

RX Family

Device Lifecycle Management for RX74M Devices

Introduction

Renesas RX Family Device Lifecycle Management (DLM) is a security control framework that includes the management of device states specific to the RX MCU Family. DLM identifies the current phase (development, production or deployment) of the device and controls the capabilities of the debug function, the serial programming interface, and Renesas test mode.

This application note provides comprehensive guidelines for using the RX Family DLM system.

Required Resources

The following resources are referenced throughout this application note:

Development Tools and Software

- Renesas Flash Programmer (RFP) v3.21 or later
<https://www.renesas.com/us/en/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html>
- Renesas Security Key Management Tool (SKMT) v1.12 or later
<https://www.renesas.com/software-tool/security-key-management-tool>
- Gpg4win
<http://www.gpg4win.org/>

Hardware

- EK-RX74M, Evaluation Kit for RX74M MCU Group ([renesas.com/ek-rx74m](https://www.renesas.com/ek-rx74m))
- PC running Windows® 11 OS
- One USB device cable Type-C® male to Type-C® male

Prerequisites and Intended Audience

You need to use development tools such as the Renesas Flash Programmer (RFP) and the Security Key Management Tool (SKMT) for handling the operations described in this application note. In addition, the application note assumes that you have some knowledge of RX Family MCU security features. As required, refer to the chapter *Security Features* in the *Renesas RX74M Group User's Manual: Hardware* and read the SKMT manual while proceeding through 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 RX Family MCUs.

Contents

1. Introduction to Device Lifecycle Management for the RX74M Group.....	4
1.1 Overview.....	4
1.2 Terminology.....	5
1.3 Lifecycle States	6
1.4 Protection and Authentication Levels.....	7
1.4.1 Protection Level Transitions.....	7
1.4.2 Authentication Level Keys and Authentication Level Transitions.....	8
1.4.3 Usage Example of Protection and Authentication Levels	8
1.5 ALx_KEY and RMA_KEY	9
2. RX74M Device Lifecycle Management Use Cases	10
2.1 Cases with IP Protection	10
2.1.1 Development – Programming the Valuable IP and Preparing for External Provision of the Device...	12
2.1.2 Development – Application.....	12
2.1.3 Production – End Product	13
2.2 Cases without IP Protection	14
2.2.1 Case with the Use of Protect Zone	14
2.2.2 Case without the Use of Protect Zone	17
2.3 Renesas RMA	19
3. ALx_KEY and RMA_KEY Generation and Injection	20
3.1 Tools Used in ALx_KEY Injection Process	20
3.2 ALx_KEY Injection Overview.....	20
3.3 Establish PGP-Encrypted Communication with the Renesas Key Wrapping Service	21
3.3.1 Overview of the Renesas Key Wrapping Service	21
3.3.2 Create a PGP Key Pair	22
3.3.3 Registration with the Renesas Key Wrapping Service.....	25
3.3.4 PGP Public Key Exchange.....	26
3.3.5 Import Renesas PGP Public Key into Kleopatra.....	29
3.4 Create the UFPK and W-UFPK.....	30
3.4.1 Introduction to Renesas Security Key Management Tool.....	30
3.4.2 Create a UFPK File	30
3.4.3 Encrypt UFPK with Renesas PGP Public Key	34
3.4.4 Send UFPK to the Renesas Key Wrapping Service	36
3.4.5 Receive the Encrypted W-UFPK.....	37
3.4.6 Decrypt the Encrypted W-UFPK	38
3.5 Create Wrapped ALx_KEY Using SKMT	39
3.5.1 Wrap AL2_KEY	39
3.5.2 Wrap AL1_KEY	40
3.6 Inject Authentication Keys Using the RFP	42

3.6.1	Inject AL2_KEY	42
3.6.2	Inject AL1_KEY	44
4.	Transitions of the Authentication and Protection Levels.....	46
4.1	State Transitions without Authentication	46
4.1.1	State Transitions without Authentication Using the RFP	46
4.1.2	State Transitions without Authentication Using the Renesas Device Partition Manager (RDPM).....	47
4.2	Authenticated Transitions	50
4.2.1	Authenticated Transitions Using the RFP	50
4.2.2	Authenticated Transition Using e ² studio	52
5.	References	54
6.	Website and Support	55

1. Introduction to Device Lifecycle Management for the RX74M Group

1.1 Overview

The Renesas RX74M 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, product development, product deployment, field updates, and potential end-of-product failure analysis.

The following two elements control the RX74M DLM.

- Lifecycle state: The MCU state corresponding to the product phase
- Protection level (PL2, PL1, and PL0) and authentication level (AL2, AL1, and AL0): These levels determine effective functions for debugging, serial programming, and the Renesas test mode.

1.2 Terminology

Table 1-1 describes some terms used in this application note.

Table 1-1 Terminology

Term	Description	Section with Details
Protection level	A DLM state stored in non-volatile RAM (NVRAM). Transition to a lower-privileged PL is always allowed. Transition to a higher-privileged PL requires reflection of the authentication level or running of the Initialize command. Privileges are in descending order, that is, PL2, PL1, then PL0.	1.4.1
OEM_PLx	Notation indicating both the OEM state in the device lifecycle and the current protection level. In the OEM state, allowable operations change depending on the protection level.	2
Authentication level	A DLM state stored in RAM. Transition to a lower-privileged AL is always allowed. Transition to a higher-privileged AL requires authentication by using ALx_KEY. Privileges are in descending order, that is, AL2, AL1, then AL0.	1.4.2
ALx_KEY	Key used for transitions to higher-privileged authentication levels. The key must be injected at the authentication level before the transition to a lower-privileged authentication level.	1.5
Initialize command	Command sent to the device via the serial programming interface or the debugger. Running the Initialize command erases memory in the device and shifts the DLM state to OEM and the protection level to PL2. This command can be issued by using the Renesas Flash Programmer (RFP) or Renesas Device Partition Manager (RDPM).	1.4.1
RMA	Return Material Authorization Operation of requesting failure analysis of a device to Renesas. The analysis may erase user applications in the device.	1.5 and 2.3
Renesas test mode	Mode of analyzing a faulty device at Renesas. This mode is available in the state where an RMA has been authenticated.	2.3
RMA_KEY	Key used for transition to the Renesas test mode. The key must be injected at the AL2 authentication level.	2.3
Renesas key wrapping service	Key management service provided by Renesas. This is used for wrapping the user factory programming key (UFPK).	3

1.3 Lifecycle States

Table 1-2 is a summary of the available lifecycle states and the various capabilities in each state.

Table 1-2 Device Lifecycle State Definitions and Capabilities in Each State

Lifecycle	Definition	Protection Level	Debug Function	Serial Programming	Renesas Test Mode
OEM	Original Equipment Manufacturer	PL2, PL1, or PL0	Depends on the authentication level (AL).		Not available
LCK_BOOT	LoCKed BOOT interface (Boot interface locked) The debug interface and the serial programming interface are permanently disabled.	PL0	Not available	Not available	Not available
RMA_REQ	Return Material Authorization REQuest (RMA request) The customer must send the device to Renesas in this state. A transition from OEM to RMA_REQ requires RMA_KEY and AL2_KEY.	PL0	Not available	Available Access to code MRAM area and data MRAM area not possible	Not available
RMA_ACK	Return Material Authorization ACKnowledged (RMA acknowledged) Failure analysis in Renesas	PL2	Non-secure and secure debug functions are enabled.	Available Access to code MRAM area and data MRAM area not possible	Available
RMA_RET	Return Material Authorization RETurn (RMA return) This device has been returned to the customer. The device does not boot.	PL0	Not available	Not available	Not available

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

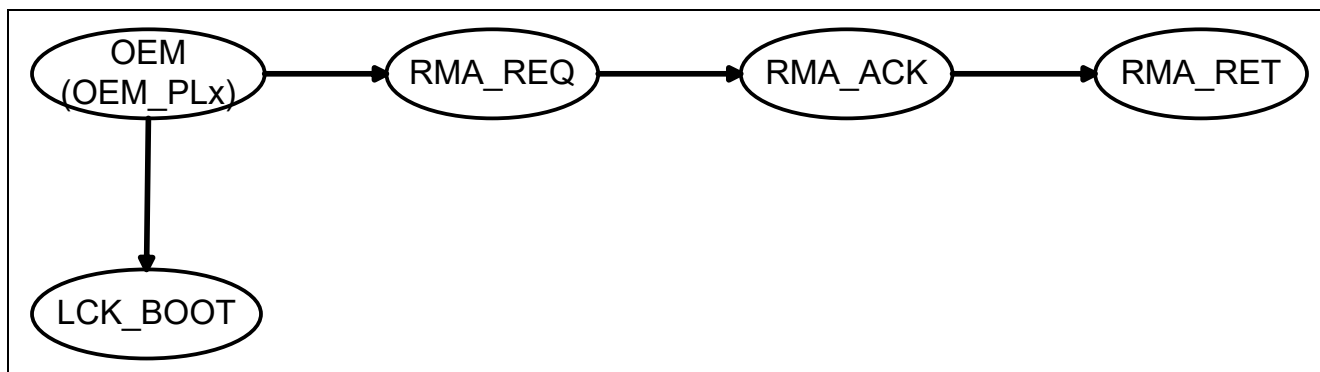


Figure 1-1 RX74M Device Lifecycles

In a transition to RMA_ACK, the contents of MRAM, except permanently block-protected memory or registers, are erased. Renesas can read the contents of the permanently block-protected memory or registers during failure analysis. Transition to RMA_REQ is not possible if AL2_KEY is disabled.

1.4 Protection and Authentication Levels

The protection level (PL) and the authentication level (AL) determine the availability of the debug function and the serial programming interface. PL and AL are fixed except in the OEM state. In the OEM state, three levels can be set for PL and AL.

AL indicates a temporary authentication status after the debugger or serial programmer has been connected. PL determines the initial value of AL after an MCU power-on reset.

PL and AL can be changed via the serial programming interfaces (UART, USB, FINE, SWD, and JTAG) in a device as shipped in the OEM state.

Table 1-3 Protection and Authentication Levels

Protection/Authentication Level	Debug, Read/Erase/Program via Serial Programming
PL2/AL2	Secure and non-secure
PL1/AL1	Non-secure only
PL0/AL0	None

1.4.1 Protection Level Transitions

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

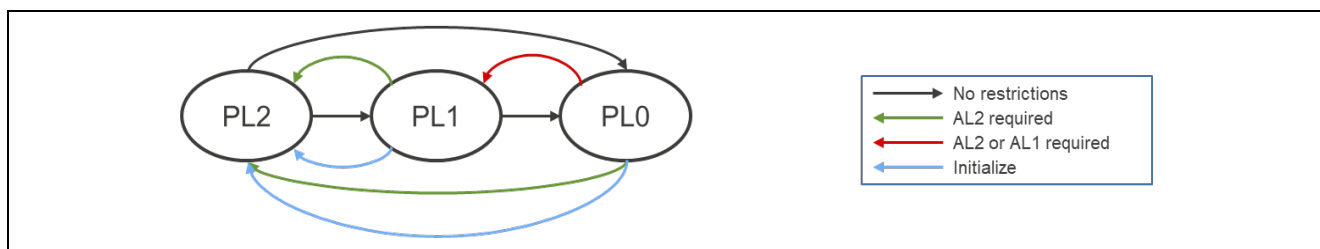


Figure 1-2 Protection Level Transitions

1.4.2 Authentication Level Keys and Authentication Level Transitions

Transition to a lower-privileged (lower-numbered) AL is always permitted with no need for authentication. Transition to a higher-privileged (higher-numbered) AL requires authentication by using the appropriate authentication level key. The boot firmware only supports changing the authentication level from a lower privilege to a higher privilege.

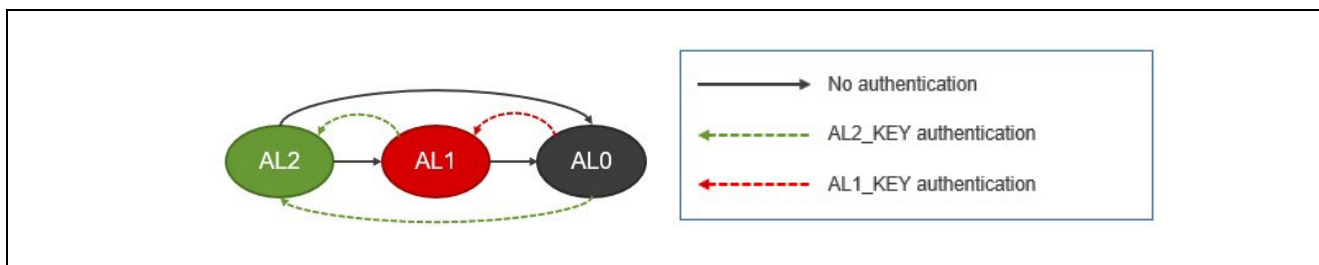


Figure 1-3 Authentication Level Transitions

As shown in Figure 1-1, 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 along with the indicated protection level.

1.4.3 Usage Example of Protection and Authentication Levels

Figure 1-4 shows a usage example of protection and authentication levels.

In the OEM state, the PL state is reflected in the AL state following a power-on reset. Transitions of the AL state are possible in the way described in sections 1.4.1 and 1.4.2 and are reflected in the PL.

When IP protection is required in the development phase, the phases can be divided by using the AL2 and AL1 states. The device is developed in the AL2 state during development phase 1. When the device is passed to development phase 2, access rights in development phase 2 can be managed by changing the state to AL1 as shown in Figure 1-4. Section 2.1 will describe details of this use case. If you will be using the end product in the OEM state, we recommend the use of PL0.

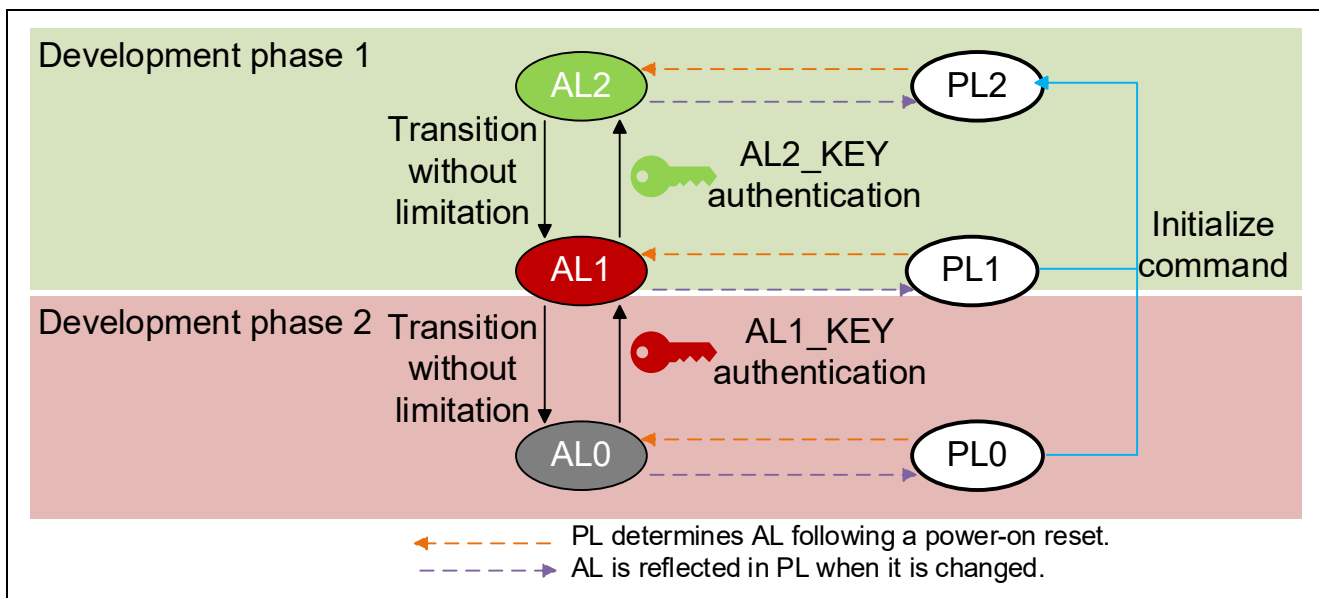


Figure 1-4 Usage Example of Protection and Authentication Levels (in the OEM State)

1.5 ALx_KEY and RMA_KEY

An authentication level key (AL key) or RMA_KEY is used for a transition between authentication levels or device lifecycle states. These keys are written to the MCU by injecting a secure key at the specific authentication level. The challenge and response authentication is used for protection against bugging or replay attacks. These keys are injected into the MRAM area which has not been mapped for the MCU. The following describes the main keys.

- **AL1_KEY:** The authentication level 1 key (AL1_KEY) can be injected in the OEM state at AL2 or AL1. This key is used in the OEM state during a transition from AL0 to AL1. The boot firmware parameter setting command can invalidate AL1_KEY in the OEM state at AL2 or AL1.
- **AL2_KEY:** The authentication level 2 key (AL2_KEY) can be injected in the OEM state at AL2. This key is used in the OEM state during a transition from AL0 or AL1 to AL2. The boot firmware parameter setting command can invalidate AL2_KEY in the OEM state at AL2.
- **RMA_KEY:** The return material authorization key (RMA_KEY) can be injected in the OEM state at AL2. This key is used when you send the MCU to Renesas for failure analysis through the process for approval to return a product. If AL2_KEY is not invalidated, this key can be used for a transition to the RMA_REQ state from any OEM state.

2. RX74M Device Lifecycle Management Use Cases

This section discusses some of the use cases for Device Lifecycle Management for RX74M MCUs with Protect Zone support.

Appropriately handling assets to be protected (e.g. valuable IP) during development requires the defense of designs by multiple layers of protection through combinations of “separating regions” and “defending against access”. With Protect Zone, valuable assets are allocated to the secure region and other applications are separately allocated to the non-secure region, implementing system-level isolation.

On the other hand, DLM involves managing external access routes such as debugging and serial programming to avoid illicit operations or unintended access.

In this way, the combination of forms of protection with different layers of “separating the execution environments” through Protect Zone and “managing access routes” through DLM enables robust security in design that does not depend on a single measure.

This section introduces use cases based on those design concepts.

2.1 Cases with IP Protection

The IP protection use case consists of valuable IP provided in a pre-programmed MCU to an external developer, who then develops the end application. In this development flow, valuable IP created by an IP developer must be protected to enable development of the application by the external developer. Specifically, the following conditions must be satisfied.

- Valuable IP should not be readable through an external interface such as a debugger or a programmer.
- An external developer can develop and debug applications.
- After the development has been completed, the device is retrieved so that the IP developer can restore the valuable IP for debugging again.

In the RX74M DLM, those requirements are satisfied by controlling the state transitions of PLs, authentication keys (AL1_KEY and AL2_KEY), and the Initialize command. That is, lifecycle management is implemented by protecting IP to enable development during the outsourcing of development and by restoring IP to be reusable after the device has been retrieved. Figure 2-1 shows the flow of development in this use case and an overview of the state transitions in DLM.

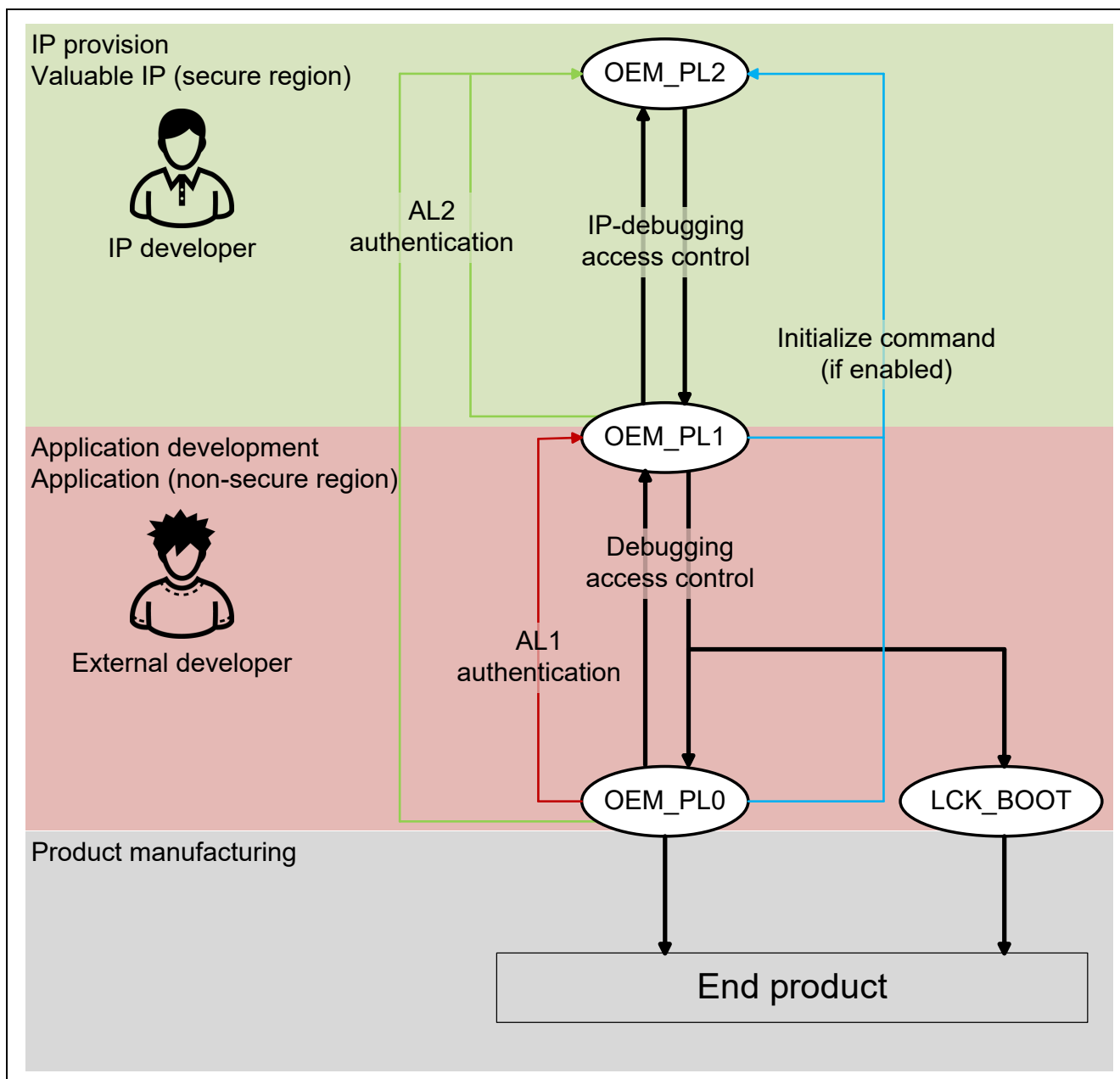


Figure 2-1 Flow of Development Including IP Protection and State Transitions in DLM

This figure shows that the valuable IP programmed by the IP developer is retained in the secure region and the device is provided to the external developer in that state to allow development of the application. In addition, it also shows that the valuable IP can be restored to availability for debugging through state transitions after the device has been retrieved.

