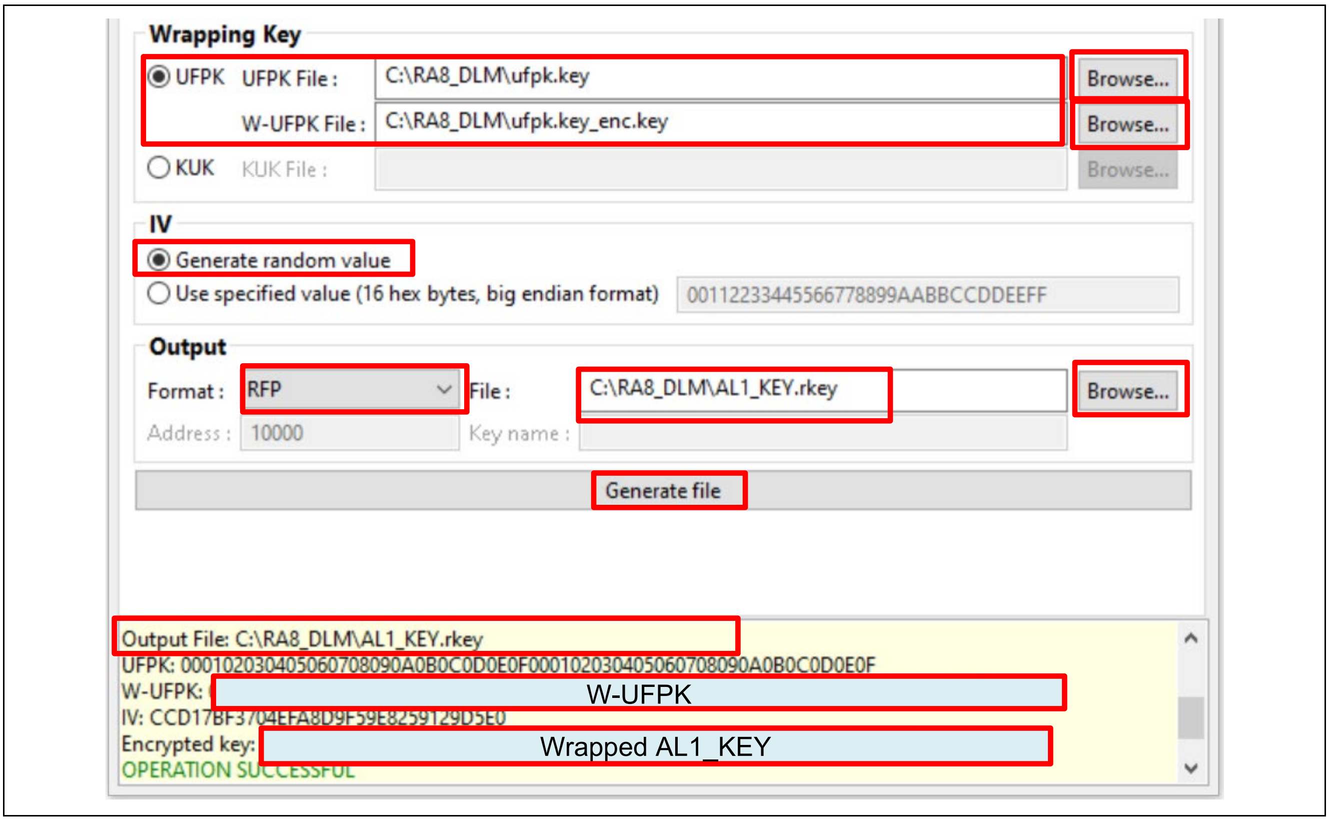
Keys must be wrapped by the UFPK for secure injection or by the KUK for secure update.

