Renesas RA Family

Device Lifecycle Management Key Installation

Introduction

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

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

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

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

Required Resources

The following resources are referenced throughout this application note:

Development tools and software

- Renesas Flash Programmer (RFP) v3.08 or later
- e² studio IDE v2020 10 or later
- RA Family Flexible Software Package (FSP) v2.0.0 or later
- SEGGER J-Link® USB driver V6.86 or later

The FSP, J-Link USB drivers, and e² studio are bundled in a downloadable platform installer available on the FSP webpage at renesas.com/ra/fsp.

Hardware

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

Prerequisites and Intended Audience

This application note assumes you have some experience with the Renesas e² studio IDE and Renesas Flash Programmer (RFP). In addition, the application note assumes that you have some knowledge of RA Family MCU security features. See the Security Features section in the Renesas RA6M4 Group MCU User’s Manual: Hardware for background information.

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

1. Introduction to Device Lifecycle Management for RA Family MCU Groups ........................................3
   1.1 Device Lifecycle Management using Debug ID Code .................................................................3
   1.2 Device Lifecycle Management System using Arm® TrustZone® Technology ...............................3
       1.2.1 Definitions of the Device Lifecycle States ........................................................................4
       1.2.2 Summary of the Device Lifecycle States and Transitions ..................................................5
       1.2.3 Advantages of Arm® TrustZone® Enabled Device Lifecycle Management .......................7
       1.2.4 Overview of Device Lifecycle State Management using Renesas Device Partition Manager ....7
       1.2.5 Overview of Device Lifecycle State Transitions using Renesas Flash Programmer ............8

2. Use Cases of Device Lifecycle Management System in Arm® TrustZone®-enabled Application ....9
   2.1 Overview of TrustZone Technology Enabled Development Models ...........................................10
       2.1.1 Split Project Development Model .....................................................................................10
       2.1.2 Combined Project Development Model .............................................................................12
   2.2 Device Lifecycle State Transitions in the Secure Application Development Stage ....................13
   2.3 Device Lifecycle State Transitions in the Non-Secure Application Development Stage .............13
   2.4 Device Lifecycle State Transitions in the Production Flow ......................................................14
   2.5 Possible Device Lifecycle State Transitions Initiated by the End User ......................................15
   2.6 Notes on Flat Project Development Model ............................................................................15

3. DLM Key Creation and Installation Procedure .................................................................16
   3.1 Wrapped Key Installation Overview ......................................................................................16
   3.2 Create Customer PGP Key Pair and Exchange Public Key with Renesas ..................................17
       3.2.1 Overview of Device Lifecycle Management (DLM) Server ..................................................17
       3.2.2 Establish Customer PGP Key Pair ................................................................…………......18
       3.2.3 Registration with DLM Server ............................................................................................21
       3.2.4 PGP Public Key Exchange Between Customer and Renesas .........................................22
       3.2.5 Import Renesas PGP Public Key into Kleopatra ................................................................25
   3.3 Creating the UFPK Using RFP and Wrapped UFPK Using DLM Sever ..................................25
       3.3.1 Create User Factory Programming Key (UFPK) ..............................................................26
       3.3.2 Encrypt UFPK with Renesas PGP Public Key ....................................................................26
       3.3.3 Send UFPK Key to Renesas DLM Server ..........................................................................28
       3.3.4 Receive the Wrapped UFPK Key Encrypted with Customer PGP Public Key ...............29
       3.3.5 Decrypt the Encrypted Wrapped UFPK Key using Customer PGP Private Key ...........29
   3.4 Generate the Encrypted DLM Key using UFPK and WUFPK .................................................30
   3.5 DLM Key Installation .............................................................................................................31
       3.5.1 Install Secure Debug Key ..................................................................................................31
       3.5.2 Install the Non-Secure Debug Key ....................................................................................33
   3.6 Authenticated DLM State Transitions ....................................................................................35
       3.6.1 Authenticated Transition from Non-Secure Debug State to Secure Debug State ...........35
       3.6.2 Authenticated Transition from Deployed State to Non-Secure Debug State .................36
1. Introduction to Device Lifecycle Management for RA Family MCU Groups