2.1.1 Development – Programming the Valuable IP and Preparing for External Provision of the Device

The IP developer places the MCU in the OEM_PL2 state and programs the valuable IP in a secure region. After that, the IP developer places the MCU in the OEM_PL1 state and sets the access control conditions (PL and key settings) of the device before providing the device containing the given IP to the external developer. In this process, the following two conditions must be satisfied.

- The valuable IP should be protected before the device is provided to the external developer.
- The degree of operability required for the application development by the external developer should be ensured.

At this point, the control of external access to the device under DLM prevents reading or altering the valuable IP. On the other hand, recoverability of the IP and constraints on methods of debugging change with the settings, so the settings must be selected to suit the purpose of usage.

Table 2-1 indicates typical DLM settings, IP protection, and possibilities of transitions of the PL and initialization of the device when a device is provided to an external developer.

Table 2-1 DLM Settings, IP Protection, and Possibilities of Transitions of the PL and Initialization of the Device when a Device is Provided to an External Developer

DLM/PL State	Initialize Command	Injected ALx_KEY	Operation (IP Protection, Transition, and Initialization)
OEM_PL1	Enabled	None	Valuable IP is protected. The Initialize command can initialize the device. Transition to PL2 is not possible because no authentication key is present (the impossibility of authentication provides a constraint against the transition).
OEM_PL1	Enabled	AL2_KEY	Valuable IP is protected. The Initialize command can initialize the device. Transition to PL2 is possible after AL2 authentication.
OEM_PL1	Disabled	None	Valuable IP is protected. The Initialize command cannot initialize the device. Transition to PL2 is not possible because no authentication key is present (the impossibility of authentication provides a constraint against the transition).
OEM_PL1	Disabled	AL2_KEY	Valuable IP is protected. The Initialize command cannot initialize the device. Transition to PL2 is possible after AL2 authentication.

Each setting has different characteristics in terms of the risk of IP erasure, recoverability of the IP, and restrictions on debugging. Select the setting for usage by considering the balances of these factors.

2.1.2 Development – Application

The external developer will receive the device in the OEM_PL1 state and can then develop and debug application code. The valuable IP was placed in the secure region and is thus protected. The external developer proceeds with development under the following conditions.

- The valuable IP cannot be read or reprogrammed.
- Applications in the non-secure region can be developed in the normal way.

The use of AL1_KEY is effective during evaluation or debugging under conditions close to the product state. This allows transitions of the DLM state while retaining the valuable IP. If AL1_KEY is not used, the following restrictions apply.

- If the Initialize command is used, the region that contains the valuable IP may be erased.
- If the Initialize command is not used, transitions of the DLM state will be restricted and full evaluation will not be possible.

For those reasons, a transition to a DLM state authenticated by AL1_KEY enables evaluation and debugging while retaining the valuable IP.

2.1.3 Production – End Product

When the device is shipped in an end product, place the device in the OEM_PL0 state and determine the DLM settings. The combination of the following elements determines the settings in the product as shipped.

- AL1_KEY: Controlling debugging in the field
- AL2_KEY: Controlling recovery by the IP developer
- Enabling or disabling of the Initialize command

Table 2-2 indicates typical DLM settings, IP protection, and possibilities of transitions of the PL and initialization of the device with the permitted access and authentication when the end product is shipped.

Table 2-2 Patterns for Setting the DLM in the End Product as Shipped with IP Protected

DLM/PL State	Initialize Command	Injected ALx_KEY	Operation (Access, Transition, and Initialization)
OEM_PL0	Disabled	AL2_KEY AL1_KEY	Limited access via the serial programming interface is available*. Transitions to PL1 and PL2 are respectively possible after AL1 and AL2 authentication.
OEM_PL0	Disabled	AL2_KEY	Limited access via the serial programming interface is available*. Transition to PL2 is possible after AL2 authentication.
OEM_PL0	Disabled	AL1_KEY	Limited access via the serial programming interface is available*. Transition to PL1 is possible after AL1 authentication.
OEM_PL0	Disabled	None	Limited data reading via the serial programming interface is available*. Transition to PL is not possible after authentication because no authentication key is present.
OEM_PL0	Enabled	AL2_KEY AL1_KEY	Limited access via the serial programming interface is available*. Transitions to PL are possible after AL authentication. The device can be initialized by the Initialize command.
OEM_PL0	Enabled	None	Limited access via the serial programming interface is available*. The device can be initialized by the Initialize command.
LCK_BOOT	Unavailable	None	No access. Transitions to PL are not possible.

Note: Limited information on the MCU such as the MCU's unique ID can be acquired.

These settings enable design allowing the possibility of debugging after shipment, maintainability of the device in the field, and security level to suit the usage.

2.2 Cases without IP Protection

This section describes the cases in which the whole application is developed by the developers in the same main party, which is not the same as cases where a device incorporating valuable IP is provided to an external developer as described in section 2.1. In such cases, external protection of the IP is not the main point; however, secret data may still require protection depending on the data or processing handled in the application. In an RX74M device, the RSIP-E50D (a security IP module which handles wrapped keys) can be used to implement secure boot and various security processing even in a configuration without Protect Zone. Thus, Protect Zone is not required for all types of secure usage.

A typical case where the use of Protect Zone might be considered is when software has to directly handle a plaintext private key, such as in a system that uses a plaintext key in RSIP-E50D compatibility mode. This section describes the following two cases.

- Case with the use of Protect Zone
- Case without the use of Protect Zone

2.2.1 Case with the Use of Protect Zone

This case assumes a situation where software handles plaintext secret data, such a case where the system uses a plaintext key in RSIP-E50D compatibility mode. In such cases, when a private key or security processing is assigned to the secure region of Protect Zone, it can be separated from other regions of the application. Doing so can control:

- improper reference from a debugger or a programmer; and
- unintended access due to bugs in the application

Note that this device can implement a wrapped-key-based security configuration by using the RSIP-E50D module. In such cases, the use of Protect Zone is not essential because the software does not directly handle the plaintext key.

Figure 2-2 shows the flow of development when the plaintext secret data are protected by using Protect Zone and an overview of state transitions in the DLM.

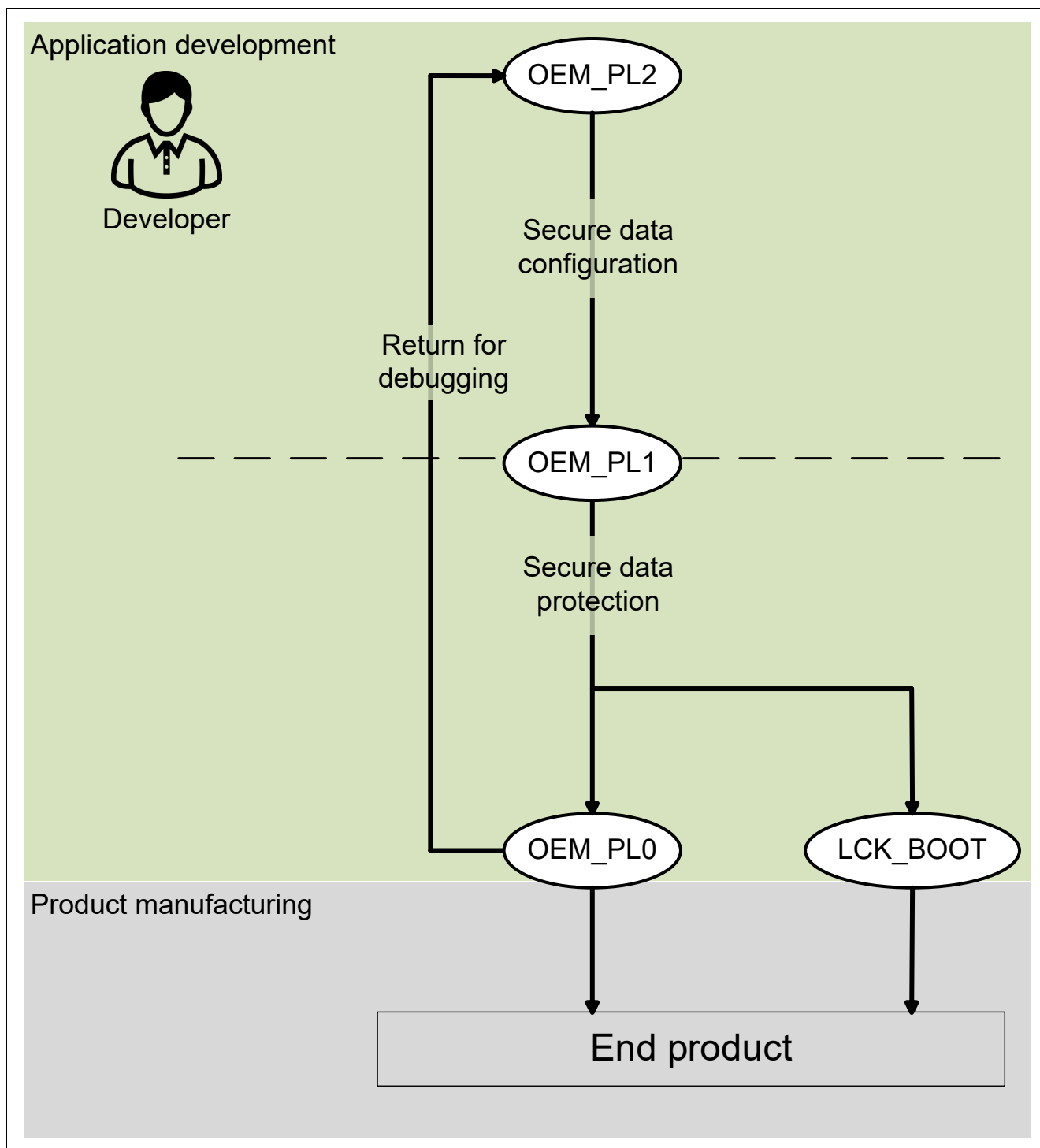


Figure 2-2 Case with the Use of Protect Zone without the Need for IP Protection

This figure shows the flow in a configuration where the plaintext private key or security processing is assigned to the secure region and separated from other regions by using Protect Zone. Although IP is not provided to an external developer because developers in the same main party are handling development, secret data configured in the secure region development are protected in the non-secure region development by Protect Zone. In addition, setting block protection in production enables a configuration that permanently prevents reprogramming of the secure region.

2.2.1.1 Development Phase

Development does not require separation from an external developer as described in section 2.1 because the whole application is developed by developers in the same main party. In development,

- processing for the handling of plaintext keys and
- security processing (e.g., root of trust)

are assigned in the OEM_PL2 state. After that, the application is assigned to a non-secure region to evaluate the system as a whole. This case does not require the use of AL keys during development.

Note: Do not set block protection for MRAM in the development phase. Doing so will disable the Initialize command, and reprogramming of the device will become impossible.

2.2.1.2 Production Phase

The mass production phase requires deciding the settings of DLM to protect the plaintext secret data which have been assigned to the secure region. The following items are the main considerations in selecting the settings.

- Is analysis or maintenance after shipment to be permitted?
- How strongly must the plaintext secret data be protected?

If strong protection is required, use block protection to prohibit reprogramming of the secure region. On the other hand, if analysis or maintenance is to be permitted, set AL1_KEY and AL2_KEY to enable authenticated transitions to the PL.

Table 2-3 DLM Settings and Possibilities of Debugging and Initializing the Device when Protect Zone is in Use

DLM/PL State	Initialize Command	Injected ALx_KEY	Operation (Access, Transition, and Initialization)
OEM_PL0	Disabled	AL2_KEY AL1_KEY	Limited access in boot mode is available*. Transitions to PL1 and PL2 are respectively possible after AL1 and AL2 authentication.
LCK_BOOT	Unavailable	None	No access. Transitions to PL are not possible.

Note: Limited information on the MCU such as the MCU's unique ID can be acquired.

When Protect Zone is in use, select the setting for usage by considering the balances of the strength of protection the plaintext secret data require and whether or not analysis or maintenance after shipment is to be permitted (convenience).

2.2.2 Case without the Use of Protect Zone

In this case, applications are developed in a flat configuration without the use of Protect Zone. In an RX74M device, the RSIP-E50D (a security IP module which handles wrapped keys) can be used to implement a configuration that does not require handling the plaintext private key. Such a configuration has an extensive range of usage, not only for systems which require security functions but also for general application development. For example, it is applicable to any of the following items.

- Secure boot with the use of a wrapped key
- Processing with the use of a key managed by a security IP
- General applications which do not require any separation of development

Figure 2-3 shows the configuration, flow of development, and an overview of the state transitions when Protect Zone is not in use.

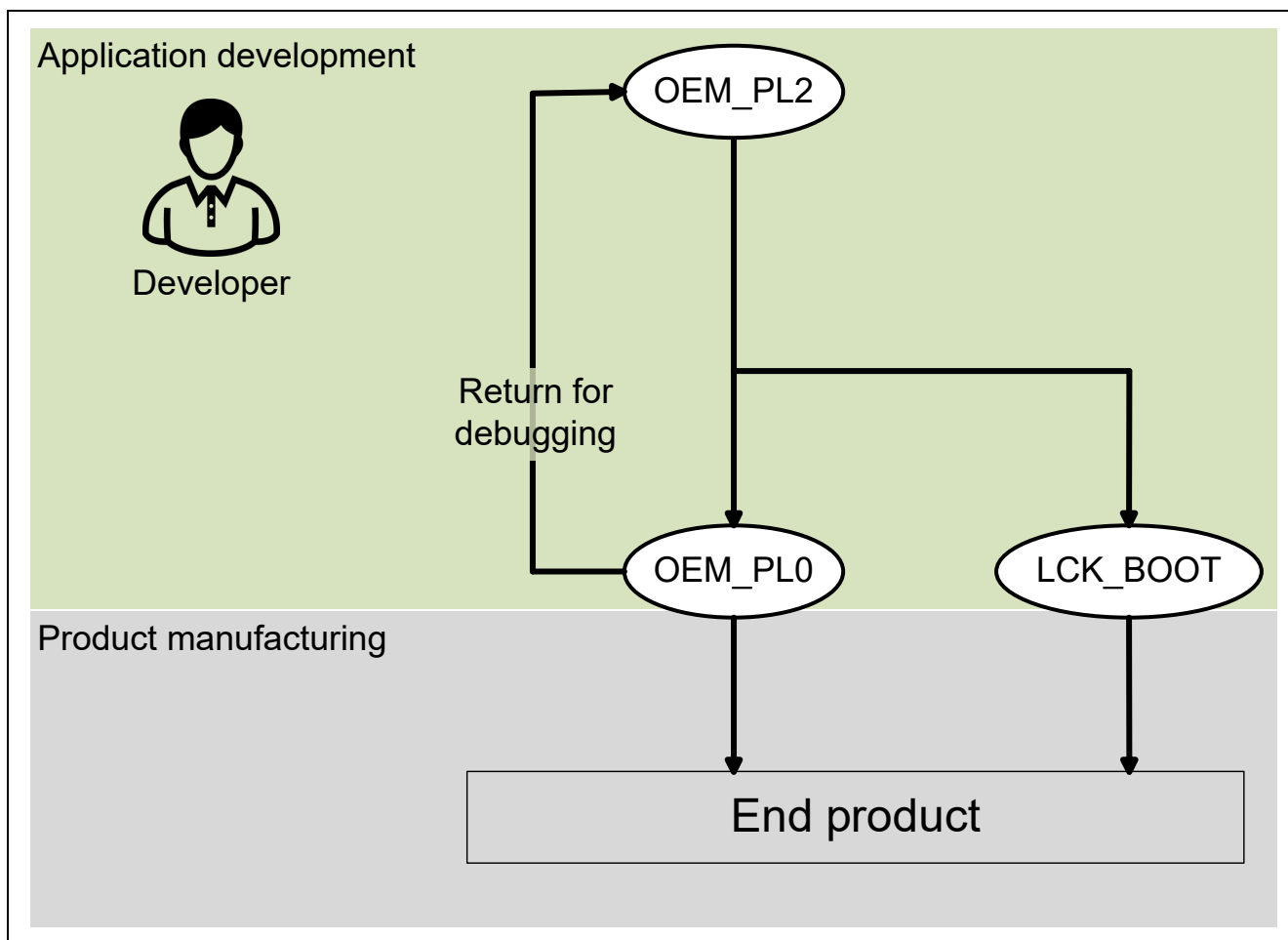


Figure 2-3 Case with No Use of Protect Zone without the Need for IP Protection

This figure shows a case where an application is developed in a flat configuration without the use of Protect Zone. Using a wrapped key with the RSIP-E50D simplifies the configuration by eliminating the need to handle the plaintext key in software. Thus, the developer can freely develop the product and the configuration depends on usage that does not require the specific separation of regions.

2.2.2.1 Development Phase

It is not necessary to use AL keys during development; 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.

Note: Do not set block protection for MRAM at the development phase. Doing so will disable the Initialize command, and reprogramming of the device will become impossible.

2.2.2.2 Production Phase

If the product will require analysis or reprogramming after shipment, set AL2_KEY. If this is not required, shipping the product in the PL0 or LCK_BOOT state will enable more restricted operation.

Table 2-4 DLM Settings, Access Control, PL Transitions, and Initialization of the Device

DLM/PL State	Initialize Command	Injected ALx_KEY	Available Functionality
OEM_PL0	Disabled	AL2_KEY	Limited access in boot mode is available*. Transition to PL2 is possible after AL2 authentication.
OEM_PL0	Enabled	AL2_KEY	Access restriction in boot mode is available*. Transition to PL2 is possible after AL2 authentication. The device can be initialized by the Initialize command.
OEM_PL0	Disabled	None	Limited access in boot mode is available*.
OEM_PL0	Enabled	None	Limited access in boot mode is available*. The device can be initialized by the Initialize command.
LCK_BOOT	Unavailable	None	No access.

Note: Limited information on the MCU such as the MCU's unique ID can be acquired.

2.3 Renesas RMA

To submit the MCU for failure analysis in Renesas through the RMA process, an RMA_KEY must be injected, and the device must be in the OEM state prior to the 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 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.

Figure 2-4 shows an overview of the RMA flow and transitions between DLM states when a failure was detected after product shipment.