○ File
● Raw 0101020304050607080 90A0B0C0D0E0F

Figure 51. Enter the AL1_KEY Data

In the **Wrapping Key** section, click the corresponding **Browse** buttons to select the **UFPK** and **W-UFPK** key pair created in the section 4.4.2 and 4.4.6. Choose **Generate random value** for the IV. In the **Output** section, select **RFP** and click the **Browse** button to enter the output file name.

Now click the **Generate File** button. The Renesas key file (`AL1_KEY.rkey)` will be generated.



Wrapping Key

● UFPK UFPK File : C:\RA8_DLM\ufpk.key

W-UFPK File : C:\RA8_DLM\ufpk.key_enc.key

○ KUK KUK File :

IV

● Generate random value

○ Use specified value (16 hex bytes, big endian format) 00112233445566778899AABBCCDDEEFF

Output

Format : RFP File : C:\RA8_DLM\AL1_KEY.rkey

Address : 10000 Key name :

```
Output File: C:\RA8_DLM\AL1_KEY.rkey
UFPK: 000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F
W-UFPK:
IV: CCD17BF3704EFA8D9F59E8259129D5E0
Encrypted key:
OPERATION SUCCESSFUL
```

Figure 52. Generate the RFP Injection File for the AL1_KEY

Note that the Wrapped UFPK is used in the AL1_KEY. When injecting the AL1_KEY to the MCU through RFP, the W-UFPK information is also included. However, the plaintext AL1_KEY and UFPK are NOT contained in the `*.rkey` file, enabling confidential transfer of the key injection file.

4.6 Inject Authentication Keys using RFP

This section demonstrates how to perform DLM/AL Key injection. 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.

4.6.1 Inject an AL2_KEY

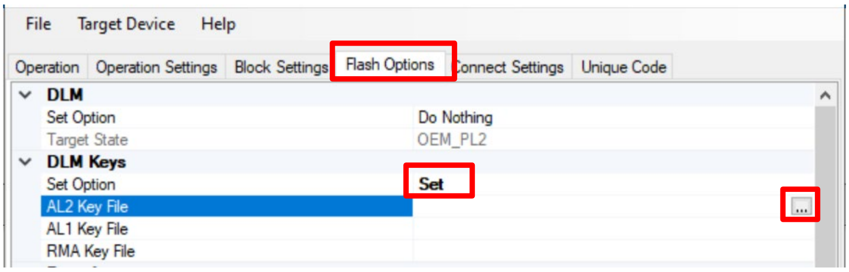
An Authentication Level 2 Key (AL2_KEY) and a Return Material Authorization Key (RMA_KEY) can be injected when the MCU is in an AL2 state.

Note: Authentication Key injection is not mandatory. It is also not mandatory to inject all possible Authentication Keys if only specific DLM and authenticated debug capabilities are required. Be sure to examine the DLM and authenticated debug states to determine which, if any, DLM and Authentication and Keys should be injected.

This example uses the AL2_KEY Renesas key file generated in the previous section as an example. The RMA_KEY can be injected simultaneously using the same general steps. Since the transition to RMA_REQ using the RMA_KEY cannot be regressed, it is not demonstrated here.

Unzip `ra8m1_dlm_key_inject_rfp.zip` to reveal `ra8m1_dlm_key_inject.rpj`. Launch RFP and open the RFP project `ra8m1_dlm_key_inject.rpj` and configure the settings explained in this section.

On the **Flash Options** tab, under **DLM Keys**, the option **Set Option** is set to **Set**. Click on AL2_KEY.rkey**,** and then click the "**…**" to the right to select the corresponding AL2_KEY key file.