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

1.1 Device Lifecycle Management using Debug ID Code

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

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

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

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

1.2 Device Lifecycle Management System using Arm® TrustZone® Technology

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

1.2.1 Definitions of the Device Lifecycle States

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Lifecycle State</th>
<th>Definition and State Features</th>
<th>Debug Level</th>
<th>Serial Programming</th>
<th>Renesas Test Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM</td>
<td>• “Chip Manufacturing”&lt;br&gt;• The device is in Renesas factory.&lt;br&gt;• Developer receives the device in this state.&lt;br&gt;• MCU Unique ID and Hardware Unique Key (HUK) are injected.</td>
<td>DBG2</td>
<td>• Available. &lt;br&gt;• Cannot access code/data flash area.</td>
<td>Not available</td>
</tr>
<tr>
<td>SSD</td>
<td>• “Secure Software Development”&lt;br&gt;• The secure part of the application is being developed.&lt;br&gt;• SECDBG_KEY and RMA_KEY can be injected.</td>
<td>DBG2</td>
<td>• Available. &lt;br&gt;• Can program/erase/read all code/data flash areas.</td>
<td>Not available</td>
</tr>
<tr>
<td>NSECSD</td>
<td>• “Non-SECure Software Development”&lt;br&gt;• The non-secure part of the application is being developed.&lt;br&gt;• NONSECDBG_KEY can be injected.&lt;br&gt;• It is possible to regress to SSD state without flash erase if SECDBG_KEY is injected in SSD state.&lt;br&gt;• It is possible to erase the entire flash to SSD state.</td>
<td>DBG1</td>
<td>• Available. &lt;br&gt;• Can program/erase/read non-secure code/data flash areas.</td>
<td>Not available</td>
</tr>
</tbody>
</table>
### Lifecycle State

<table>
<thead>
<tr>
<th>Lifecycle State</th>
<th>Definition and State Features</th>
<th>Debug Level</th>
<th>Serial Programming</th>
<th>Renesas Test Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPL</td>
<td>“DePLoyed”</td>
<td>DBG0</td>
<td>• Available.</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>• The device is in the field.</td>
<td></td>
<td>• Cannot access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• It is possible to regress to</td>
<td></td>
<td>code/data flash</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NSECSD state without flash</td>
<td></td>
<td>areas.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>erase if NONSECDBG_KEY is</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>injected in NSECSD state.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• It is possible to erase the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>entire flash to SSD state.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCK_DBG</td>
<td>“LoCKed DeBuG”</td>
<td>DBG0</td>
<td>• Available.</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>• The debug interface is</td>
<td></td>
<td>• Cannot access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>permanently disabled.</td>
<td></td>
<td>code/data flash</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>areas.</td>
<td></td>
</tr>
<tr>
<td>LCK_BOOT</td>
<td>“LoCKed BOOT interface”</td>
<td>DBG0</td>
<td>• Not available.</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>• The debug interface and the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>serial programming interface</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>are permanently disabled.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMA_REQ</td>
<td>“Return Material Authorization</td>
<td>DBG0</td>
<td>• Available.</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td>REQUEST”</td>
<td></td>
<td>• Cannot access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Request for RMA.</td>
<td></td>
<td>code/data flash</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The customer must send the</td>
<td></td>
<td>areas.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>device to Renesas in this</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>state.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMA_ACK</td>
<td>“Return Material Authorization</td>
<td>DBG2</td>
<td>• Available.</td>
<td>Available</td>
</tr>
<tr>
<td></td>
<td>ACKnowledged”</td>
<td></td>
<td>• cannot access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Failure analysis by Renesas.</td>
<td></td>
<td>code/data flash</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>areas.</td>
<td></td>
</tr>
</tbody>
</table>

### 1.2.2 Summary of the Device Lifecycle States and Transitions

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

![Figure 2. RA MCU Device Lifecycles](image)

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

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

DLM Authentication Keys

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

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

Authenticated Transitions

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

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

For more details on the use cases of device lifecycle transitions, please refer to section 2.
1.2.3 Advantages of Arm® TrustZone® Enabled Device Lifecycle Management

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

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

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

Renesas Device Partition Manager is a utility integrated with e² studio for Device Lifecycle State management during production development. A user can use the Renesas Device Partition Manager to perform the following functions:

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

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

Figure 3. Open the Renesas Device Partition Manager

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

Figure 4. Initialize RA6M4 Using Renesas Device Partition Manager
During Product Development, users typically use the Renesas Device Partition Manager to advance the device lifecycle:

- SSD -> NSECSD
- NSECSD -> DPL

Limitations of the Renesas Device Partition Manager

- Renesas Device Partition Manager does not support authenticated transitions. For failure analysis, use RFP.
- Renesas Device Partition Manager supports transitions to limited device lifecycle states and requires e² studio IDE environment to operate. For mass production where transitioning to LCK_DBG or LCK_BOOT is required, the user should use RFP.