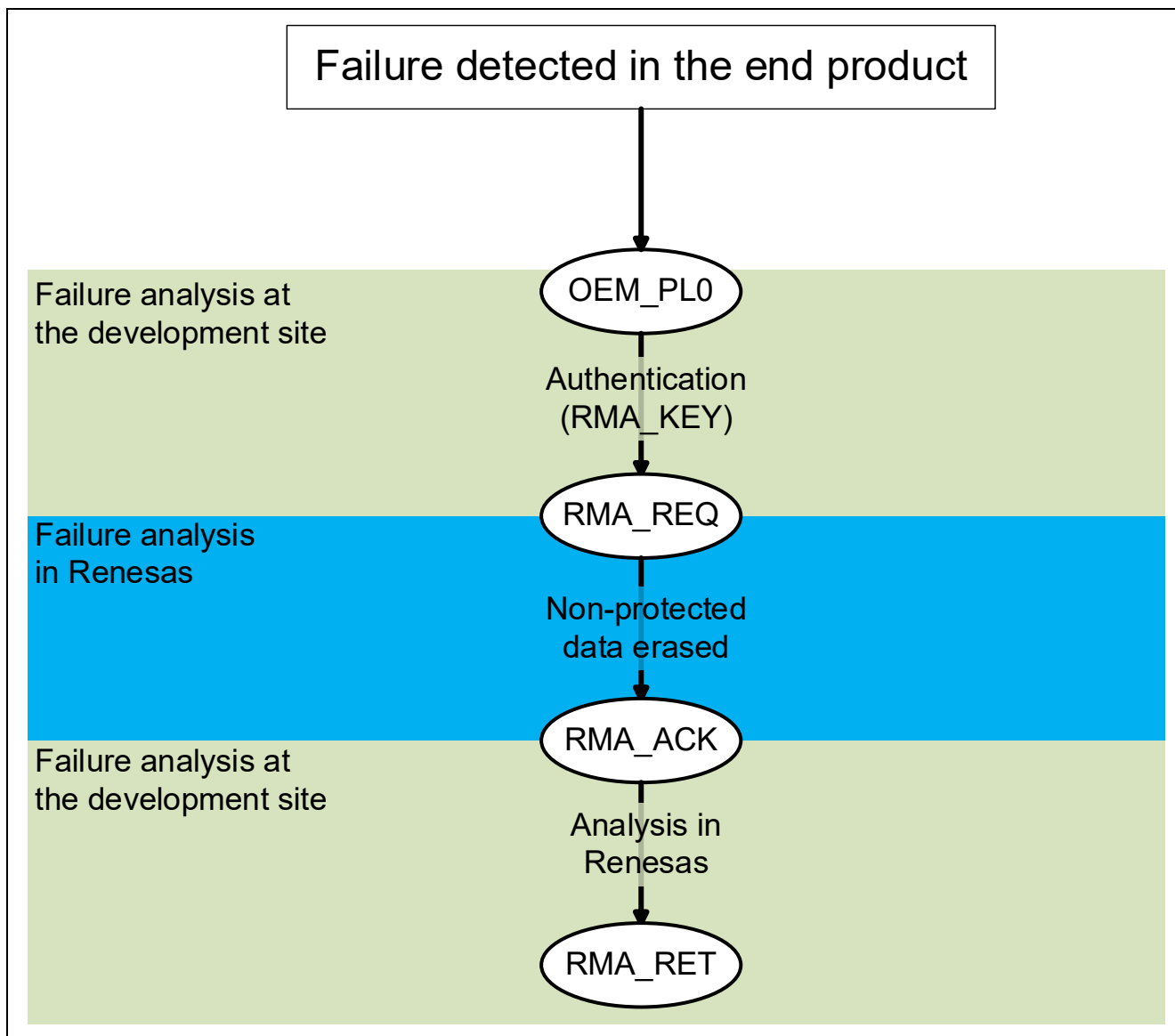


Figure 2-4 Renesas RMA

This figure shows the flow for placing the device in the RMA_REQ state by authentication with RMA_KEY when a failure was detected after product shipment. This transition will erase unprotected code and data and enable failure analysis in Renesas.

3. ALx_KEY and RMA_KEY Generation and Injection

This section guides the user in generating and injecting ALx_KEY into the MCU. Generating and injecting the RMA_KEY uses a similar procedure.

3.1 Tools Used in ALx_KEY Injection Process

Three tools are used in the process of generating and injecting ALx_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 wrapping service. 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 ALx_KEY injection.
- Renesas Flash Programmer (RFP) - This tool is used to inject ALx_KEY.

3.2 ALx_KEY Injection Overview

There are three high-level steps required to inject ALx_KEY into the MCU.

1. Create an arbitrary 256-bit user factory programming key (UFPK). The UFPK is used to encrypt ALx_KEY, to ensure that no plaintext keys are exposed during the injection process.
2. Obtain a wrapped UFPK (W-UFPK). The UFPK must be wrapped by the Renesas key wrapping service to obtain the W-UFPK.

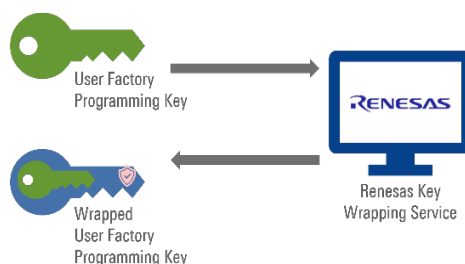


Figure 3-1 Wrap the UFPK

3. Wrap ALx_KEY using the UFPK.

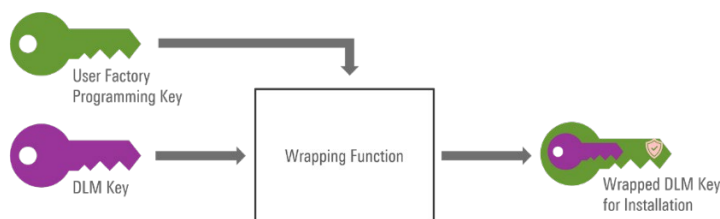


Figure 3-2 Wrap ALx_KEY Using the UFPK

4. Inject ALx_KEY using the serial programming interface by providing the W-UFPK and wrapped ALx_KEY.

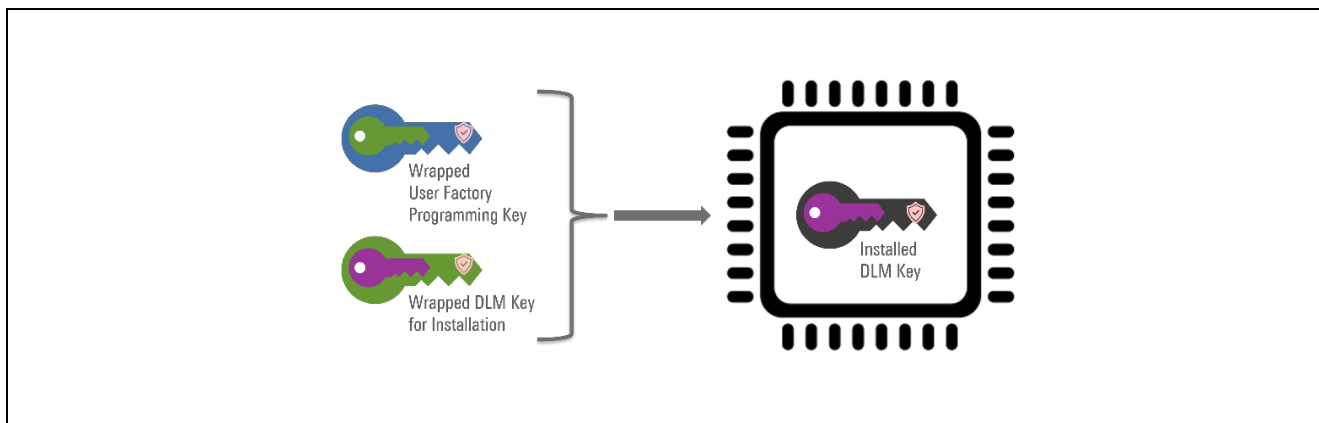


Figure 3-3 Generate Wrapped ALx_KEY

3.3 Establish PGP-Encrypted Communication with the Renesas Key Wrapping Service

The UFPK must be delivered to the Renesas key wrapping service using PGP encryption to obtain the W-UFPK. This requires an exchange of public keys. This is a one-time process.

3.3.1 Overview of the Renesas Key Wrapping Service

Using a web browser, enter the URL <https://dlm.renesas.com/keywrap/> to access the Renesas key wrapping service.

The *Key Wrap Service Operation Manual* can be accessed through the link [FAQ](#) on the right side of the window. The link to this manual is indicated in the answer to the question “Is there a manual of this system?” in the FAQ as shown in Figure 3-4.

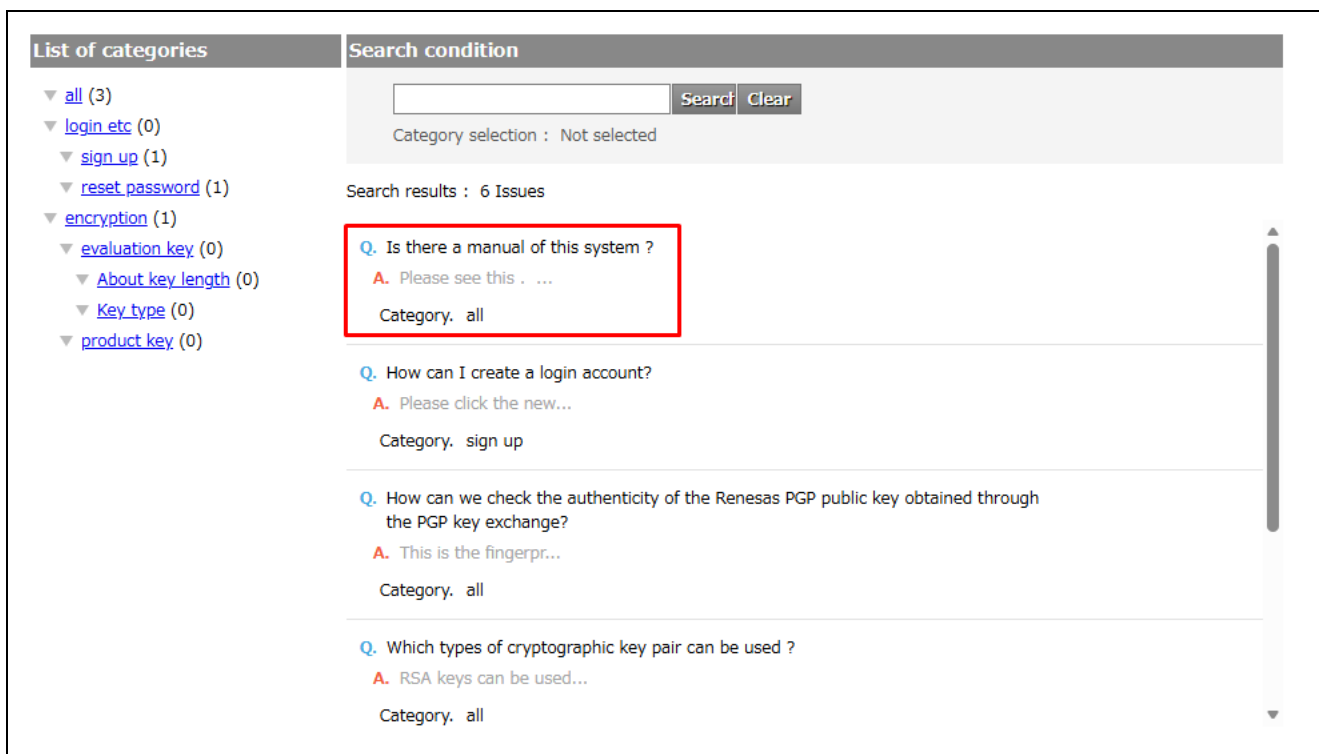


Figure 3-4 FAQ for the Renesas Key Wrapping Service and the Link to the Key Wrap Service Operation Manual

The information communicated between the user and the Renesas key wrapping service needs to be encrypted by OpenPGP. Figure 3-5 shows an overview of the operational flow for the key wrapping service with OpenPGP encryption as a security measure to protect the UFPK in transition.

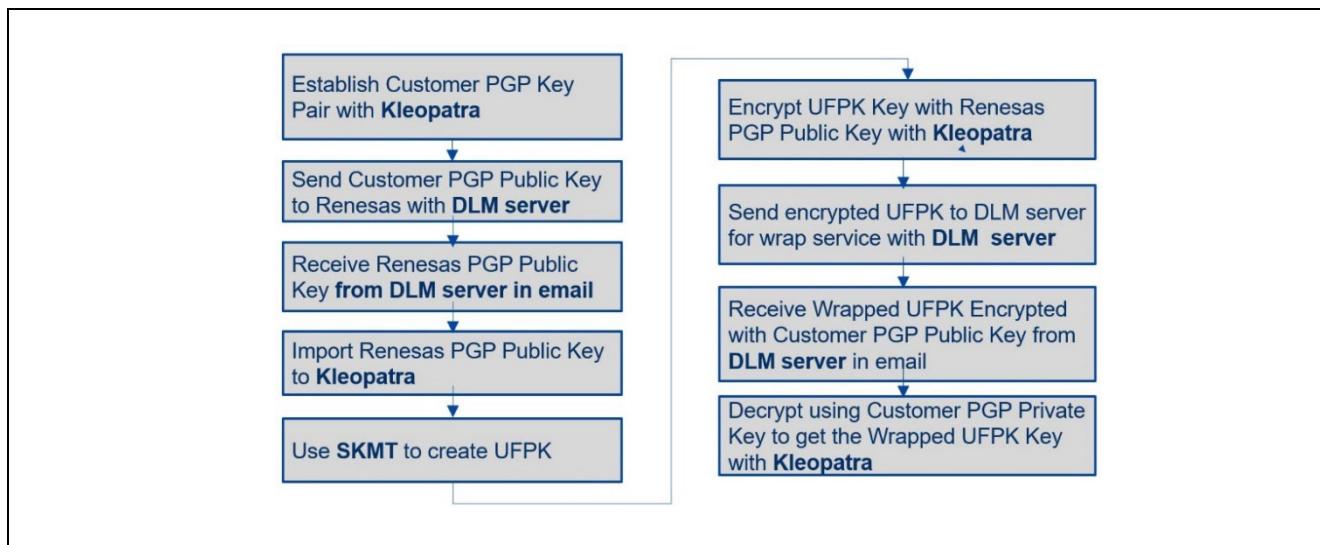


Figure 3-5 Overview of the Renesas Key Wrapping Service Using PGP

3.3.2 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.

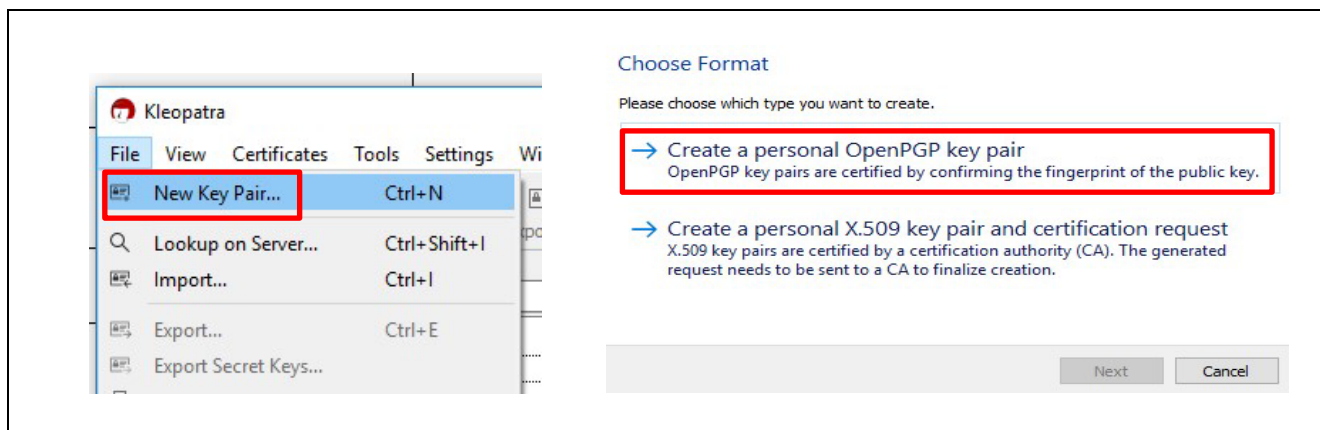
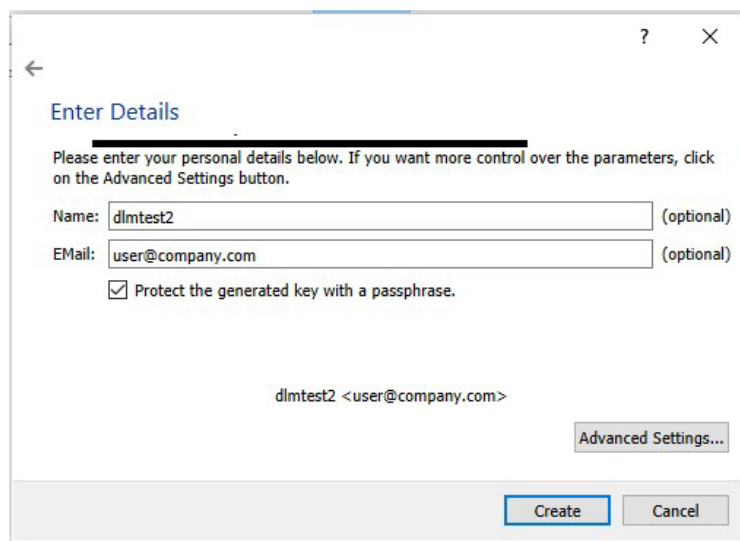


Figure 3-6 Generate the PGP Key Pair

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

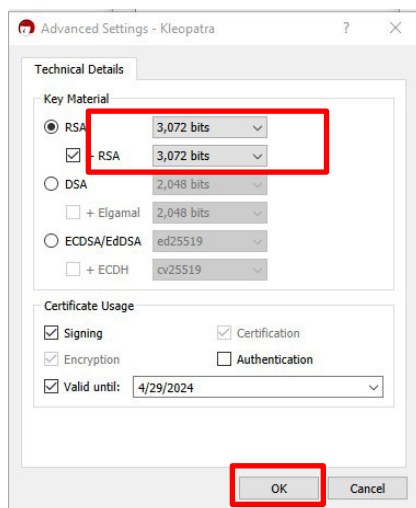


The 'Enter Details' dialog box contains the following fields and options:

- Name:** dlmtest2 (optional)
- Email:** user@company.com (optional)
- ☒ Protect the generated key with a passphrase.
- Summary text: dlmtest2 <user@company.com>
- Advanced Settings...** button
- Create** and **Cancel** buttons at the bottom.

Figure 3-7 Provide Credentials to Create the PGP Key Pair

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

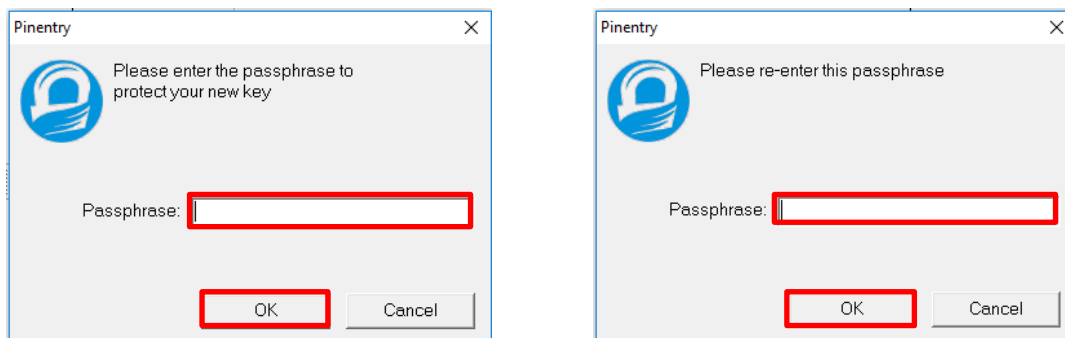


The 'Advanced Settings - Kleopatra' dialog box shows the following configuration:

- Key Material:**
 - ☒ RSA, 3,072 bits (highlighted with a red box)
 - ☒ RSA, 3,072 bits (highlighted with a red box)
 - ☐ DSA, 2,048 bits
 - ☐ + ElGamal, 2,048 bits
 - ☐ ECDSA/EdDSA, ed25519
 - ☐ + ECDH, cv25519
- Certificate Usage:**
 - ☒ Signing
 - ☒ Encryption
 - ☒ Valid until: 4/29/2024
 - ☒ Certification
 - ☐ Authentication
- OK** and **Cancel** buttons at the bottom (OK is highlighted with a red box).

Figure 3-8 Advanced Settings for the PGP Key Pair

Provide a passphrase to protect the private key. The passphrase is used to use the private key in the following process. **Be sure to save your passphrase.**



The figure shows two sequential Pinentry dialog boxes:

- First dialog:** "Please enter the passphrase to protect your new key". It has a "Passphrase:" label, an input field (highlighted with a red box), and "OK" and "Cancel" buttons (OK is highlighted with a red box).
- Second dialog:** "Please re-enter this passphrase". It has a "Passphrase:" label, an input field (highlighted with a red box), and "OK" and "Cancel" buttons (OK is highlighted with a red box).