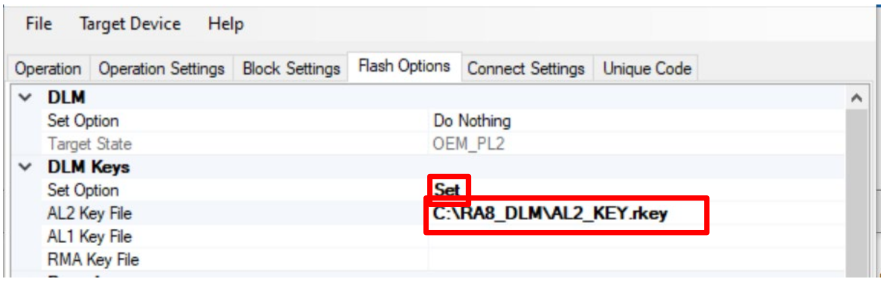


DLM
Set Option: Do Nothing
Target State: OEM_PL2

DLM Keys
Set Option: Set
AL2 Key File
AL1 Key File
RMA Key File

Figure 53. Select the AL2_KEY to Inject

Select the `AL2_KEY.rkey` generated earlier.

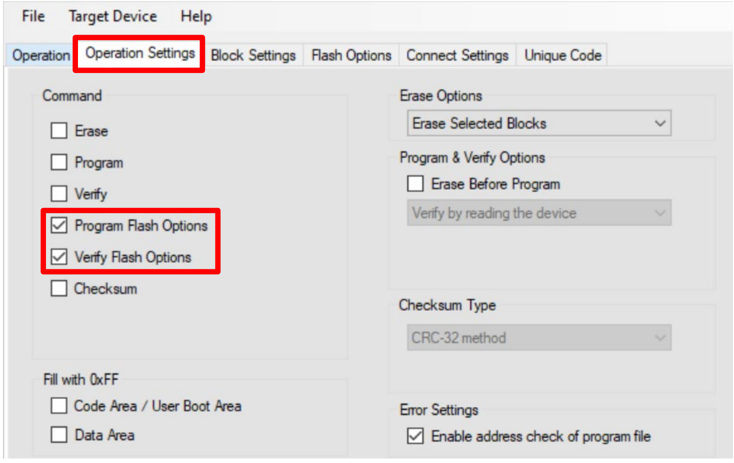


DLM
Set Option: Do Nothing
Target State: OEM_PL2

DLM Keys
Set Option: Set
AL2 Key File: C:\RA8_DLM\AL2_KEY.rkey
AL1 Key File
RMA Key File

Figure 54. Select the AL2_KEY to Inject

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



Command
☐ Erase
☐ Program
☐ Verify
☑ Program Flash Options
☑ Verify Flash Options
☐ Checksum