1.2.5 Overview of Device Lifecycle State Transitions using Renesas Flash Programmer

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

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

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

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

- Flat Project Development Model
  — No TrustZone technology awareness during development
  — Benefits from the Device Lifecycle Management during production stage
- Split Project Development Model
  — Uses both Secure and Non-secure project types
  — This development model allows product development by two teams.
Secure Developers - create Root of Trust (5.1) or an isolated subsystem. Development is carried out in SSD state.

Non-secure Developers - create application that is built on the Root of Trust and uses the isolated subsystem. Development is carried out in NSECSD state.

Secure system is locked from access from the non-secure developers.

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

![Figure 9. Renesas RA Arm® TrustZone® Technology Enabled Project Types](image)

### 2.1 Overview of TrustZone Technology Enabled Development Models

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

#### 2.1.1 Split Project Development Model

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

- At one of the final steps during the MCU manufacturing, Renesas will inject an MCU Unique ID and HUK to the MCU. The MCU is then delivered to the Secure Developer.
- The Secure Product Developer develops the secure application and locks the secure region from access by the Non-secure region. The MCU is then delivered to the non-secure developer.
- The Non-secure Product Developer develops the non-secure application. In this stage, the secure system is not visible to the non-secure product developer.
- The End Product User may receive the MCU with the debug interface temporarily locked or permanently locked. The serial programming interface may be permanently locked or available (but with limited functionality).
Figure 10. Split Project Development Model

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

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

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

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

![Figure 12. Combined Project Development Model](image)

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

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

![Figure 13. Device Lifecycle States in Combined Project Development Model](image)
2.2 Device Lifecycle State Transitions in the Secure Application Development Stage

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

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

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

- Prior to start of secure application development:
  
  **CM -> SSD**: no authentication needed, no flash erase.

- During development, the secure developer can execute the `All erase` command to recover the MCU. The resulting Device Lifecycle State is still SSD.
  
  **SSD -> SSD**: no authentication, erase entire flash except permanently locked block.

  Note that the secure developer can use RFP to disable the `All erase` command in SSD state.

- After injecting the RMA_KEY during SSD lifecycle state, the secure developer can use the RMA_KEY to change the lifecycle of the MCU to RMA_REQ state after successful challenge response authentication.
  
  **SSD -> RMA_REQ**: authentication using RMA_KEY, entire flash erased except permanently locked block.

- After the secure application is developed and tested:
  
  **SSD -> NSECSD**: no authentication, no flash erase.

If the development follows the Combined Project Development Model, the product developer will finish both secure and non-secure application development in SSD state.

Device Lifecycle State Regression from NSECSD to SSD with Authentication

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

- The secure developer can use SECDBG_KEY to change the MCU state from NSECSD to SSD and perform the corresponding debug or failure analysis.
  
  **NSECSD -> SSD**: Authentication with SECDBG_KEY, no flash erase.

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

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

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

Device Lifecycle State Change from NSECSD to SSD without Authentication

- If there is need to erase the entire flash to recover the MCU, the non-secure developer can erase the entire flash to regress the MCU lifecycle state from NSECSD to SSD. But the non-secure developer needs to return the MCU to the secure developer to reprogram the secure application.

  **NSECSD -> SSD**: erase entire flash

  A non-secure developer can use RFP to disable the `All erase` command in SSD state.