Figure 3-9 Provide Passphrases

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

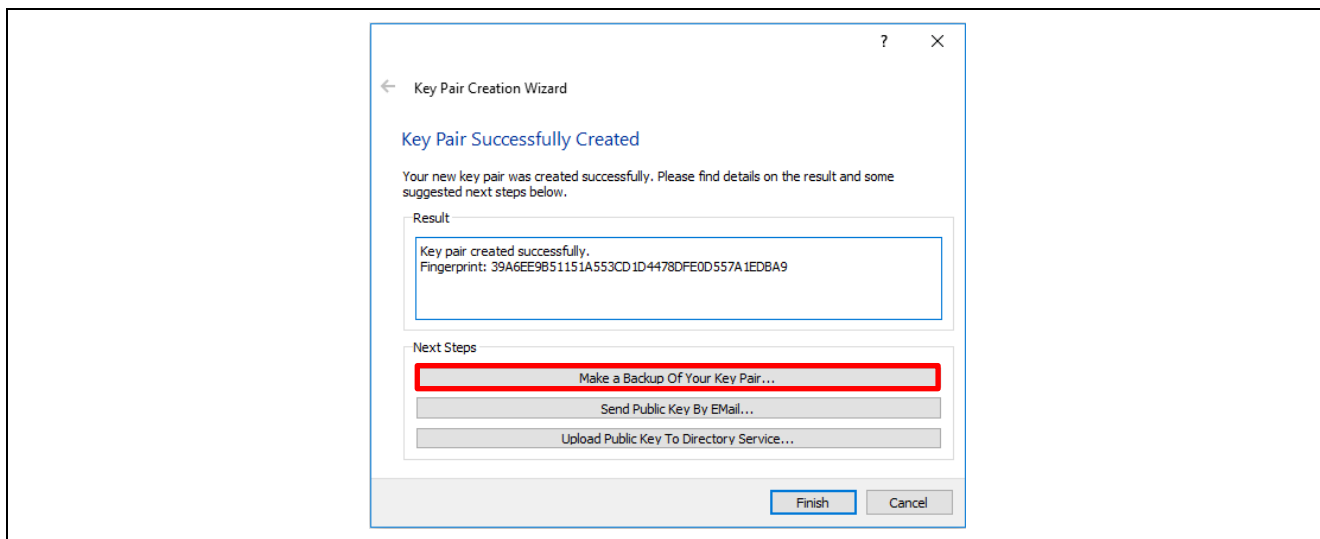


Figure 3-10 Key Pair Successfully Created

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

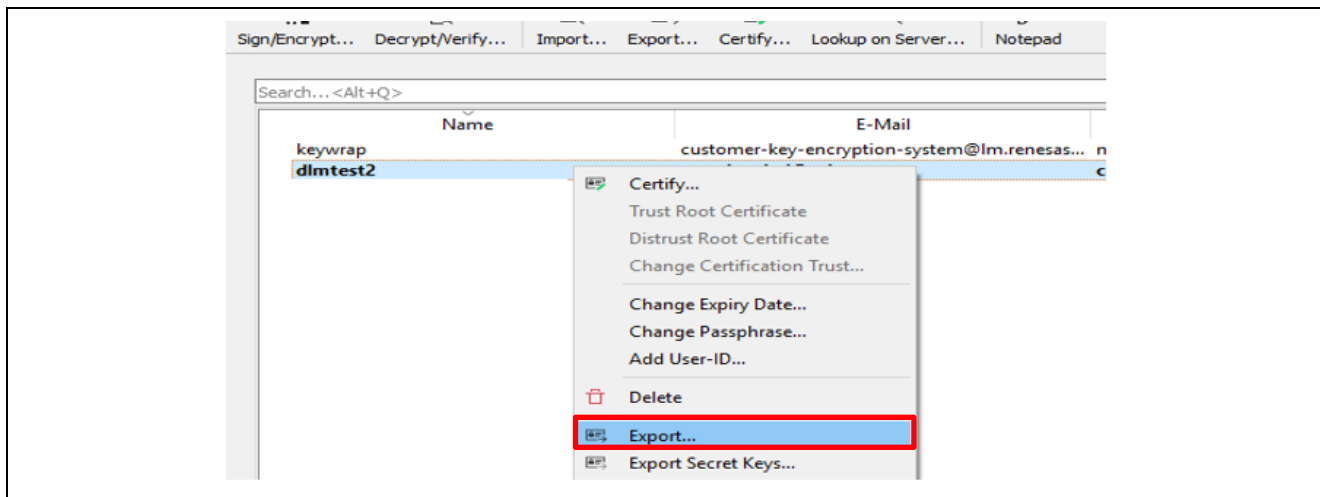


Figure 3-11 Export the Public Key

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

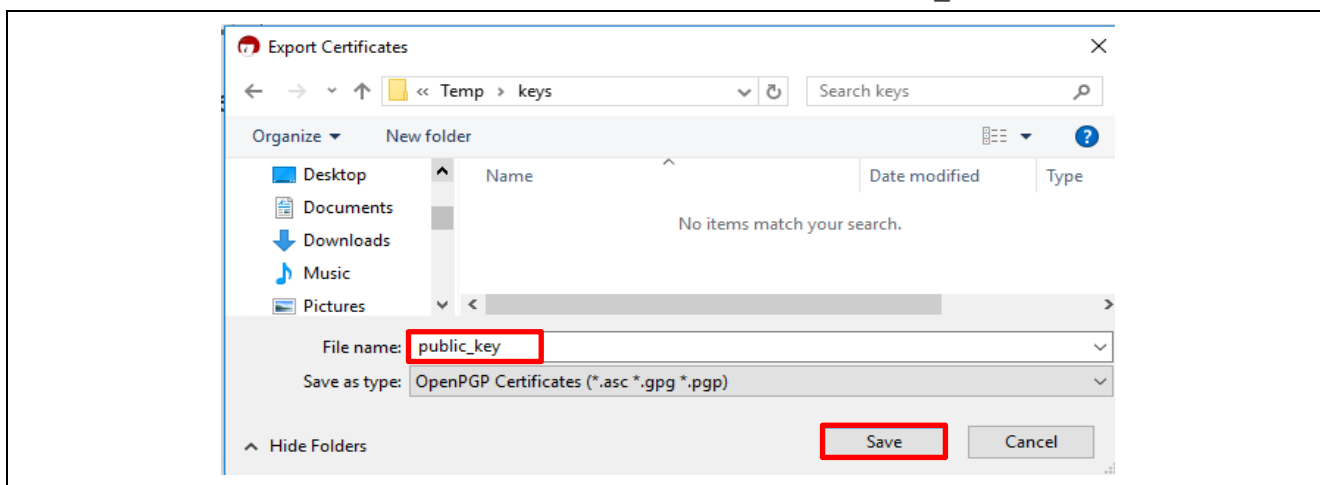


Figure 3-12 Save the Public Key

3.3.3 Registration with the Renesas Key Wrapping Service

This section provides a brief walk-through of the Renesas key wrapping service registration steps. This is a one-time process for new users. You can also review the section “New registration” in the Key Wrap Service Operation Manual for further details on the new registration steps.

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

Login screen of Key Wrap service

E-mail address:

Password:

Login

New registration
If you forgot your password...

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

E-mail address :

Send mail **Return**

Figure 3-13 Provide Email Address for New Registration

You will receive the registration link in an email, as shown in Figure 3-14.

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¶m=id%3Duf2SL2o3Cd7ODijiEN%252B0mISX7ghfiv8H9mP%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 3-14 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 :

Name :

Company Name :

Password :

Re-enter your password :

Figure 3-15 Register Customer Information

3.3.4 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 :

Name :

Company Name :

Figure 3-16 Start Using the Renesas Key Wrapping 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 Renesas key wrapping service.

--- 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.

I agree

I disagree.

[View PDF](#)

Figure 3-17 Agree with the Renesas Key Wrap Service Agreement

Communication between the user and the Renesas key wrapping service 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 wrapping service without exchanging PGP keys, you will receive the error message shown below.

PGP key exchange

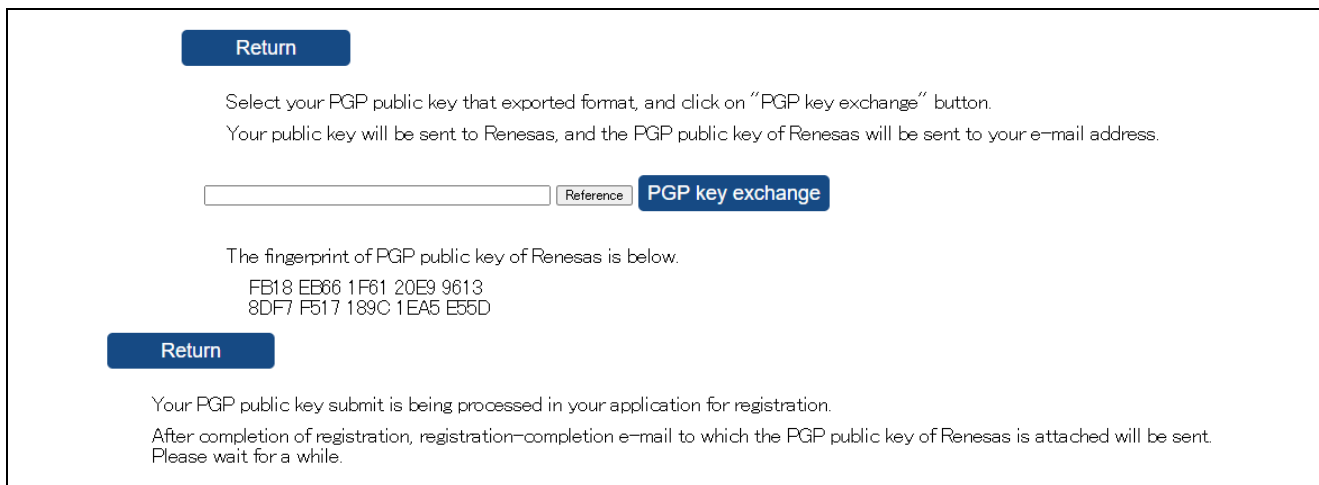
Display history

Refresh screen

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

Figure 3-18 Request for PGP Key Exchange

To exchange PGP keys, click the **PGP key exchange** button. The dialog shown in Figure 3-19 will open. Click **Reference** and select the public key that you created and exported earlier (Figure 3-12 in section 3.3.2). Next, click **PGP key exchange** and wait to receive the Renesas PGP public key at the email address you provided during registration.



Return

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.

Reference PGP key exchange

The fingerprint of PGP public key of Renesas is below.
FB18 EB66 1F61 20E9 9613
8DF7 F517 189C 1EA5 E55D

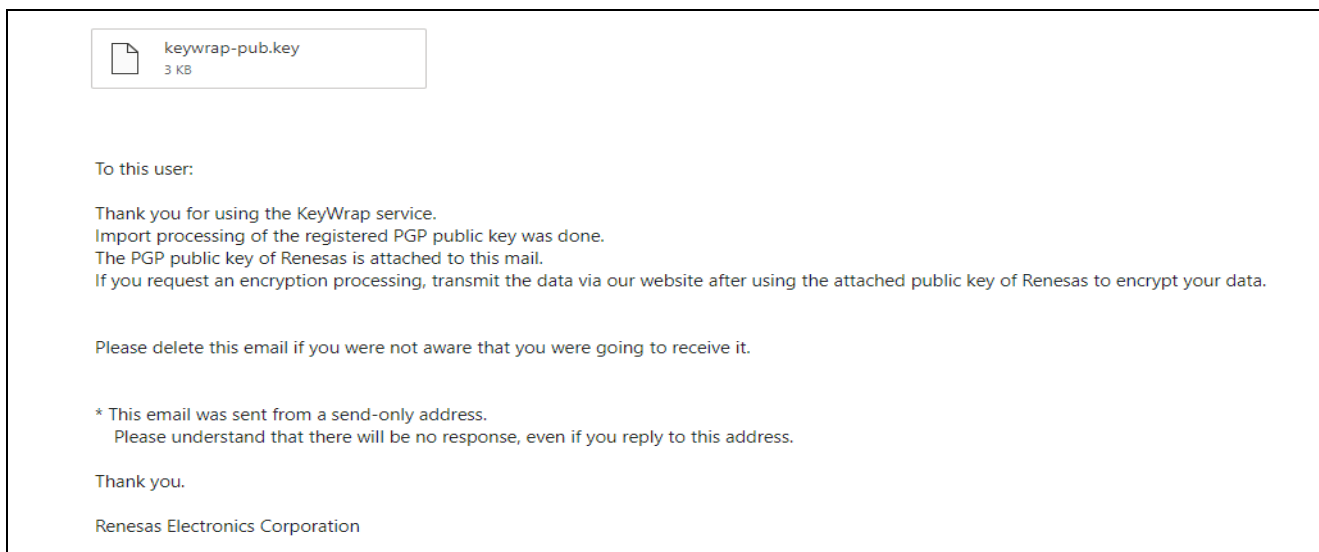
Return

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 3-19 Provide PGP Public Key to the Renesas Key Wrapping Service

You will receive an email with the content as shown in Figure 3-20, including the Renesas PGP 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 3-19.

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



keywrap-pub.key
3 KB

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 3-20 Receive the Renesas PGP Public Key

3.3.5 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 3-21.

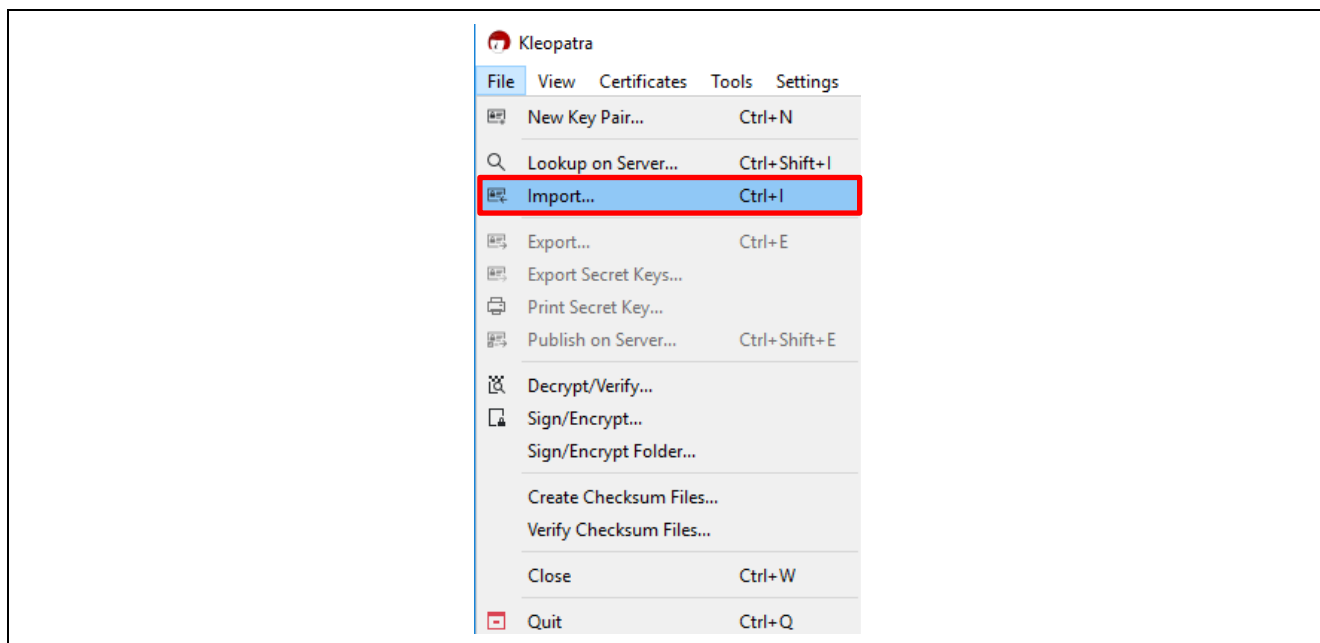


Figure 3-21 Import Renesas Public Key into Kleopatra

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

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

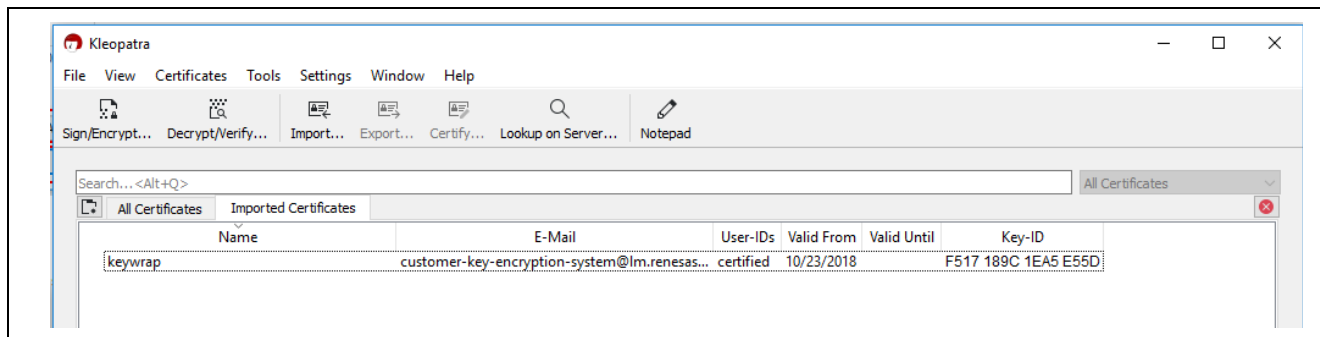


Figure 3-22 Renesas PGP Public Key

3.4 Create the UFPK and W-UFPK

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

3.4.1 Introduction to Renesas Security Key Management Tool

The Renesas Security Key Management Tool (SKMT) performs several functions during ALx_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 SKMT installer, and install it as required for your operating environment. This tool supports Windows, Linux, and macOS.

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 for users: a command line interface (CLI) and a graphical user interface (GUI). The GUI 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.

3.4.2 Create a UFPK File

The SKMT can be used in either graphical user interface (GUI) or command line interface (CLI) mode to create a UFPK file. This example uses the GUI.

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

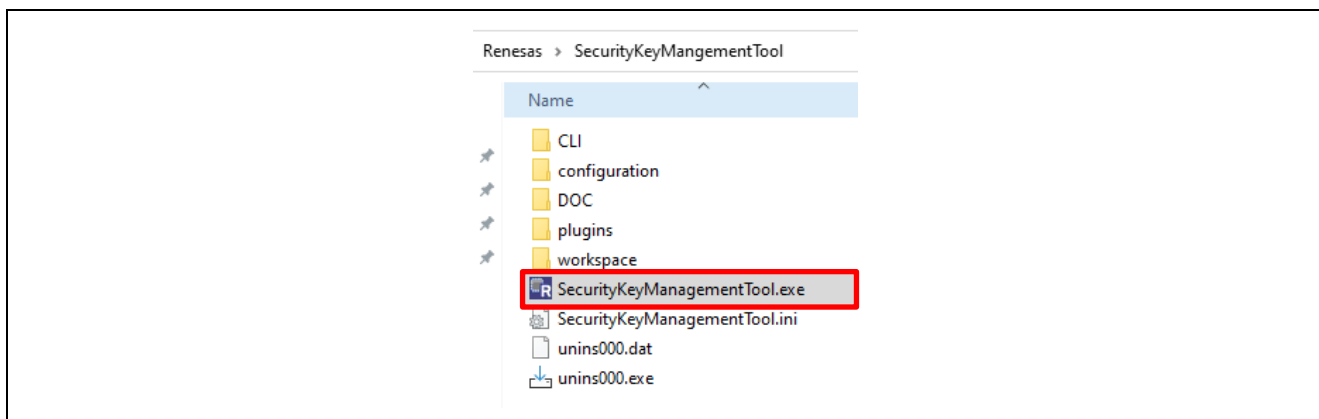


Figure 3-23 Launch SKMT GUI

In the **Overview** tab, select the **MCU/MPU and security engine as RX Family, RSIP-E50D Security Functions and Protected Mode**.

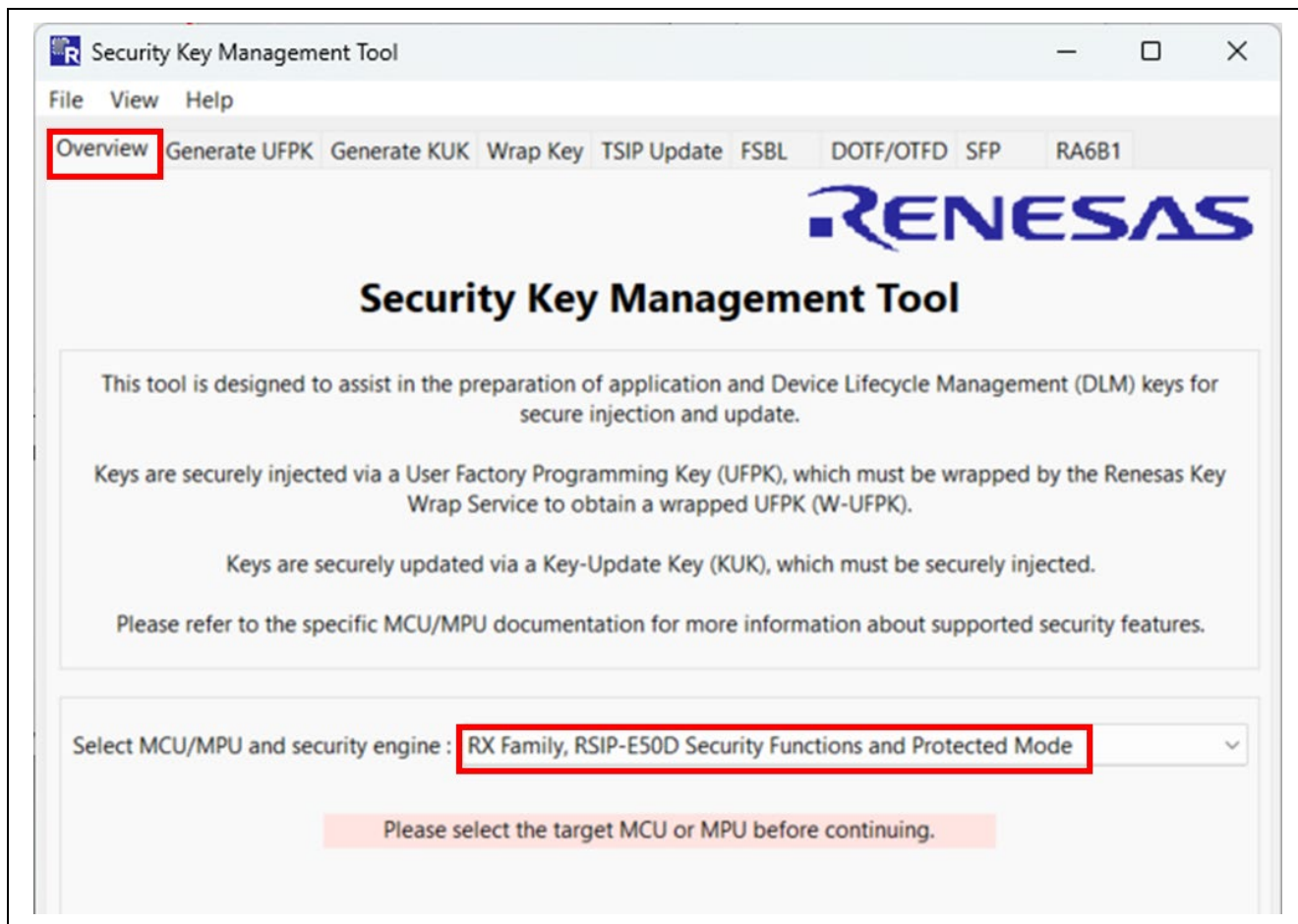


Figure 3-24 Select RX Family, RSIP-E50D Security Functions and Protected Mode

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

As shown in Figure 3-25 and Figure 3-26, you can specify a desired value for the UFPK, or the tool can create a random value. Figure 3-25 shows the tool generating a random value by selecting **Generate random value**. Figure 3-26 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`.