Fill with 0xFF
☐ Code Area / User Boot Area
☐ Data Area

Erase Options
Erase Selected Blocks

Program & Verify Options
☐ Erase Before Program
Verify by reading the device

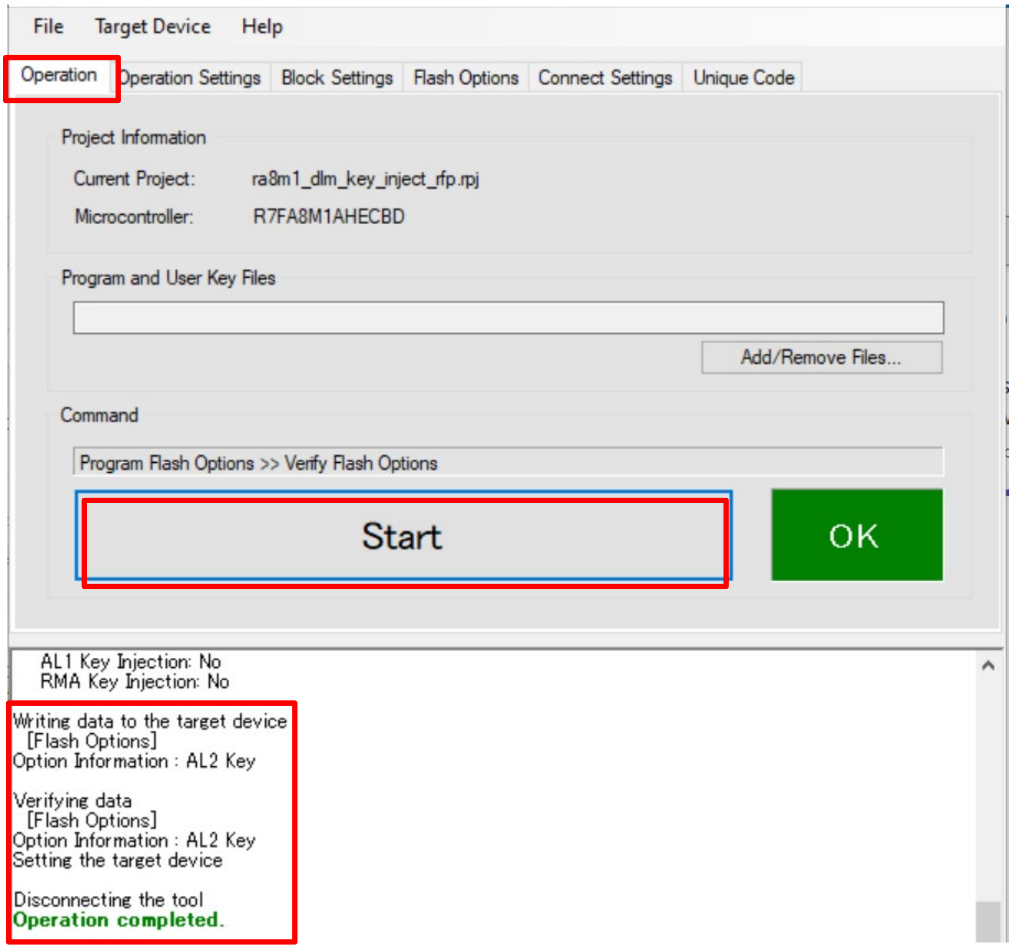
Checksum Type
CRC-32 method

Error Settings
☑ Enable address check of program file

Figure 55. Select Program and Verify Flash Option Setting

Note that DLM/AL keys and program code can be programmed at the same time or separately.

Navigate to the **Operation** tab, select the secure application binary (if desired), and click **Start** to program the data.



Current Project: ra8m1_dlm_key_inject_rfp.rpj

Microcontroller: R7FA8M1AHECBD

Program Flash Options >> Verify Flash Options

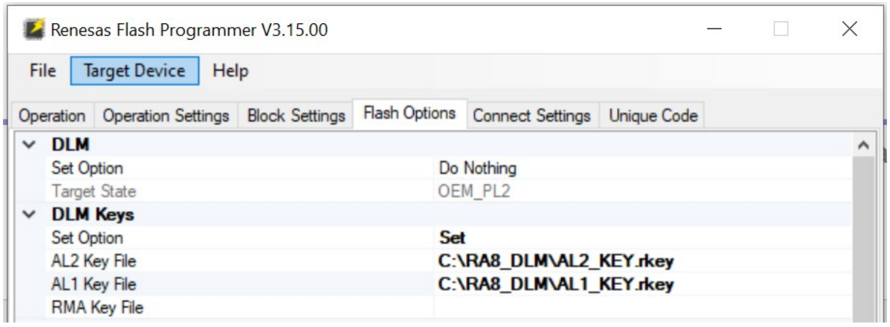
```
 AL1 Key Injection: No
 RMA Key Injection: No

Writing data to the target device
  [Flash Options]
Option Information : AL2 Key

Verifying data
  [Flash Options]
Option Information : AL2 Key
Setting the target device

Disconnecting the tool
Operation completed.
```

Figure 56. Inject the AL2_KEY

4.6.2 Inject an AL1_KEY

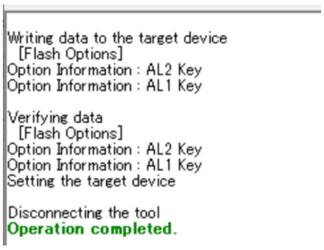
The AL1_KEY can be injected in OEM_PL2 or OEM_PL1. It can be injected with or without an injected AL2_KEY.