- This transition is typically not needed for the Combined Development Model as the Development is typically carried out in SSD for both Secure and Non-Secure Projects.
Device Lifecycle State Regression from DPL to NSECSD with Authentication

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

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

Device Lifecycle State Change from NSECSD to DPL without Authentication

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

  **NSECSD -> DPL**: no authentication, no flash erase

2.4 Device Lifecycle State Transitions in the Production Flow

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

Split Project Development Model

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

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

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

Combined Project Development Model

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

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

Flat Project Development Model

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

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

**Case 1: End Product is in DPL State**

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

**Transiting from DPL to Renesas Return Material Request (RMA) State**

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

**Note on Transition to RMA_REQ**

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

**Case 2: End Product is in LCK_DBG State**

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

**Case 3: End Product is in LCK_BOOT State**

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

2.6 Notes on Flat Project Development Model

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

3.1 Wrapped Key Installation Overview

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

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

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

![Figure 14. Wrapping the User Factory Programming Key (UFPK)](image)

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

![Figure 15. Encrypt the DLM Key Using UFPK](image)

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

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

![Figure 16. Created Wrapped DLM Keys](image)
3.2 Create Customer PGP Key Pair and Exchange Public Key with Renesas

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

3.2.1 Overview of Device Lifecycle Management (DLM) Server

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

![Figure 17. Operational Flow of Using Key Wrapping Using DLM Server](image)

DLM Server FAQ and User’s Manual

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

![Figure 18. DLM Server FAQ and User’s Manual](image)
DLM Server FAQ

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

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

![Overview of DLM Key Wrapping Service Using PGP](image)

3.2.2 Establish Customer PGP Key Pair

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

1. Download PGP Software from the following website

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

![Download GPG Software](image)
3. Click **File > New Key Pair** and choose format **Create a personal OpenPGP key pair**.

![Figure 21. Generate PGP Key Pair](image)

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

![Figure 22. Advanced Settings for the PGP Key pair](image)

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

![Figure 23. Provide Passphrase](image)
6. Observe that the PGP Key Pair is created successfully.

![Figure 24. Key Pair Successfully Created](image)

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

![Figure 25. Export the Customer Public key](image)

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

![Figure 26. Save the Customer Public key](image)
3.2.3 Registration with DLM Server

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

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

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

**Figure 27. Provide Email Address for New Registration**

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

```
Dear customer,

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

https://dlm.renesas.com/keywrap/?menu=reg%2Fregist%2Ffixed%2F&param=9id%3D0u12%7C0v3Cd7ODqJEy%252B0mr6X7ph6v8h9mD%252F1e07a88%253D

- Notes
  - Please register your information within three hours of receiving this e-mail.
  - The URL expires after the three hours, after which you will need to start the registration process anew.
  - Please delete this email if you were not aware that you were going to receive it.

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

Thank you.
Renesas Electronics Corporation
```

**Figure 28. Email with Registration Link**
Click on the URL in the confirmation mail and provide your name, company name, password, and re-enter the password. Click **Next (confirmation)** button. After the confirmation screen is displayed, click on the **Registration** button to complete the user registration.

![Register Customer Information](image)

**Figure 29. Register Customer Information**

### 3.2.4 PGP Public Key Exchange Between Customer and Renesas

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

![Start Using the DLM Service](image)

**Figure 30. Start Using the DLM Service**

Accept the **Trusted Secure IP Key Wrap Agreement** as shown below by clicking **Agree**. Note that the following Agreement will come up every time the customer logs into the DLM server.
--- CAUTION!! ---

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

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

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

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

The End

Trusted Secure IP Key Wrap Agreement

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

........

Article 15 (ENTIRE AGREEMENT)

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

[Links to agree or disagree]

Figure 31. Agree with the Key Wrap Agreement

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

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

Figure 32. Request for PGP Key Exchange
Click the **PGP key exchange** button. The user interface shown in Figure 33 will open. Click **Reference** as shown in Figure 33 and select the customer public key exported from section 3.2.2 as shown in Figure 26 (public_key.asc). Next, click **PGP key exchange** in the interface shown in Figure 33 and wait to receive the Renesas PGP public key from the customer’s email address provided at the New user registration stage.