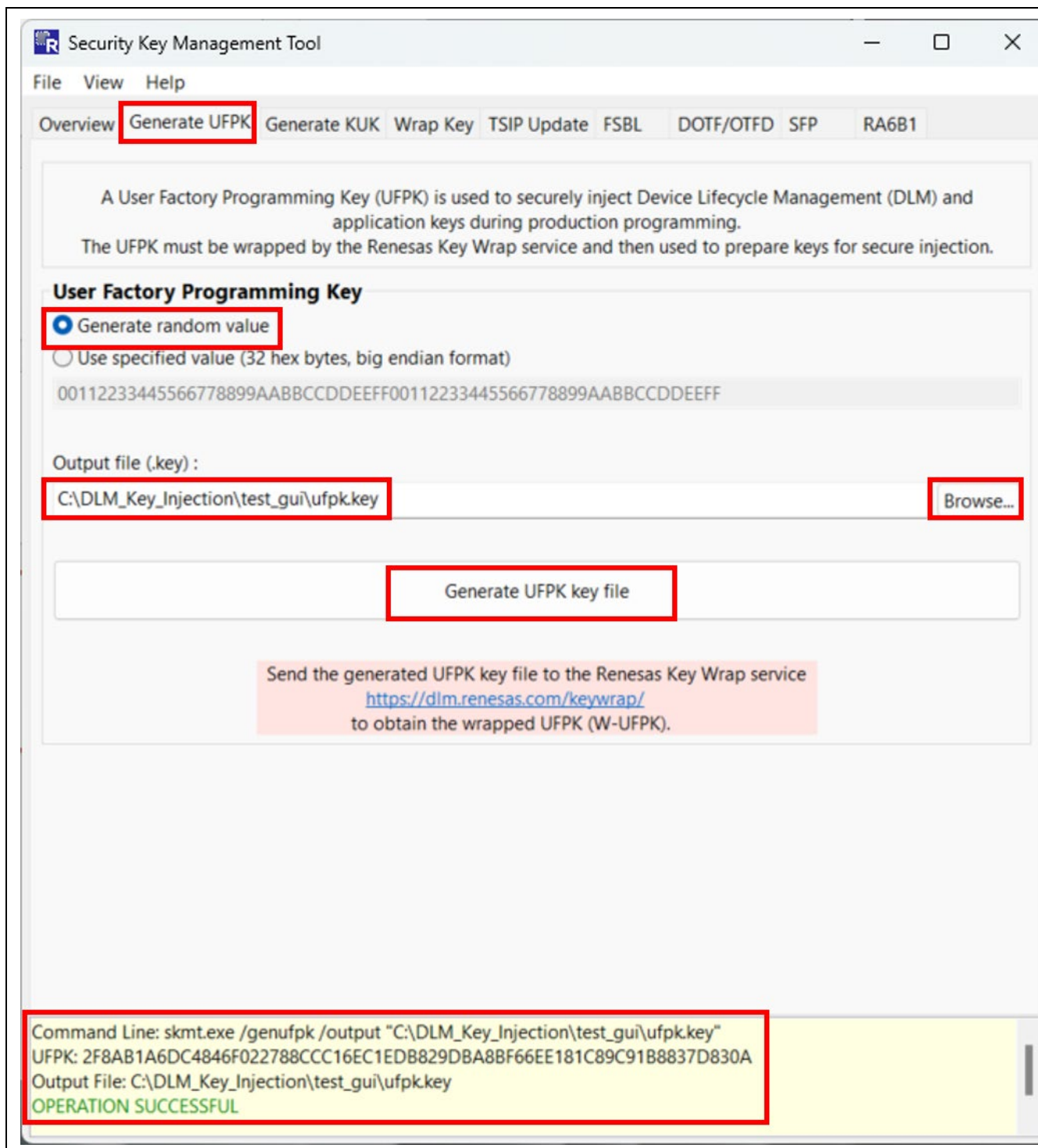


Figure 3-25 Generate a Random UFPK Using the GUI

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

000102030405060708090A0B0C0D0E0F000102030405060708090a0b0c0d0e0f

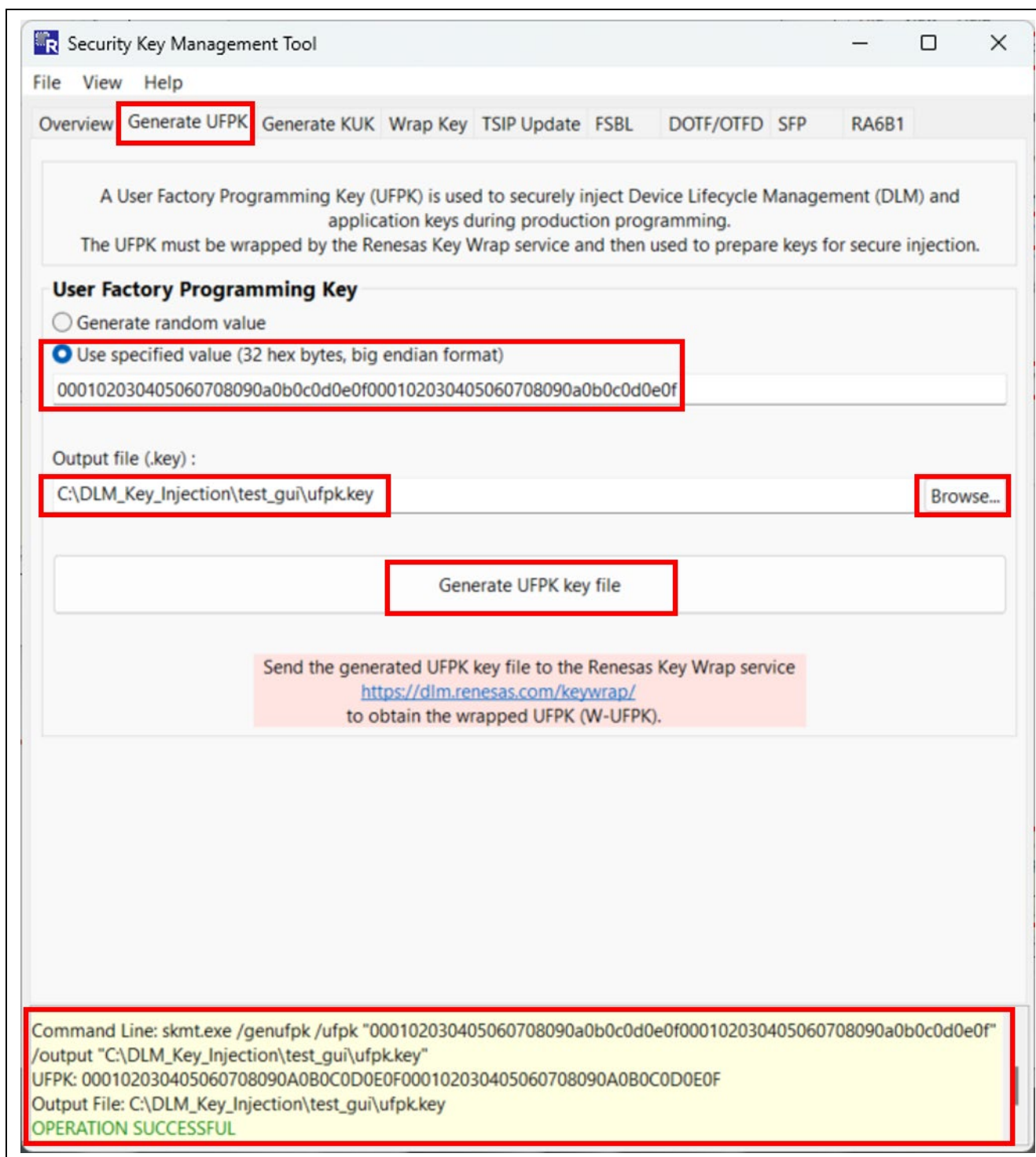


Figure 3-26 Create a Fixed UFPK Using the GUI

3.4.3 Encrypt UFPK with Renesas PGP Public Key

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

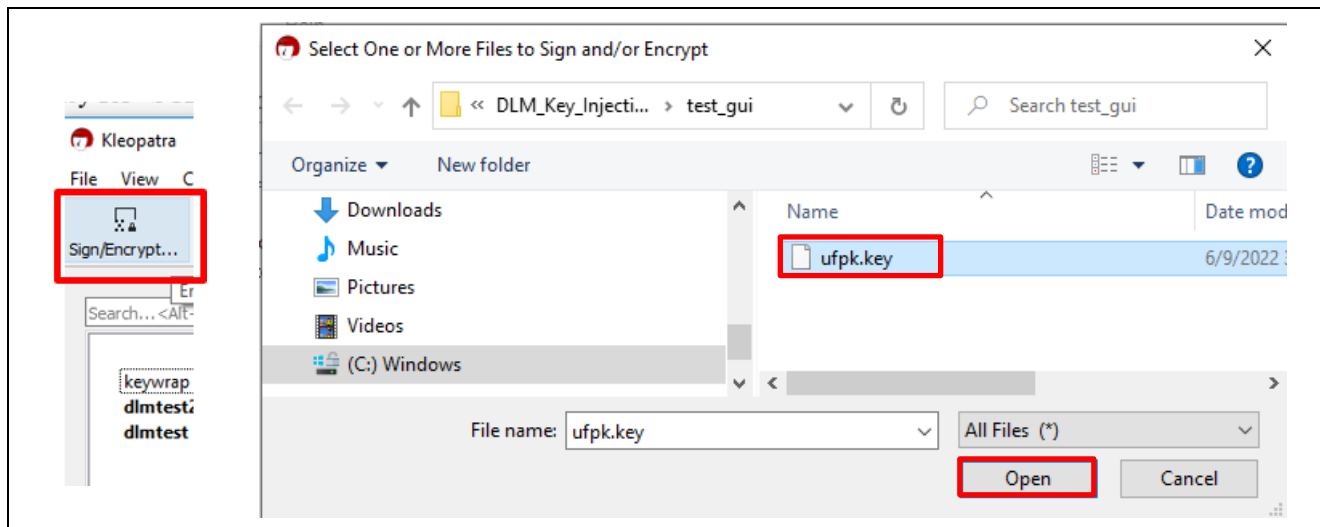


Figure 3-27 Select the UFPK Key to be Encrypted

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

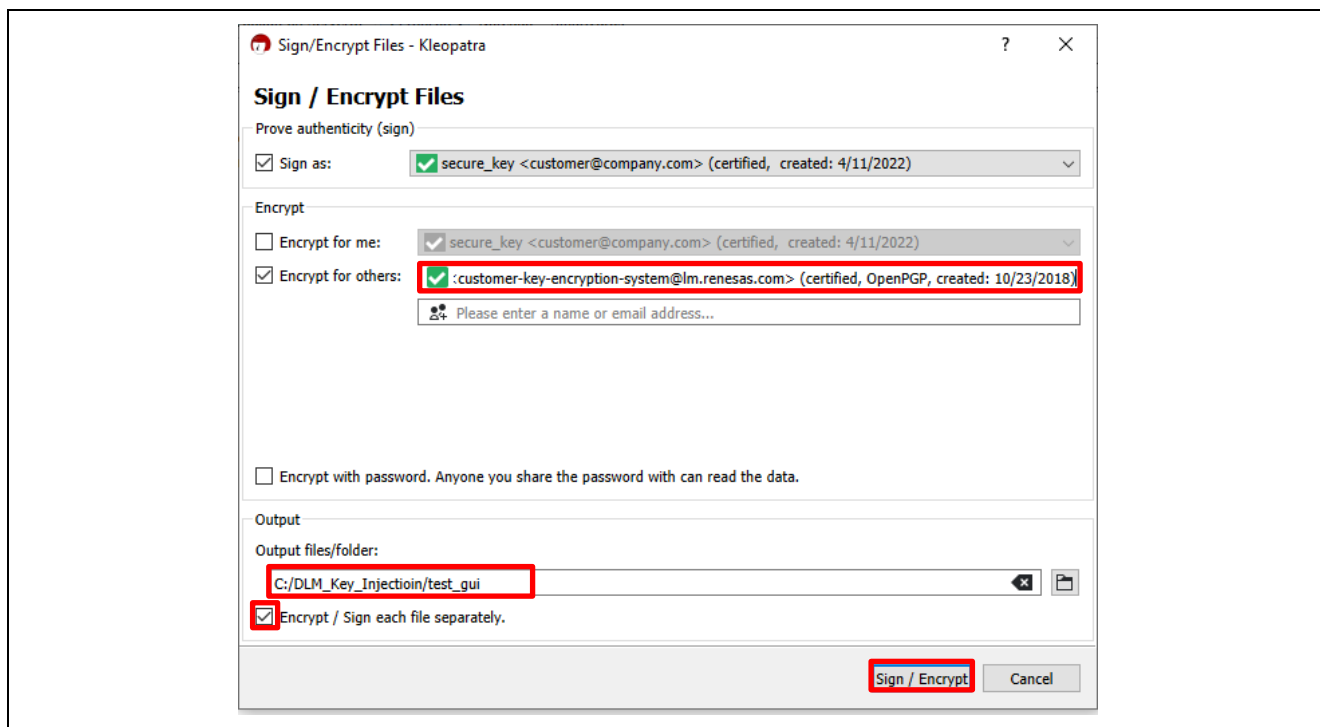


Figure 3-28 Use Renesas Public Key to Encrypt UFPK

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

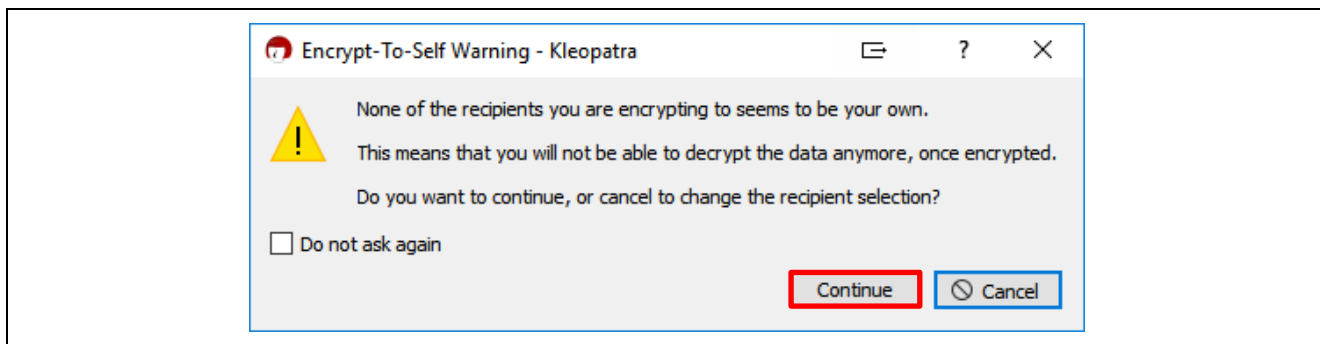


Figure 3-29 Confirm Encryption Option

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**.

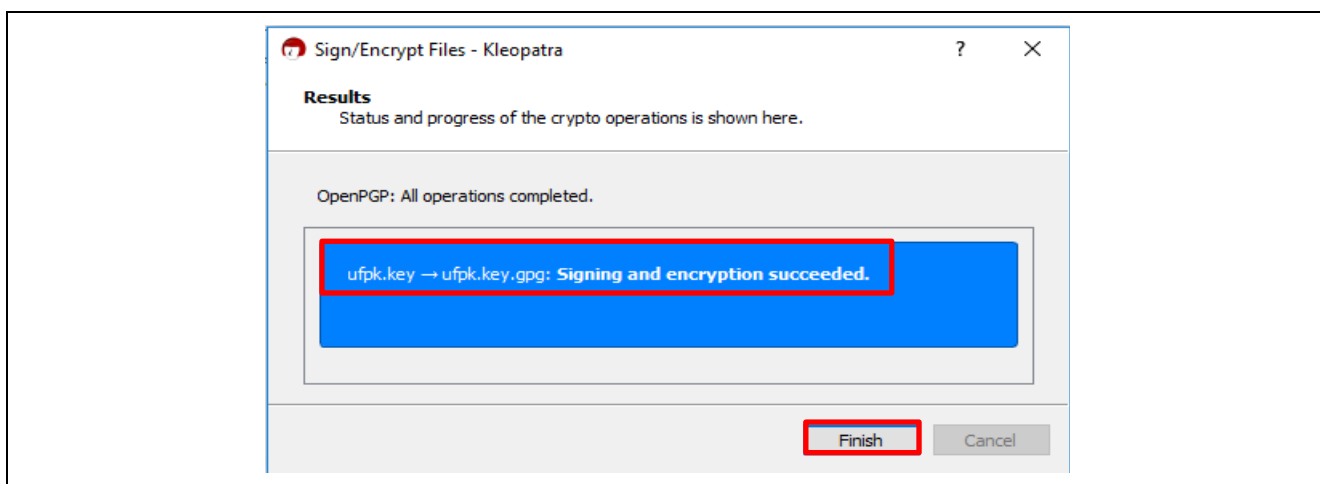


Figure 3-30 Encrypt the UFPK with Renesas Public Key

3.4.4 Send UFPK to the Renesas Key Wrapping Service

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

From the Renesas key wrapping service user interface, select the RX Family series and choose **RX74M Wrap UFPK** as shown below.

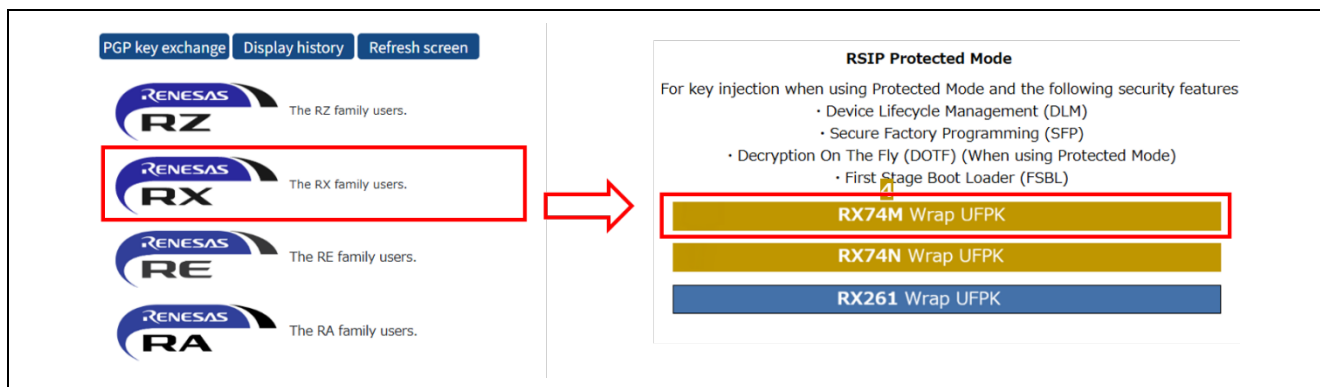


Figure 3-31 Select RX Device RSIP Protected Mode

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

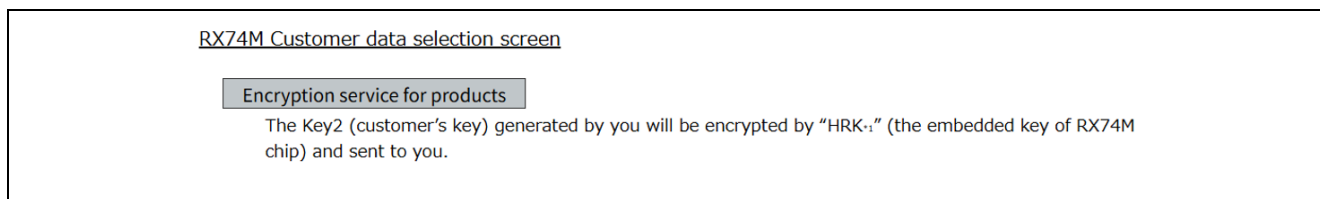


Figure 3-32 Select Encryption Service for Products

Next, click **Browse** and select the .gpg file generated from Figure 3-30.