When injecting the AL1_KEY in OEM_PL2, it is possible to inject AL1_KEY alone or inject both AL1_KEY and AL2_KEY in the **Flash Options** setting, as shown in Figure 57.



DLM	
Set Option	Do Nothing
Target State	OEM_PL2
DLM Keys	
Set Option	**Set**
AL2 Key File	**C:\RA8_DLM\AL2_KEY.rkey**
AL1 Key File	**C:\RA8_DLM\AL1_KEY.rkey**
RMA Key File	

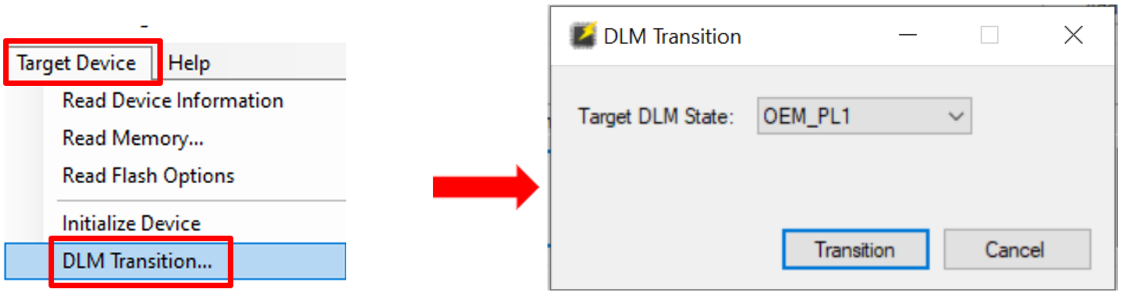
Figure 57. Include the AL1_KEY and AL2_KEY for Key Injection



```
Writing data to the target device
  [Flash Options]
Option Information : AL2 Key
Option Information : AL1 Key

Verifying data
  [Flash Options]
Option Information : AL2 Key
Option Information : AL1 Key
Setting the target device

Disconnecting the tool
Operation completed.
```

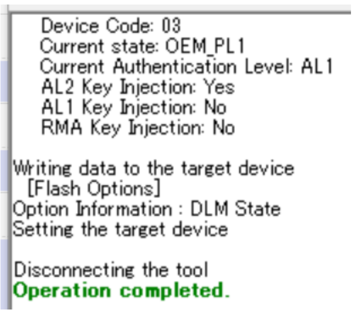
Figure 58. Successful Injection of AL1_KEY and AL2_KEY

The next example shows how the AL1_KEY can be injected in the OEM_PL1 state. In this case, the transition to OEM_PL1 following Figure 59 first and then inject the AL1_KEY alone, as shown in Figure 61.



Target DLM State: OEM_PL1

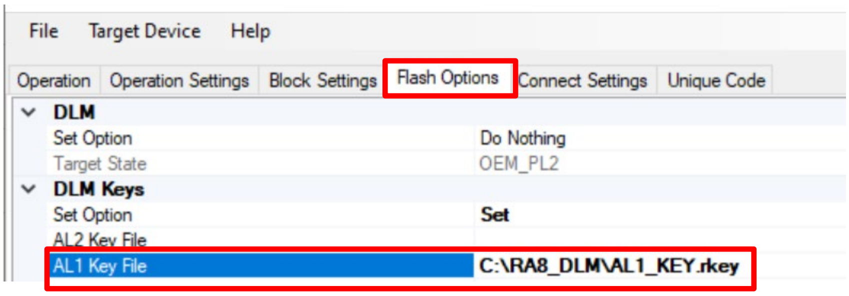
Figure 59. Transition the MCU Device Lifecycle State to AL1



```
  Device Code: 03
  Current state: OEM_PL1
  Current Authentication Level: AL1
  AL2 Key Injection: Yes
  AL1 Key Injection: No
  RMA Key Injection: No

Writing data to the target device
  [Flash Options]
Option Information : DLM State
Setting the target device

Disconnecting the tool
Operation completed.
```