**Figure 33. Provide Customer PGP Public Key to DLM Server**

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

**Figure 34. Receive the Renesas PGP Public Key**

Save the Renesas PGP public key received (keywrap-pub.key). This key will be used in the following sections.
3.2.5 Import Renesas PGP Public Key into Kleopatra

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

![Import Renesas Public Key into Kleopatra](image)

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

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

![Renesas PGP Public Key](image)

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

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

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

- The UFPK is wrapped by the Renesas DLM Server: [https://dlm.renesas.com/](https://dlm.renesas.com/). This webpage communicates with the user through HTTPS and PGP, thus offering security for protecting user data in transit.
- The DLM Key is wrapped when RFP installs the DLM Keys in the MCU.
3.3.1 Create User Factory Programming Key (UFPK)
We can use the `rfp-util.exe` included with RFP installation folder to generate UFPK. Open a command line window and activate the `rfp-util.exe` with an example shown as follows.

**Example Command Line Input:**
```
rfp-util /genufpk /output "C:\DLM_Key_Installation\test\ufpk.key"
```

![Figure 37. Generate User Factory Programming Key](image)

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

![Figure 38. Select the UFPK Key to be Encrypted](image)
Choose **Encrypt for others** and select the Renesas PGP Public key. Click **Sign/Encrypt**.

![Sign/Encrypt Files](image)

**Figure 39. Use Renesas Public Key to Encrypt UFPK**

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

![Encrypt-To-Self Warning](image)

**Figure 40. Confirm Encryption Option**

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

![Encrypt the UFPK Key](image)

**Figure 41. Encrypt the UFPK Key with Renesas Public Key**
3.3.3 Send UFPK Key to Renesas DLM Server

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

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

![Select RA Device](image)

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

![Send Encrypted UFPK to DLM Server](image)

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

![Message from DLM Server](image)
3.3.4 Receive the Wrapped UFPK Key Encrypted with Customer PGP Public Key

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

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

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

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

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

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

Figure 46. Decrypt with Customer PGP Private Key

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

Figure 47. Decrypt with Customer PGP Private Key
3.4 Generate the Encrypted DLM Key using UFPK and WUFPK

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

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

Example Command Line Input:

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

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

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

![Figure 48. Use rfp-util.exe to Generate DLM Key SECDBG_KEY](image)

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

```
rfp-util /genkey /ufpk "C:\DLM_Key_Installation\test\ufpk.key" /wufpk "010102030405060708090A0B0C0D0E0F" /output "C:\DLM_Key_Installation\test\NONSECDBG.rkey"
```

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

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

![Figure 49. Use rfp-util.exe to Generate DLM Key NONSECDBG_KEY](image)

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

```
rfp-util /genkey /ufpk "C:\DLM_Key_Installation\test\ufpk.key" /wufpk "020102030405060708090A0B0C0D0E0F" /output "C:\DLM_Key_Installation\test\RMA.rkey"
```
3.5 DLM Key Installation

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

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

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

3.5.1 Install Secure Debug Key

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

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

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

Unzip ra6m4_dlm_key_install.rfp.zip to reveal ra6m4_dlm_key_install.rfp.rpj. Launch RFP and open project ra6m4_dlm_key_install.rfp.rpj and review the settings explained in this section.
For the Flash Options, use the Set Option to set up Set. Now, highlight Encrypted SECDBG Key and then click the “…” on the right to select the corresponding SECDBG_KEY.

**Figure 51. Select the SECDBG_Key to install**

Select the SECDBG.rkey generated in section 3.4.

**Figure 52. Select the SECDBG_KEY to Install**

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

**Figure 53. Select Program and Verify Flash Option Setting**
RFP operation for setting the DLM keys can be done at the same time as programming a valid binary file selected in the Operation tab. Downloading with a binary is optional.

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

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

![Figure 54. Install the SECDBG_KEY](image)

**3.5.2 Install the Non-Secure Debug Key**

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

![Figure 55. Transition the MCU Device Lifecycle State to NSECSD](image)

Next, follow steps similar to section 3.5.1 to install the Non-Secure Debug Key. In this example, we can use the NONSECDBG_KEY (NONSECDBG.rkey) generated with Figure 49 to illustrate the operation.
Note that the user needs to delete the SECDBG key file entry and add the NONSECDBG key file entry as shown below.

![Figure 56. Select the NONSECDBG.rkey to Install](image)

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

As explained previously, for testing purpose, the `bare_metal_minimal_ns.srec` binary is included with this application note for user’s convenience.
This section provides operational steps for authenticated DLM state transitions. The assumption is that SECDBG_KEY and NONSECDBG_KEY are already installed using the steps discussed previously.