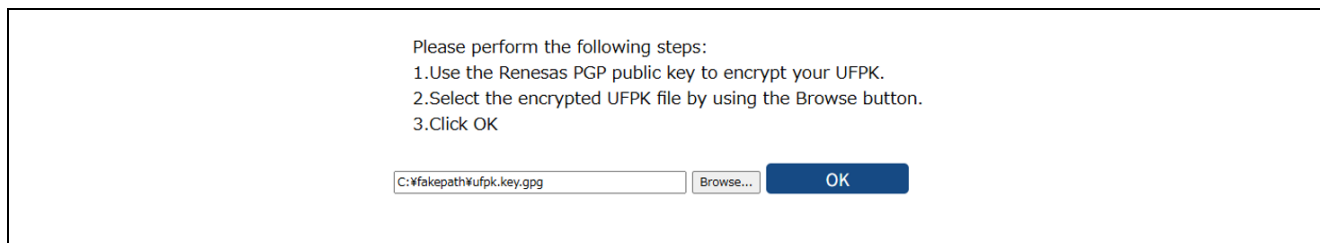


Figure 3-33 Send Encrypted UFPK to the Renesas Key Wrapping Service

Click **OK**. The Renesas key wrapping service will display a message that the file has been accepted, and the W-UFPK will be sent to the registered email address.

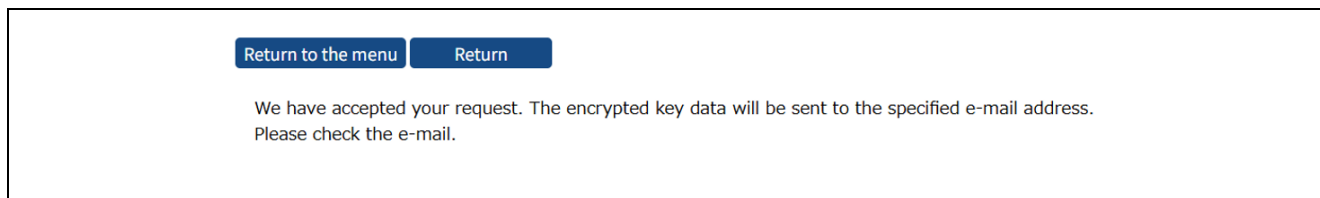


Figure 3-34 Message from the Renesas Key Wrapping Service

3.4.5 Receive the Encrypted W-UFPK

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

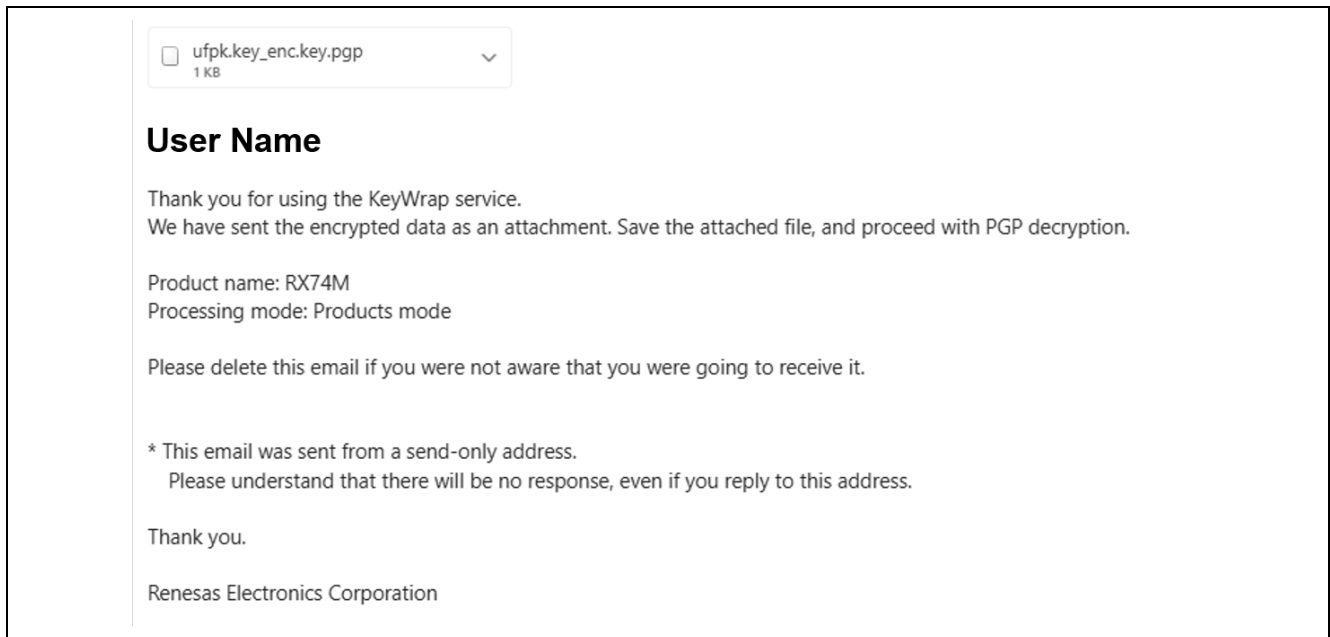


Figure 3-35 Receive the W-UFPK

Save this file for use in the next step.

3.4.6 Decrypt the Encrypted W-UFPK

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 file.

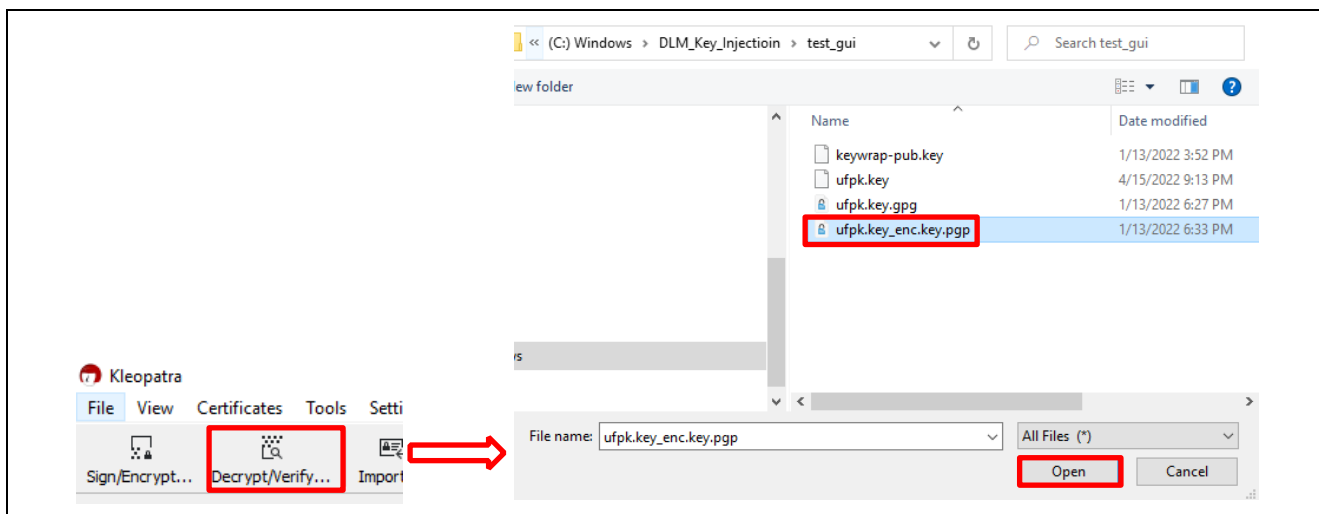


Figure 3-36 Encrypted W-UFPK File

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

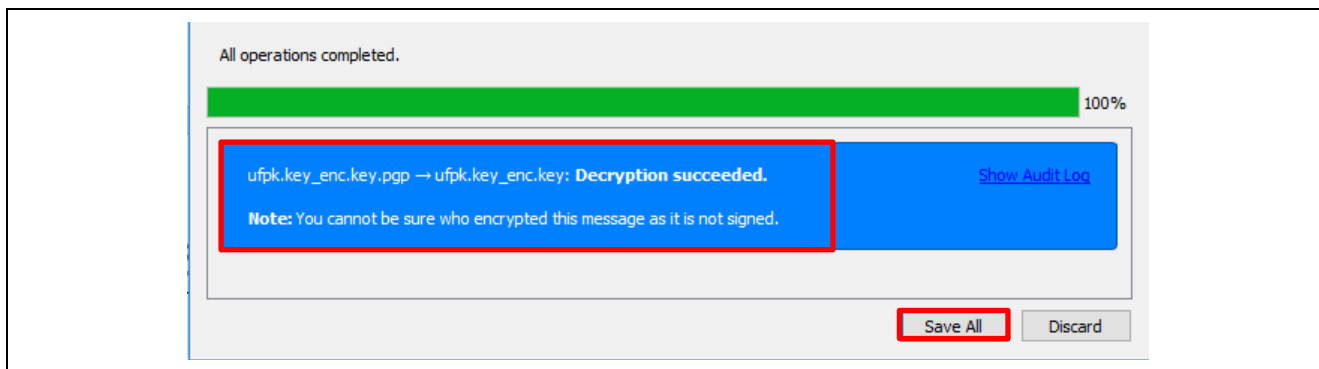


Figure 3-37 Decrypt with PGP Private Key

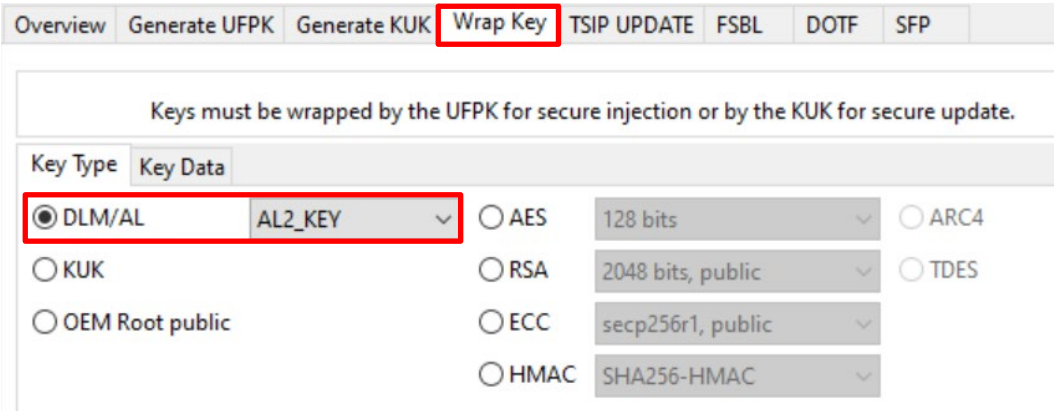
3.5 Create Wrapped ALx_KEY Using SKMT

To inject ALx_KEY, the key needs to be wrapped by the UFPK, and both wrapped ALx_KEY and 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 wrapped ALx_KEY and W-UFPK.

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

3.5.1 Wrap AL2_KEY

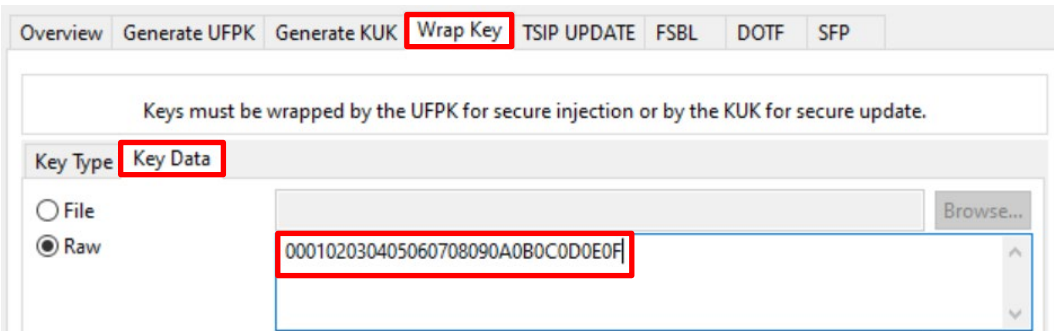
Launch the SKMT GUI and select the **Wrap Key** tab. Then open the **Key Type** tab and select **DLM/AL**. First, we will inject AL2_KEY.



The screenshot shows the SKMT GUI with the 'Wrap Key' tab selected. A message states: 'Keys must be wrapped by the UFPK for secure injection or by the KUK for secure update.' Under the 'Key Type' tab, 'DLM/AL' is selected with a radio button, and 'AL2_KEY' is chosen from a dropdown menu. To the right, several encryption options are listed with radio buttons: AES (128 bits), ARC4, KUK, RSA (2048 bits, public), TDES, OEM Root public, ECC (secp256r1, public), and HMAC (SHA256-HMAC).

Figure 3-38 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.



The screenshot shows the SKMT GUI with the 'Wrap Key' tab selected. The 'Key Data' tab is active, showing two options: 'File' and 'Raw'. The 'Raw' option is selected with a radio button. A text input field contains the hexadecimal key value '000102030405060708090A0B0C0D0E0F'. A 'Browse...' button is visible next to the 'File' option.

Figure 3-39 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 sections 3.4.2 and 3.4.6. Choose **Generate random value** for 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:\DLM_Key_Injection\test_gui\rx74m_ufpk.key Browse...
W-UFPK File: C:\DLM_Key_Injection\test_gui\rx74m_ufpk.key_enc.key Browse...

☐ KUK KUK File:

IV

☒ Generate random value
☐ Use specified value (16 hex bytes, big endian format) 00112233445566778899AABBCCDDEEFF

Output

Format: RFP File: C:\DLM_Key_Injection\AL2_KEY.rkey Browse...
Endian: Little ☐ Output additional data
Address: 10000 Key name:

Generate file

Command Line: skmt.exe /genkey /ufpk "C:\DLM_Key_Injection\test_gui\rx74m_ufpk.key" /wufpk "C:\DLM_Key_Injection\test_gui\rx74m_ufpk.key_enc.key" /mcu "RX-RSIP-E50D" /keytype "DLM-AL2" /key "000102030405060708090A0B0C0D0E0F" /filetype "rfp" /output "C:\DLM_Key_Injection\AL2_KEY.rkey"

Output File: C:\DLM_Key_Injection\AL2_KEY.rkey

UFPK: 000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F
W-UFPK: [Redacted]
IV: 7B4F8FA46C128AC44CC1C61F424FE148
Encrypted key: [Redacted] Wrapped AL2_KEY
OPERATION SUCCESSFUL

Figure 3-40 Generate the RFP Injection File for AL2_KEY

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

3.5.2 Wrap AL1_KEY

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

Overview Generate UFPK Generate KUK **Wrap Key** TSIP UPDATE FSBL DOTF SFP

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

Key Type Key Data

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

Figure 3-41 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 (010102030405060708090A0B0C0D0E0F).

The screenshot shows the 'Wrap Key' tab with the following details:

- Navigation tabs: Overview, Generate UFPK, Generate KUK, **Wrap Key**, TSIP UPDATE, FSBL, DOTF, SFP.
- Message: Keys must be wrapped by the UFPK for secure injection or by the KUK for secure update.
- Key Type: **Raw** (selected), File.
- Key Data: 010102030405060708090A0B0C0D0E0F

Figure 3-42 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 sections 3.4.2 and 3.4.6. Choose **Generate random value** for 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.

The screenshot shows the 'Wrapping Key' section with the following details:

- Wrapping Key:**
 - UFPK:** Selected. UFPK File: C:\DLM_Key_Injection\test_gui\rx74m_ufpk.key. W-UFPK File: C:\DLM_Key_Injection\test_gui\rx74m_ufpk.key_enc.key.
 - KUK:** Not selected.
- IV:**
 - Generate random value:** Selected.
 - Use specified value (16 hex bytes, big endian format):** 00112233445566778899AABBCCDDEEFF
- Output:**
 - Format:** RFP (selected).
 - File:** C:\DLM_Key_Injection\AL1_KEY.rkey.
 - Endian:** Little.
 - Address:** 10000.
 - Key name:** (empty).
 - Output additional data:** (unchecked).
 - Generate file:** Button.
- Command Line:**

```
skmt.exe /genkey /ufpk "C:\DLM_Key_Injection\test_gui\rx74m_ufpk.key" /wufpk "C:\DLM_Key_Injection\test_gui\rx74m_ufpk.key_enc.key" /mcu "RX-RSIP-E50D" /keytype "DLM-AL1" /key "000102030405060708090A0B0C0D0E0F" /filetype "rfp" /output "C:\DLM_Key_Injection\AL1_KEY.rkey"
```
- Summary:**
 - Output File:** C:\DLM_Key_Injection\AL1_KEY.rkey
 - UFPK:** 000102030405060708090A0B0C0D0E0F000102030405060708090A0B0C0D0E0F
 - W-UFPK:** (redacted)
 - IV:** F40A6D18D7C62462B066B312E333A01E
 - Encrypted key:** (redacted)
 - Wrapped AL1_KEY:** (redacted)
 - OPERATION SUCCESSFUL**

Figure 3-43 Generate the RFP Injection File for AL1_KEY

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

3.6 Inject Authentication Keys Using the RFP

This section demonstrates how to inject ALx_KEY. For instructions on how to create an RFP project and establish connections to the target board, see the *RFP User's Manual*. In addition, set pins 1 and 2 of SW6 on the EK-RX74M board to the OFF position to connect the EK-RX74M to the PC via a USB device cable. This section provides key configuration settings for each of the state transitions.

3.6.1 Inject AL2_KEY

AL2_KEY and RMA_KEY can be injected when the MCU is in an AL2 state.

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

This example uses the AL2_KEY Renesas key file generated in section 3.5.1 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 is a procedure for failure analysis, it is not demonstrated here.**

Unzip rx74m_dlm_key_inject_rfp.zip to reveal rx74m_dlm_key_inject_rpj. Launch the RFP and open the rx74m_dlm_key_inject.rpj RFP project file then 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 File**, and then click the “...” to the right to select the corresponding AL2_KEY file.

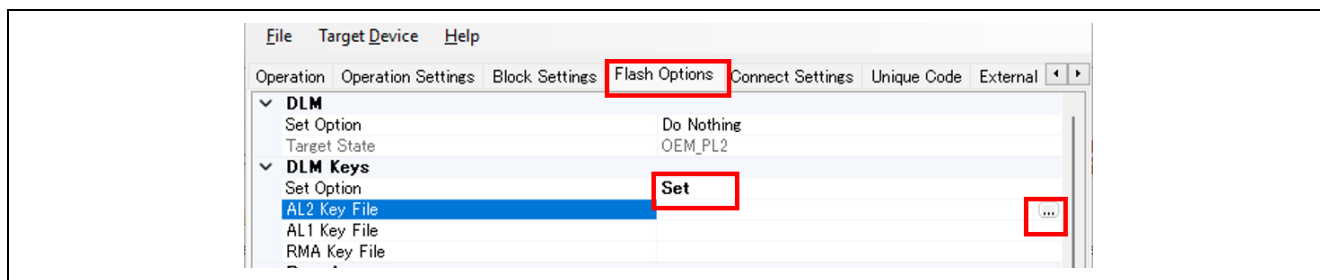


Figure 3-44 Select AL2_KEY to Inject

Select AL2_KEY.rkey generated earlier.