Figure 60. Successful Transition to OEM_PL1

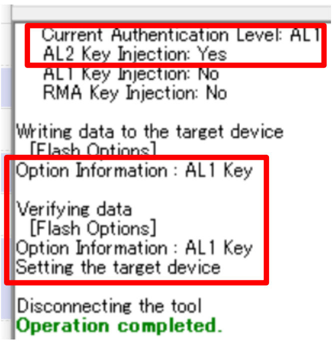
Add the AL1_KEY key file entry as shown below.



DLM
Set Option Do Nothing
Target State OEM_PL2
DLM Keys
Set Option Set
AL2 Key File
AL1 Key File C:\RA8_DLM\AL1_KEY.rkey

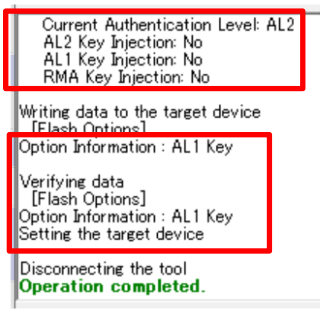
Figure 61. Select the AL1_KEY to Inject

Navigate to the **Operation** tab and click **Start** to program the data.



```
Current Authentication Level: AL1
AL2 Key Injection: Yes
AL1 Key Injection: No
RMA Key Injection: No

Writing data to the target device
  [Flash Options]
Option Information : AL1 Key

Verifying data
  [Flash Options]
Option Information : AL1 Key
Setting the target device

Disconnecting the tool
Operation completed.
```

Figure 62. AL1_KEY injected with Existing AL2_KEY in OEM_PL1



```
Current Authentication Level: AL2
AL2 Key Injection: No
AL1 Key Injection: No
RMA Key Injection: No

Writing data to the target device
  [Flash Options]
Option Information : AL1 Key

Verifying data
  [Flash Options]
Option Information : AL1 Key
Setting the target device

Disconnecting the tool
Operation completed.
```

Figure 63. AL1_KEY injected without Existing AL2_KEY in OEM_PL1

5. Regress the Authentication and Protection Levels

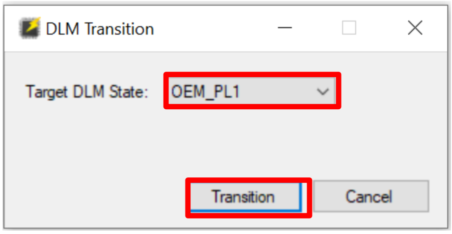
This section explains how to do authenticated state regression using RFP and RDPM.

5.1 Authenticated Transition using RFP

As explained earlier, if the DLM/PL state is OEM_PL0 and an AL1_Key is injected, it is possible to regress the Protection Level to OEM_PL1. This can be achieved by providing the AL1_Key.

The following are some example steps of performing authenticated transitions.

1. Initialize the MCU to OEM_PL2.
2. Inject AL2_KEY and AL1_KEY.
3. Possible state regression sequences:
   - Case 1: transition from OEM_PL0 to OEM_PL1 using AL1_KEY. Refer to step 4 for the example operation for this use case.
   - Case 2: transition from OEM_PL1 to OEM_PL2 using AL2_KEY. Refer to step 5 for the example operation for this use case.
   - Case 2: transition from OEM_PL0 to OEM_PL2 using AL2_KEY. The example operation in step 5 also applies to this use case.
4. Transition from OEM_PL0 to OEM_PL1 using the AL1_KEY.