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

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

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

Assume that the current Device Lifecycle State is NSECSD and that SECDBG_KEY has been previously installed. User can read out the **Device Information** to confirm this.
Use the following steps to regress the Device Lifecycle State to SSD:

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

   ![Figure 59. Select Regress MCU Device Lifecycle State back to SSD](image)

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

   ![Figure 60. Provide the Authentication Key for SECDBG](image)

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

   ![Figure 61. Confirm Device Lifecycle Transitions Back to SSD](image)

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

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

![Figure 62. Confirm the Device Lifecycle State and DLM Key status](image)
Use the following steps to regress the Device Lifecycle State to NSECSD.

1. Select transition to NSECSD.

![Figure 63. Select Transition to NSECSD](image)

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

![Figure 64. Provide the authenticate key for NONSECDBG_KEY](image)

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

![Figure 65. Confirm Device Lifecycle has Transitioned Back to NSECSD](image)

4. References

Available on renesas.com:

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

5. Appendix

5.1 Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCE9</td>
<td>Secure Crypto Engine 9 is a hardware unit which resides on Renesas Arm® Cortex®-M33 MCU</td>
</tr>
<tr>
<td>Device Certificate</td>
<td>Certificate uniquely identifying an individual device. It is digitally signed, asserting that the certificate comes from a known source and has not been modified, and that the device is trusted.</td>
</tr>
<tr>
<td>Term</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Root of Trust</td>
<td>Roots of trust are highly-reliable hardware, firmware, and software components that perform specific, critical security functions. (<a href="https://csrc.nist.gov/projects/hardware-roots-of-trust">https://csrc.nist.gov/projects/hardware-roots-of-trust</a>)</td>
</tr>
<tr>
<td>SCE</td>
<td>Secure Crypto Engine – A module in the MCU that provides for efficient, low-power cryptographic acceleration, TRNG (True Random Number Generation), and creation and isolation of cryptographic keys.</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure – A set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates, which are typically used to manage secure identity via public key cryptography.</td>
</tr>
<tr>
<td>Key Pair</td>
<td>Asymmetric keys are generated in pairs – a public and private key. The private key is held in secret by only one party and can be used to assert that party’s identity. The public key is freely distributed and is uniquely associated with the private key.</td>
</tr>
<tr>
<td>Secure Code</td>
<td>A function or group of functions that resides in a secure region of internal flash, as defined and enforced by the MPUs. These secure functions can access both secure data and non-secure data regions.</td>
</tr>
<tr>
<td>Non-Secure code</td>
<td>A function or group of functions that resides in a non-secure region of internal flash. These non-secure codes cannot access the secure region. They can only access the non-secure region.</td>
</tr>
<tr>
<td>HUK</td>
<td>Hardware Unique Key. This is a unique key stored inside the RA Family MCU.</td>
</tr>
<tr>
<td>Challenge String</td>
<td>Randomly generated string at the host application. This string is used by the host application to validate the ownership of the private key by the target.</td>
</tr>
<tr>
<td>Unique ID</td>
<td>An identification value, unique to each individual RA Family MCU, that is stored inside the MCU. The unique ID is used by the SCE when it wraps a key.</td>
</tr>
<tr>
<td>Challenge Response String</td>
<td>The response to the challenge string. The Challenge Response String is the signature of the challenge data as created by signing the Challenge String with the receiver’s private key.</td>
</tr>
</tbody>
</table>
Website and Support

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

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

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Oct.01.20</td>
<td>—</td>
<td>First release document</td>
</tr>
<tr>
<td>1.10</td>
<td>Dec.09.20</td>
<td>—</td>
<td>Updated importing Renesas Public Key to Kleopatra procedure</td>
</tr>
<tr>
<td>1.11</td>
<td>Apr.30.21</td>
<td>—</td>
<td>Improve wording.</td>
</tr>
</tbody>
</table>
Notice

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

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

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

4. You shall 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 systems; key plants; military 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 implants; etc.), or may cause serious property damage (space system; underwater repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

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 WILL BE INVulnerable OR FREE FROM CORRUPTION, ATTACK, VIRUS, INTERERENCE, 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.

(Nota1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Nota2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

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

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.