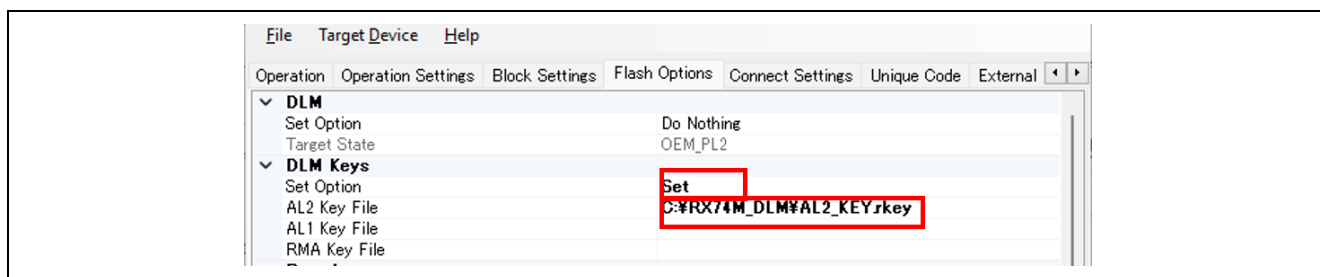


Figure 3-45 Select AL2_KEY to Inject

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

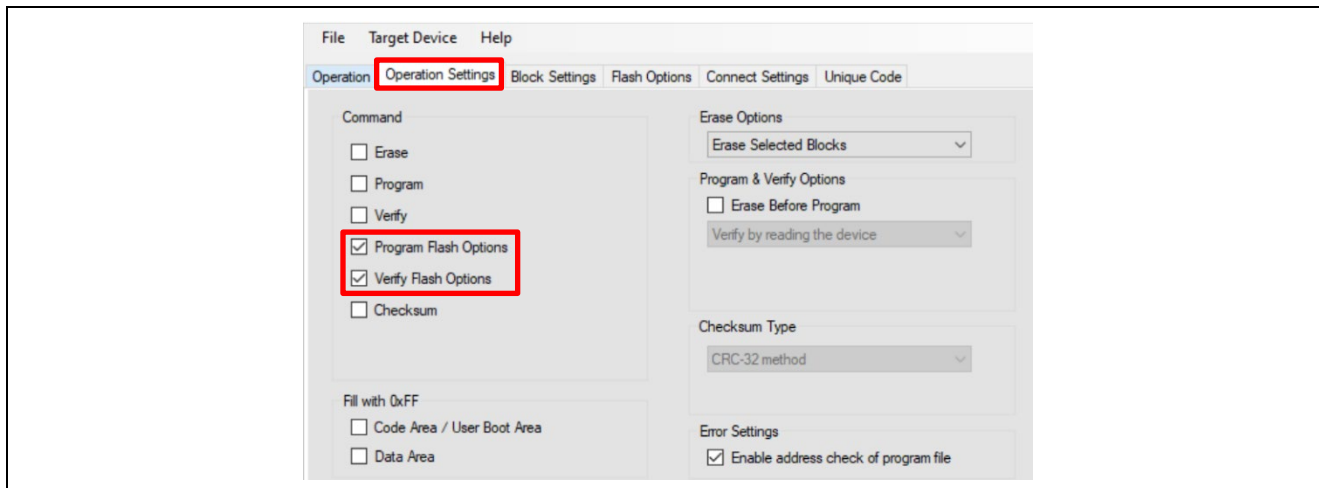


Figure 3-46 Select Program and Verify Flash Option Settings

Note that ALx_KEY 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.

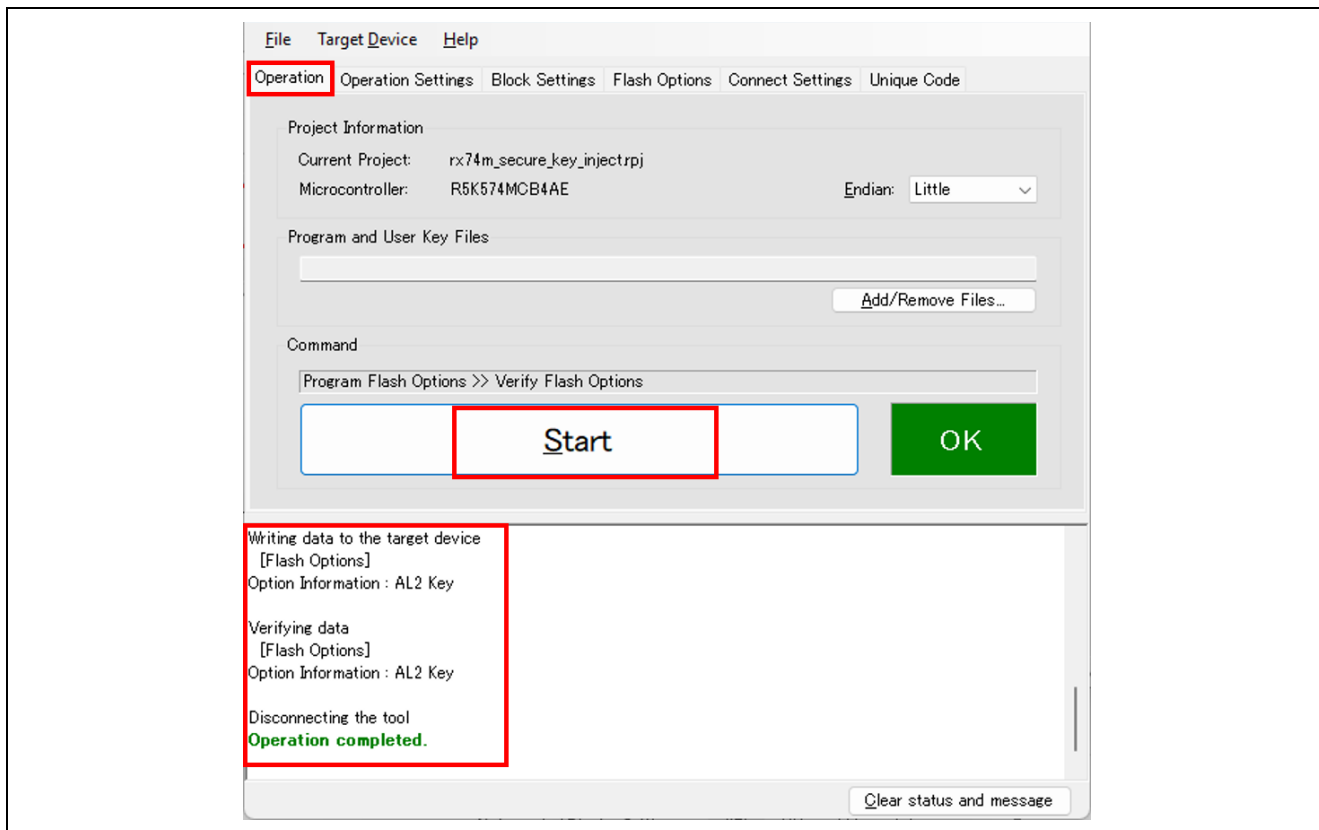


Figure 3-47 Inject AL2_KEY

3.6.2 Inject AL1_KEY

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

When injecting AL1_KEY at OEM_PL2, it is possible to inject AL1_KEY alone or inject both AL1_KEY and AL2_KEY by setting **AL2 Key File** and **AL1 Key File** for **DLM Keys** on the **Flash Options** tab, as shown in Figure 3-48.

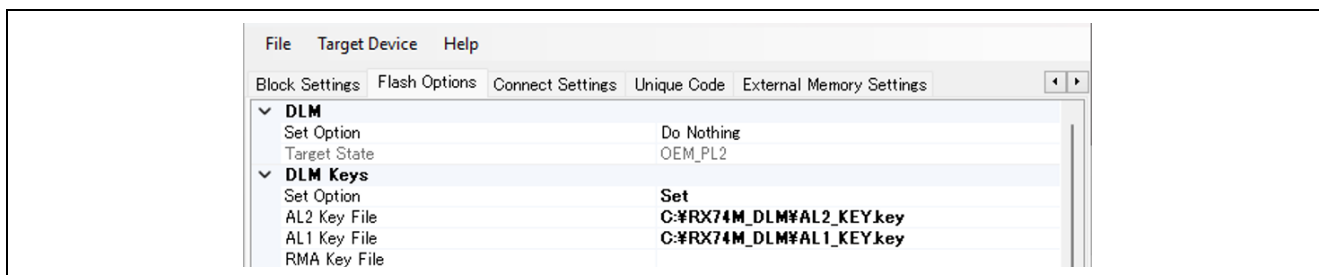


Figure 3-48 Include AL1_KEY and AL2_KEY for Key Injection

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

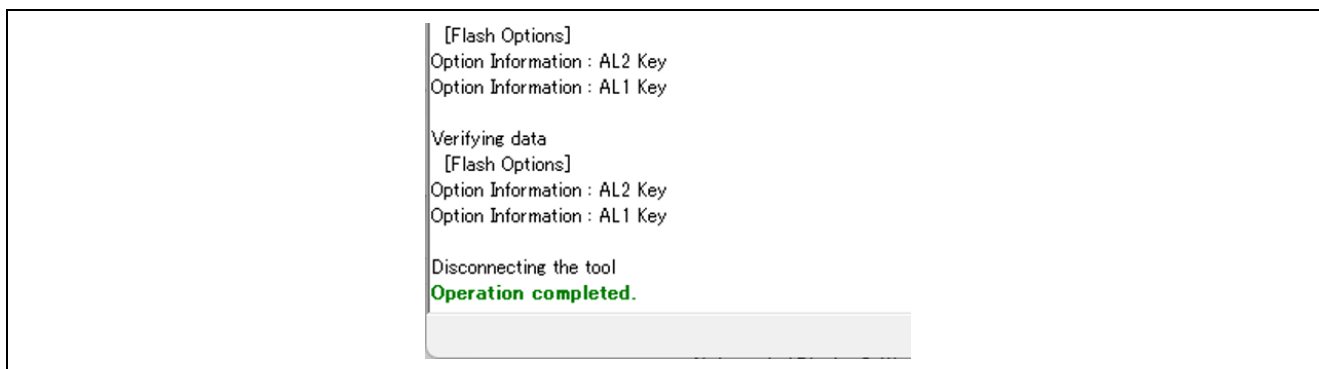


Figure 3-49 Successful Injection of AL1_KEY and AL2_KEY

The next example shows how AL1_KEY can be injected in the OEM_PL1 state. In this case, transition to OEM_PL1 following Figure 3-50 first and then inject AL1_KEY alone, as shown in Figure 3-52.

Select **Target Device -> DLM Transition**, set **Target DLM State** as **OEM_PL1**, and click on **Transition**.

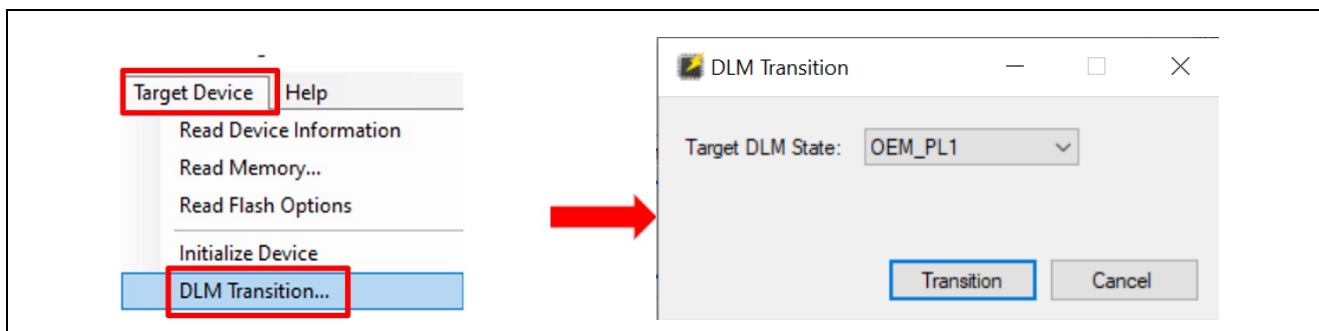


Figure 3-50 Transition of the MCU Device Lifecycle State to AL1

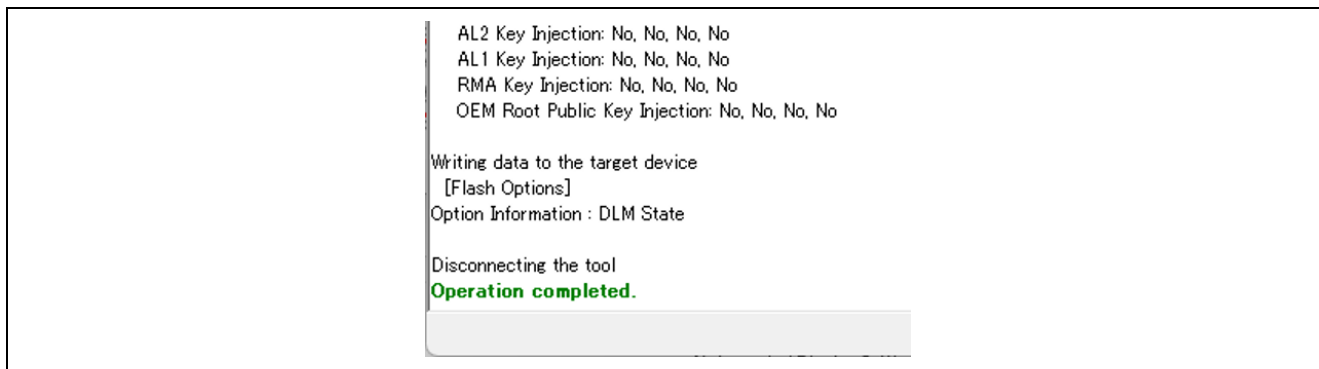


Figure 3-51 Successful Transition to OEM_PL1

Set **AL1 Key File** for **DLM Keys** on the **Flash Options** tab and add the AL1_KEY file entry as shown below.

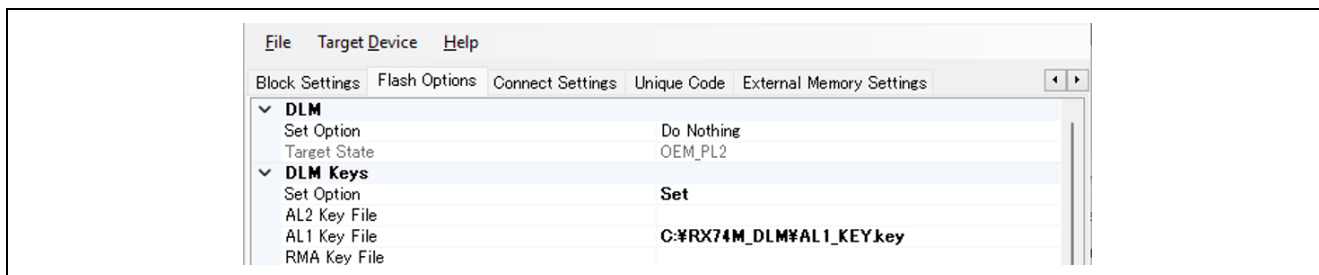


Figure 3-52 Select AL1_KEY to Inject

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

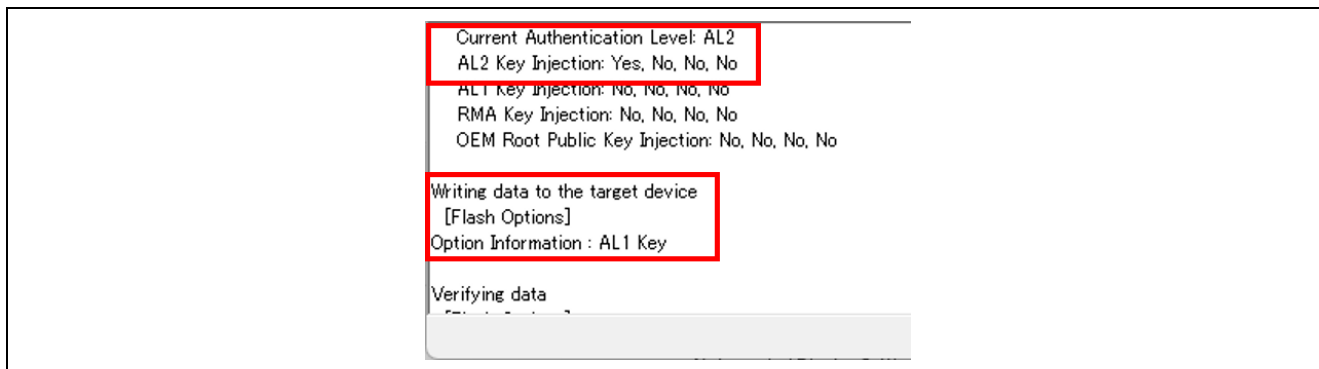


Figure 3-53 AL1_KEY Injected with Existing AL2_KEY in OEM_PL1

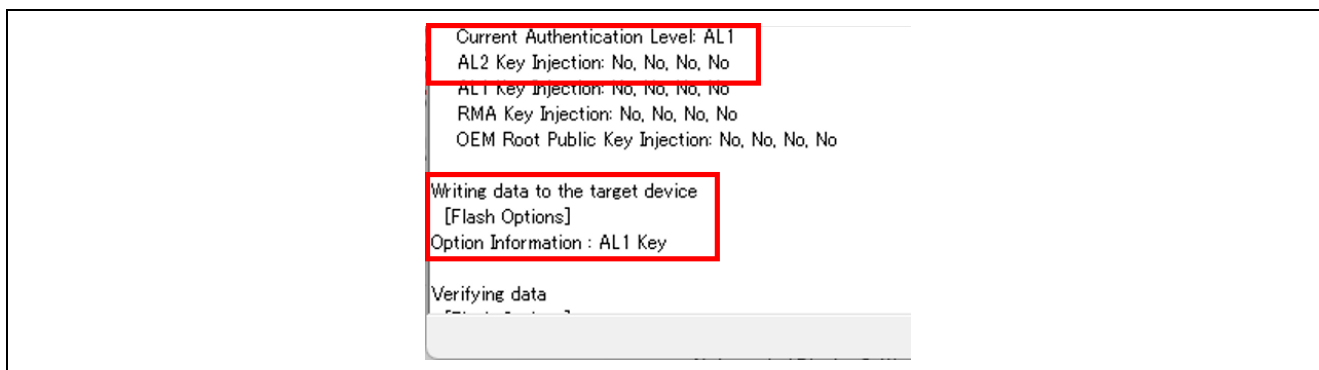


Figure 3-54 AL1_KEY Injected without Existing AL2_KEY in OEM_PL1

4. Transitions of the Authentication and Protection Levels

This section explains how to handle transitions between DLM states by using the RFP and RDPM.

4.1 State Transitions without Authentication

This section explains how to handle transitions of the state of device lifecycle management (DLM), protection level (PL), and authentication level (AL) **without authentication**.

The general lifecycle flow from development, production, to deployment is reflected in the transitions, and access rights are lowered each time a transition is made. For details on connecting the EK-RX74M and the PC, refer to section 3.6.

4.1.1 State Transitions without Authentication Using the RFP

The RFP supports end-to-end production processes and also supports the following functionality.

- All state transitions with authentication enabled and disabled.
- Key injection of ALx_KEY and RMA_KEY.
- Disabling of the Initialize command. This may be desirable when the device is to be deployed in the OEM_PL0 state and accidental erasure of MRAM contents must be avoided. **However, if the Initialize command is disabled, it cannot be re-enabled.**
- Disabling of AL2 and AL1 authentication. This may be required when the device is deployed in the OEM_PL0 state and debugging of the system is disabled. **However, if AL2 authentication and AL1 authentication are disabled, they cannot be re-enabled.**
- Disabling of entry to the LCK_BOOT transition. This may be required when the device is deployed in the PL0 state and is to be kept in a state that enables transitions for other usage in the future. **However, if LCK_BOOT transition is disabled, it cannot be re-enabled.**

To use the RFP to initialize a device, select **Target Device -> Initialize Device**.

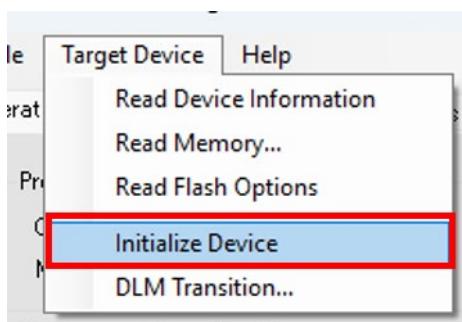


Figure 4-1 Use the Initialize Device Command

A transition of PL is enabled by selecting the target state of the transition for **Target DLM State** from **Target Device -> DLM Transition** as shown in Figure 4-2. The PL determines the corresponding AL.

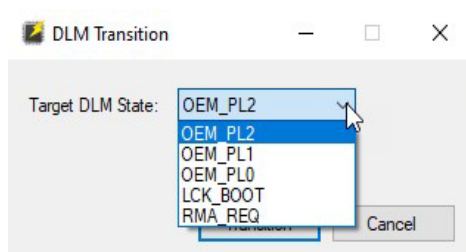


Figure 4-2 Available State Transitions of Lifecycle Supported by the RFP

The Initialize command is disabled by selecting **Set** for **Set Option** and **Yes** for **Disable Initialize Command** under **Security** on the **Flash Options** tab as shown below. As stated in the [Warning] message box, once this setting is made, the Initialize command cannot be re-enabled.

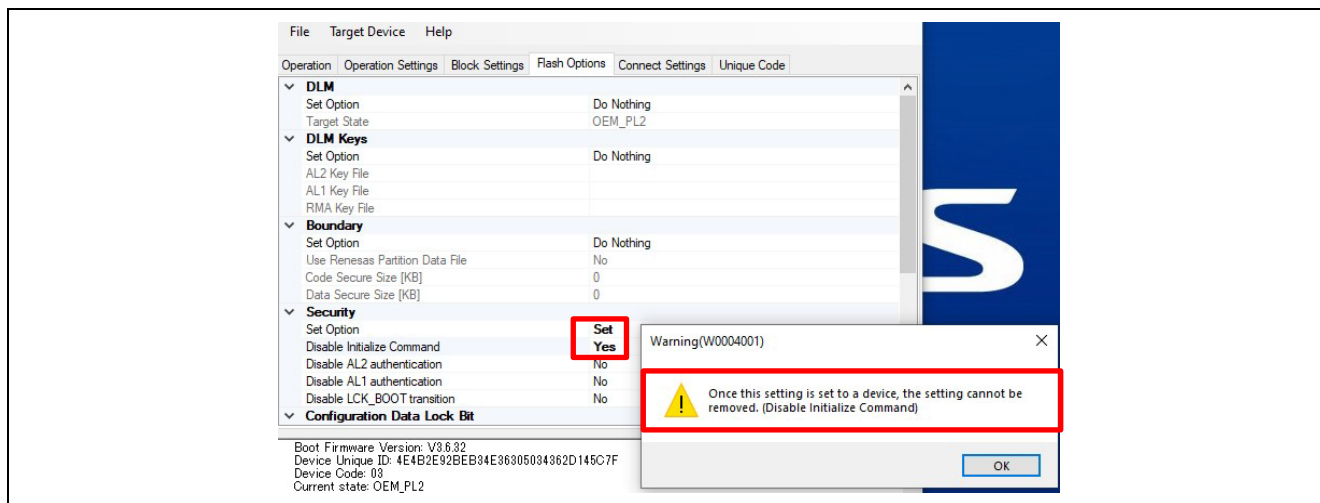


Figure 4-3 Disable the Initialize Command

If **Yes** is selected for the **Disable AL2/AL1 authentication** or **Disable LCK_BOOT transition** option, the same warning message will be displayed.