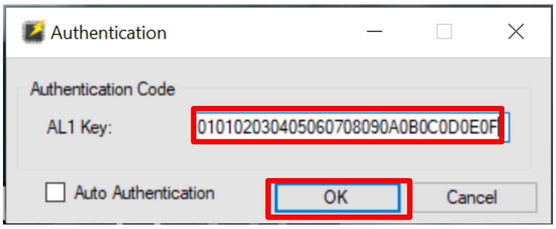


DLM Transition

Target DLM State: OEM_PL1

Figure 64. Transition from OEM_PL0 to OEM_PL1

Provide the AL1_KEY plaintext data to transition to the OEM_PL1 state. If desired, read the device information out to confirm the resulting state after the transition is finished.



Authentication

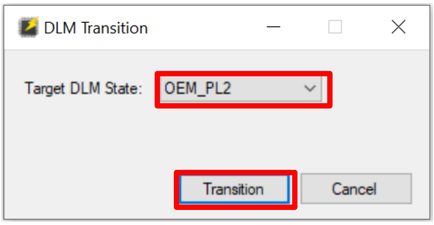
Authentication Code

AL1 Key: 01010203040506070809OA0B0C0D0E0F

□ Auto Authentication

Figure 65. Provide the AL1_KEY to Transition to OEM_PL1

5. With the MCU in OEM_PL1 or OEM_PL0 and AL2_KEY injected, it is possible to transition from OEM_PL1 or OEM_PL0 state to OEM_PL2 state using AL2_KEY.

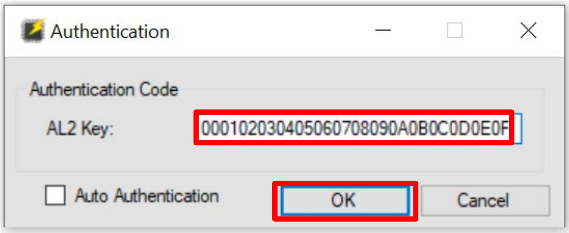


DLM Transition

Target DLM State: OEM_PL2

Figure 66. Transition from OEM_PL1 to OEM_PL2

Provide the AL2_KEY plaintext data to transition to OEM_PL2. If desired, read the device information out to confirm the resulting state after the transition is finished.



Authentication

Authentication Code

AL2 Key: 000102030405060708090A0B0C0D0E0F

□ Auto Authentication

Figure 67. Provide the AL2_KEY to Transition to OEM_PL2

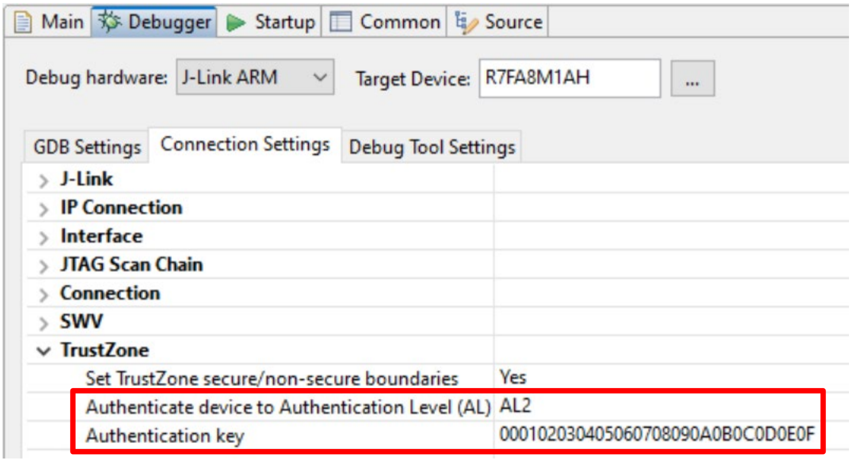
5.2 Authenticated Transition using e^2 studio

When using e^2 studio, it is not possible to change the Protection Level. However, it is possible to perform authenticated debugging by changing the Authentication Level for the current debug session.

Authenticated Transition from AL1 to AL2

1. Initialize the RA8M1 to OEM_PL2.
2. Inject AL2_KEY and AL1_KEY to RA8M1 using RFP.

3. Develop secure and non-secure applications using the combined development model with default debug configuration. In this configuration, when we debug the non-secure application, both the secure and non-secure applications will be programmed, and the TrustZone boundary will be set up.
4. Program the secure application and non-secure application and set up the TrustZone boundary by debugging the non-secure application or using RFP.
5. Change the Device Lifecycle to OEM_PL1 using RFP or RDPM.
6. Update the non-secure application Debug Configuration:
   - Disable the TrustZone setup boundary setup
   - Transition to AL2 for debugging purposes by providing the AL2_KEY plaintext data.



Debug hardware: J-Link ARM Target Device: R7FA8M1AH

Set TrustZone secure/non-secure boundaries	Yes
Authenticate device to Authentication Level (AL)	AL2
Authentication key	000102030405060708090A0B0C0D0E0F

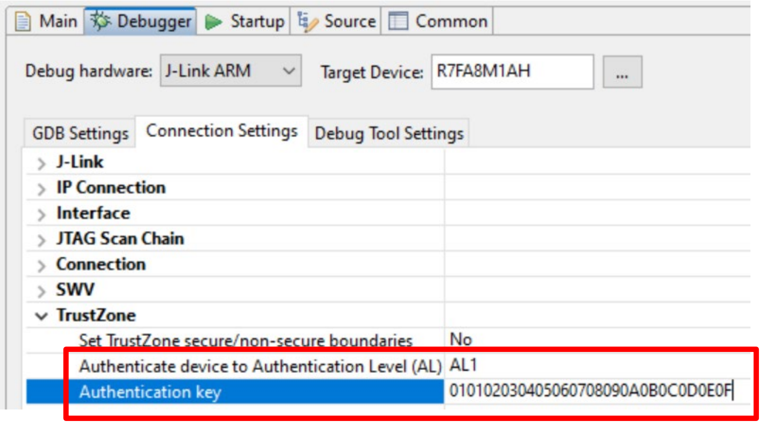
Figure 68. Debug the Secure Application and Transition to AL2

7. Applying the new configuration and debugging the non-secure application should proceed smoothly. The system will authenticate to transition to AL2 and start to debug the secure application. Note that this transition is temporary - the system will revert to OEM_PL1 upon the next power cycle.

Authenticated Transition from AL0 to AL1

1. Initialize the RA8M1 to OEM_PL2.
2. Inject AL2_KEY and AL1_KEY to RA8M1.
3. Develop secure and non-secure applications using the combined development model with default debug configuration. In this configuration, when we debug the non-secure application, both the secure and non-secure applications will be programmed, and the TrustZone boundary will be set up.
4. Program the secure application and non-secure application and set up the TrustZone boundary by debugging the non-secure application or using RFP.
5. Change the Device Lifecycle to OEM_PL0 using RDPM or RFP. The Authentication Level will now be at AL0.
6. Update the non-secure application project Debug Configuration:
   - Download the non-secure application only
   - Disable the TrustZone setup boundary setup
   - Transition to AL1 for debugging purposes by providing the AL1_KEY plaintext data.

   Figure 69 is an example configuration using the example AL1_KEY used in this application note.



Debug hardware: J-Link ARM

Target Device: R7FA8M1AH

Set TrustZone secure/non-secure boundaries	No
Authenticate device to Authentication Level (AL)	AL1
Authentication key	0101020304050607080 90A0B0C0D0E0F

Figure 69. Authenticated Transition to AL1 During Debugging

7. Applying the new configuration and debugging the non-secure application should proceed smoothly. The system will authenticate to transition to AL1 and start to debug the non-secure application. Note that this transition is temporary - the system will revert to OEM_PL0 upon the next power cycle.

6. References

1. Flexible Software Package (FSP) User's Manual
2. Renesas RA8M1 Group User's Manual: Hardware
3. Renesas RA Family Installing and Utilizing the Device Lifecycle Management Keys (R11AN0469)

7. Website and Support

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

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	Sept.02.24	—	Initial release

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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

Trademarks

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

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/.