4.1.2 State Transitions without Authentication Using the Renesas Device Partition Manager (RDPM)

The RDPM is a utility tool which has been integrated into the e² studio IDE for the purpose of managing the lifecycle states of devices being developed. The RDPM has the following functionality:

- referring to the lifecycle state of the current device;
- referring to the Protect Zone boundary settings of the device;
- **initializing** the device to the OEM_PL2 state (except for the case when MRAM is permanently protected); and
- setting the Protect Zone boundaries.

In the e² studio IDE, select **Run -> Renesas Debug Tools -> Renesas Device Partition Manager** to open the RDPM.



Figure 4-4 Open the RDPM

Figure 4-5 shows the method of **initializing** the device to the default state at shipment. Select **Renesas RX** for **Device Family**, turn on **Initialize device**, specify the connection methods, and click on **Run**.

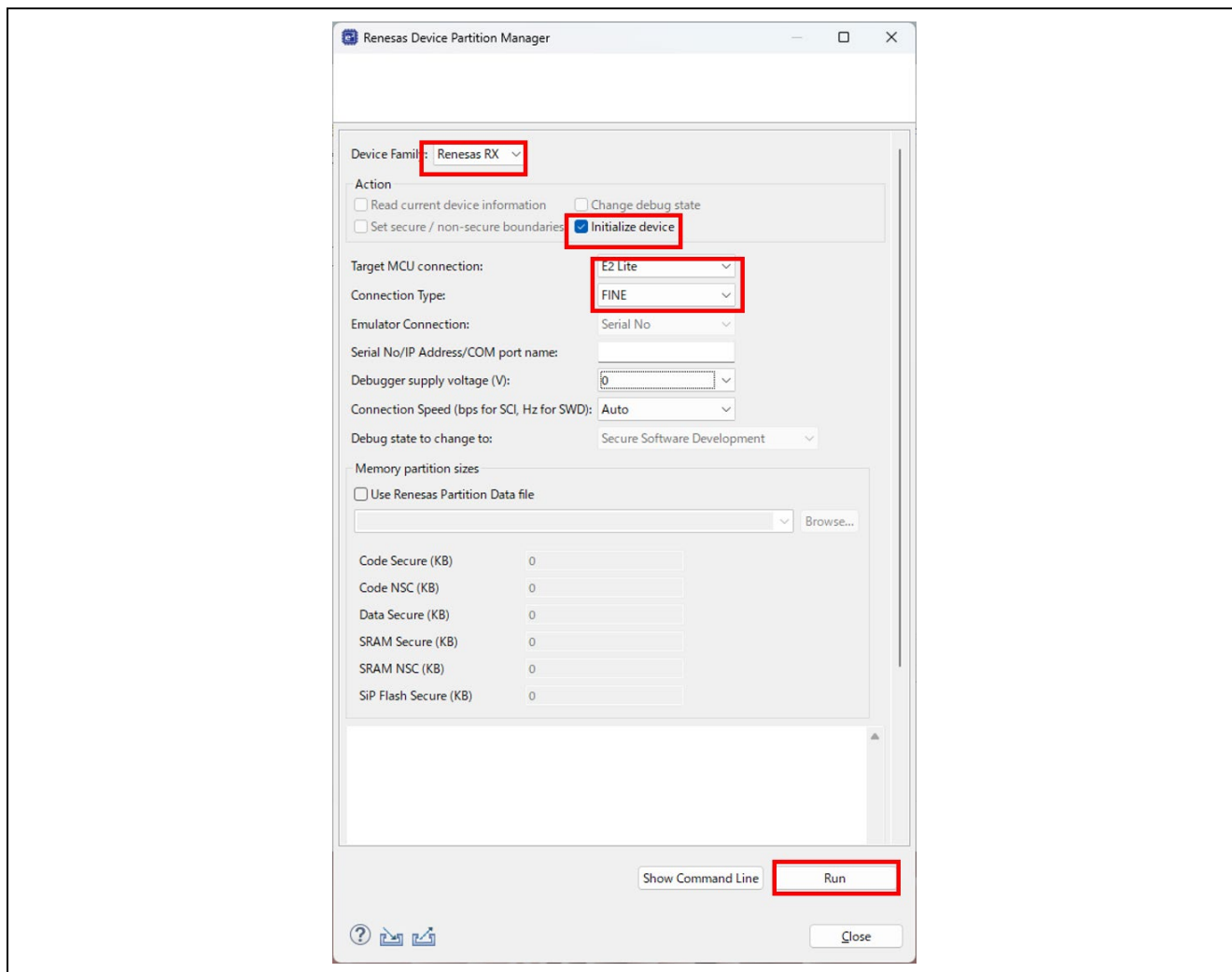


Figure 4-5 Initialize an RX MCU by Using the RDPM

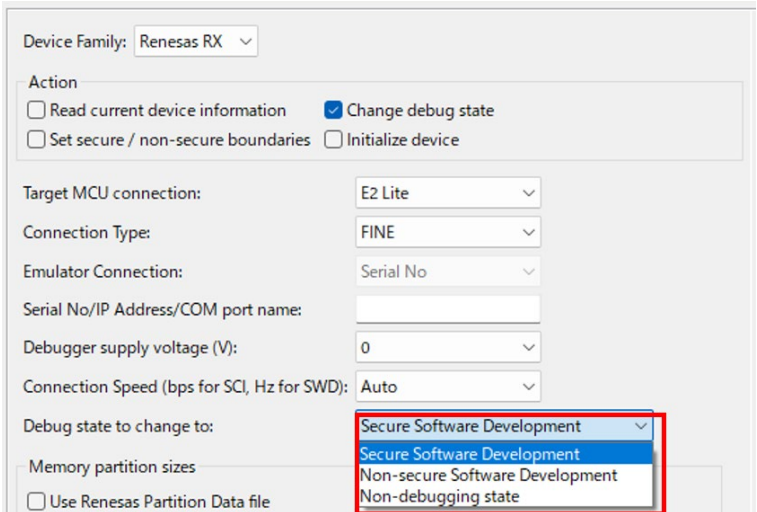
The user generally uses the RDPM for transitions of the protection level from OEM_PL2 to OEM_PL1 or from OEM_PL1 to OEM_PL0, and for **initialization** of the device, which erases all of the device's memory and sets the protection level as OEM_PL2 during product development. The PL determines the corresponding AL.

Restrictions on the RDPM:

- It does not support transitions that require authentication.
- It supports transitions to the device lifecycle state that does not require authentication.
- The operation of the RDPM requires the e² studio IDE.

When the RDPM is used with the RX74M, the following states correspond to each of the protection levels. Select a state for **Debug state to change to** as shown in Figure 4-6.

- | | | |
|--|---|---------|
| • Secure Software Development | → | OEM_PL2 |
| • Non-secure Software Development | → | OEM_PL1 |
| • Non-debugging state | → | OEM_PL0 |



The screenshot displays the RDPM (Remote Device Programming Manager) interface for Renesas RX devices. The 'Device Family' is set to 'Renesas RX'. Under the 'Action' section, the 'Change debug state' checkbox is selected. The 'Debug state to change to:' dropdown menu is open, showing four options: 'Secure Software Development' (highlighted in blue), 'Secure Software Development', 'Non-secure Software Development', and 'Non-debugging state'. A red rectangle highlights the dropdown menu and its options.

Device Family: Renesas RX

Action

☐ Read current device information ☒ Change debug state

☐ Set secure / non-secure boundaries ☐ Initialize device

Target MCU connection: E2 Lite

Connection Type: FINE

Emulator Connection: Serial No

Serial No/IP Address/COM port name:

Debugger supply voltage (V): 0

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

Debug state to change to:

- Secure Software Development
- Secure Software Development
- Non-secure Software Development
- Non-debugging state

Memory partition sizes

☐ Use Renesas Partition Data file

Figure 4-6 Use the RDPM to Change the Device Lifecycle State

4.2 Authenticated Transitions

4.2.1 Authenticated Transitions Using the RFP

If the DLM/PL state is OEM_PL0 and AL1_KEY is injected, it is possible to transit the protection level to OEM_PL1.

The following is an example of some steps in making authenticated transitions.

1. Initialize the MCU to OEM_PL2.
2. Inject AL2_KEY and AL1_KEY.
3. Available state transition sequences:

	State Transition	ALx_KEY to be Used	Example Operation
Case 1	From OEM_PL0 to OEM_PL1	AL1_KEY	Refer to step 4.
Case 2	From OEM_PL1 to OEM_PL2	AL2_KEY	Refer to step 5.
Case 3	From OEM_PL0 to OEM_PL2	AL2_KEY	Refer to step 5.

4. Make a transition from OEM_PL0 to OEM_PL1 by using AL1_KEY. Select **Target Device -> DLM Transition**, specify **OEM_PL1** for **Target DLM State**, and click on **Transition**.

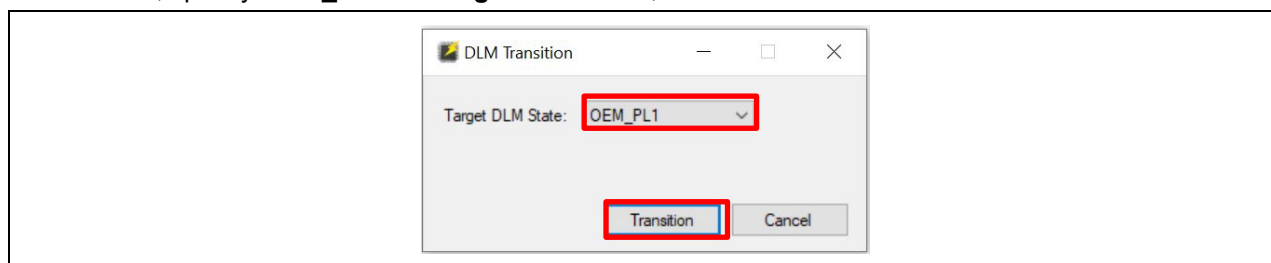


Figure 4-7 Transition from OEM_PL0 to OEM_PL1

Provide the AL1_KEY plaintext data for a transition to the OEM_PL1 state. Specify the plaintext key for **AL1 Key** and click on **OK**. If desired, read the device information out to confirm the resulting state after the transition is finished.

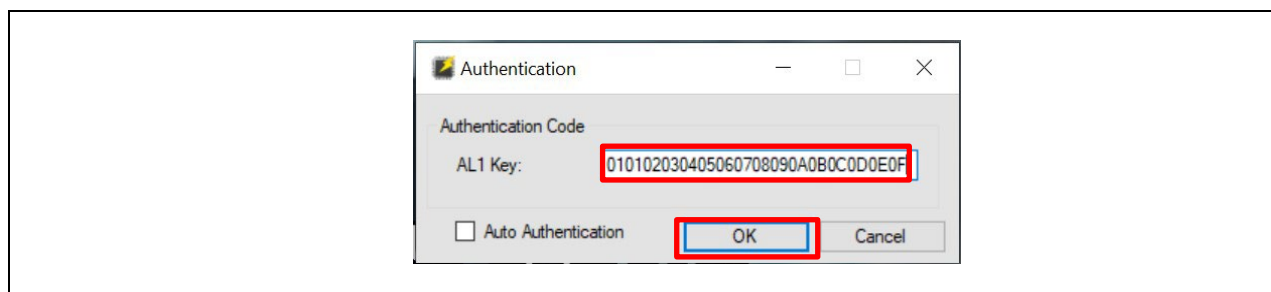


Figure 4-8 Provide AL1_KEY for a Transition to OEM_PL1

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 by using AL2_KEY. Select **Target Device -> DLM Transition**, specify **OEM_PL2** for **Target DLM State**, and click on **Transition**.

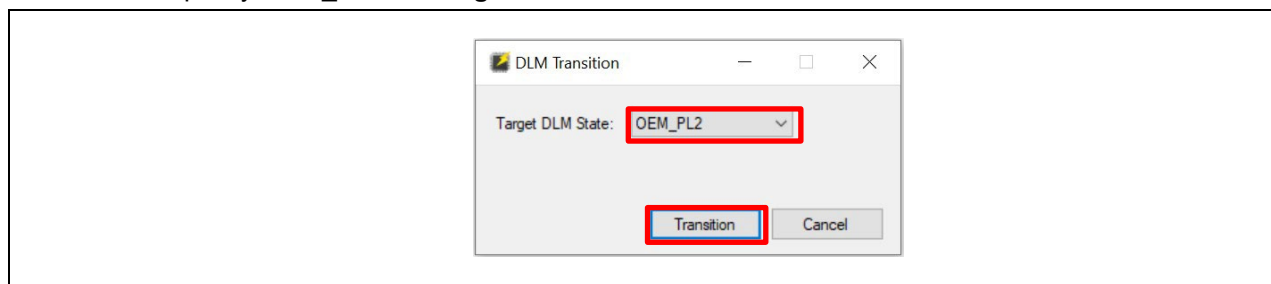


Figure 4-9 Transition from OEM_PL1 to OEM_PL2

Provide the AL2_KEY plaintext data for a transition to the OEM_PL2 state. Specify the plaintext key for **AL2 Key** and click on **OK**. If desired, read the device information out to confirm the resulting state after the transition is finished.

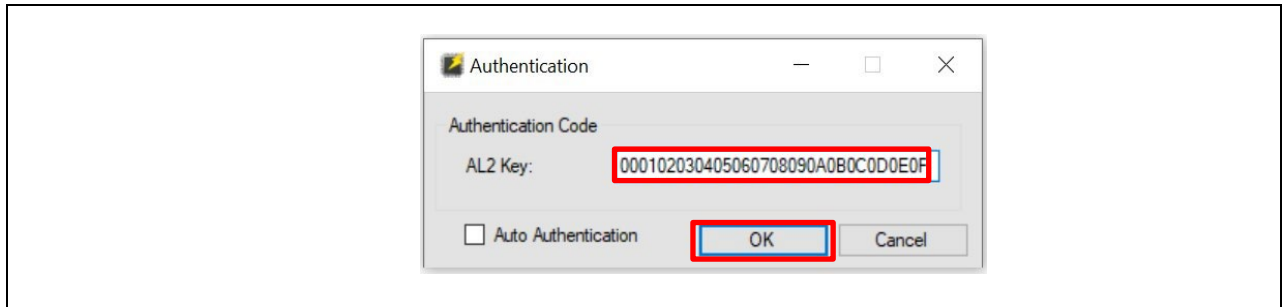


Figure 4-10 Provide AL2_KEY for a Transition to OEM_PL2

4.2.2 Authenticated Transition Using e² studio

When using e² 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 RX74M to OEM_PL2.
2. Inject AL2_KEY and AL1_KEY into the RX74M by using the RFP.
3. Develop secure and non-secure applications by using the development model described in 2.1 or 2.2. In this configuration, when we debug the non-secure application, both the secure and non-secure applications will be programmed, and the security attribution boundary will be set up.
4. Program the secure and non-secure applications and set up the security attribution boundary by debugging the non-secure application or using the RFP.
5. Change the device lifecycle state to OEM_PL1 by using the RFP or RDPM.
6. Update the debug configuration of the non-secure application.
 - Disable the setting of the security attribution boundary.
 - Make a transition to AL2 for debugging purposes by providing the AL2_KEY plaintext data.

Figure 4-11 shows an example of setting a debug configuration by using AL2_KEY which is used as an example in this application note. Specify AL2 for **Authenticate device to Authentication Level (AL)** and the AL2 plaintext key for **Authentication Key** under **Protect Zone** on the **Debugger -> Connection Settings** tab.

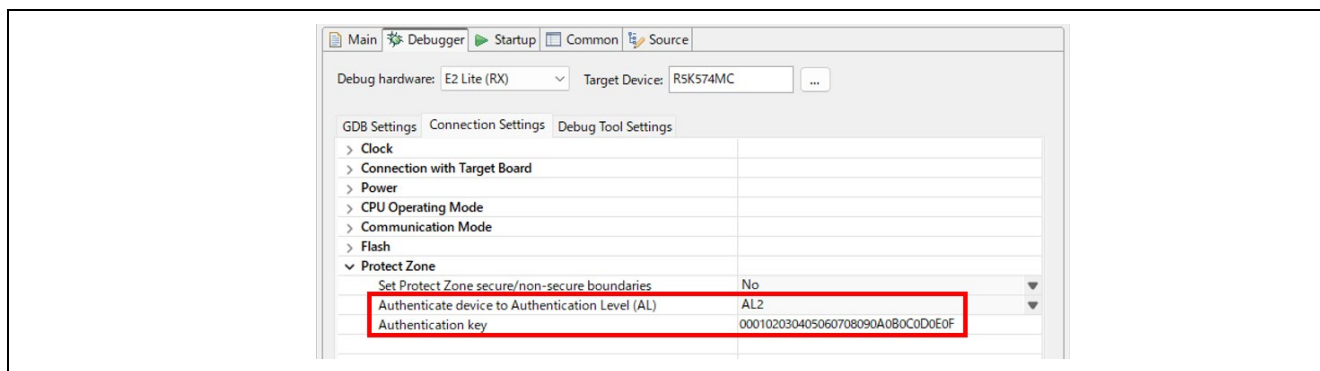


Figure 4-11 Debugging of the Secure Application and Transition to AL2

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

Authenticated Transition from AL0 to AL1

1. Initialize the RX74M to OEM_PL2.
2. Inject AL2_KEY and AL1_KEY into the RX74M.
3. Develop secure and non-secure applications by using the development model described in 2.1 or 2.2. In this configuration, when we debug the non-secure application, both the secure and non-secure applications will be programmed, and the security attribution boundary will be set up.
4. Program the secure and non-secure applications and set up the security attribution boundary by debugging the non-secure application or using the RFP.
5. Change the device lifecycle state to OEM_PL0 using the RDPM or RFP. The authentication level will now be at AL0.
6. Update the debug configuration of the non-secure application project.
 - Download the non-secure application only.
 - Disable the setting of the security attribution boundary.
 - Make a transition to AL1 for debugging purposes by providing the AL1_KEY plaintext data.

Figure 4-12 shows an example of setting a debug configuration by using AL1_KEY which is used as an example in this application note. Specify AL1 for **Authenticate device to Authentication Level (AL)** and the AL1 plaintext key for **Authentication Key** under **Protect Zone** on the **Debugger -> Connection Settings** tab.

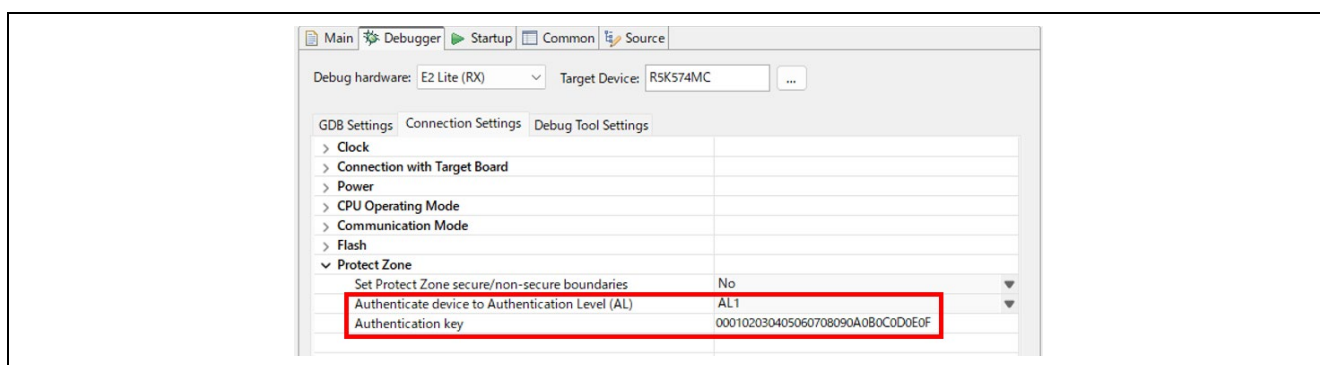


Figure 4-12 Authenticated Transition to AL1 During Debugging

7. Applying the new configuration and debugging the non-secure application should proceed smoothly. The system will authenticate the 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.

5. References

1. Renesas RX74M Group User's Manual: Hardware (R01UH1055)
2. Security Key Management Tool User's Manual (R20UT5349)
3. Renesas Flash Programmer Flash memory programming software User's Manual (R20UT5784)

6. Website and Support

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

RX Product Information	renesas.com/rx
RX Product Support Forum	community.renesas.com/mcu/rx
RX Flexible Software Package (FSP)	renesas.com/rx-fsp
Renesas Support	renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.22.26	—	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/.