# GigaDevice Semiconductor Inc.

# GD32H7xx Secure Memory Management

# Application Note
# AN113

# Table of Contents

# List of Figures

# List of Tables

# 1. Introduction

Besides security protection ( SPC ), erase/program protection ( WP ) and execute-only dedicated code read protection area ( DCRP ), GD32H7xx series also provides algorithm encryption and decryption scheme and secure mode to protection code and data. Secure mode can support secure boot, licensed firmware update ( LFU ), as well as user-defined secure application development.

# 2. Encryption and decryption scheme

GD32H7xx series provides AES (128) algorithm encryption and decryption scheme for on-chip flash and external OSPI flash.

On-chip flash and external OSPI flash have similar protection mechanism, encryption is implemented by software tool, decryption is implemented by hardware. For encryption, user are required to use software tools provided by Gigadevice to perform external encryption before writing code or data.

Decryption is disabled by default.

For on-chip flash, user can enable AES decryption function by set AESEN bit in efuse's user control parameter. User can modify the high 96 bits of the initial vector by modifying the FMC_AESIVx_MDF register and AES key is also configured by efuse. After the configuration is complete, the hardware decrypts automatically when reading code or data. User do not need to participate in the decryption process.

**Note:** FMC_NODEC register can be configured to partition a non-decrypting area. When this area is valid, it will not decrypt even if the AESEN bit is 1.

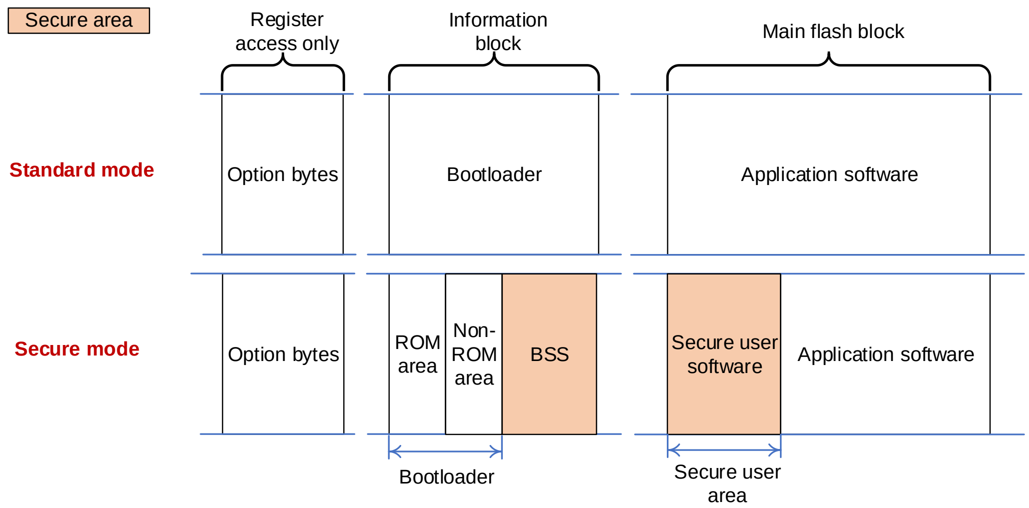
For external OSPI flash, user can decrypt through RTDEC module. After the configuration is complete, the hardware decrypts automatically when reading code or data. User do not need to participate in the decryption process. For more information, please refer to ***AN122 GD32H7xx OSPI Flash Execution Environment User Guide***.

# 3. Secure mode

Security should be performed for some sensitive programs to avoid potentially malware attacks. For example, licensed firmware update software requires highly protection because it processes confidential data ( such as encryption keys ) that cannot get by other processes.

Secure areas with limited access is provided. In this area, secure services can be built that can be executed before any user application. These secure areas and their included software can be accessed only in secure mode. ***Figure 3-1. Memory architecture in standard mode and secure mode*** shows the details of the area.

**Figure 3-1. Memory architecture in standard mode and secure mode**



Secure user area is accessed once after reset, the area code is hidden after execution. Basic security service is Gigadevice software to configure secure services. Secure user software is located in secure user area and executed once after reset. Secure user software can be used to implement secure boot and LFU.

Secure mode and secure user area can be configured by option bytes or efuse. User can set the secure user area in the option byte by BSS to make the secure code and data be configured in the secure user area.

## 3.1. Secure boot flow

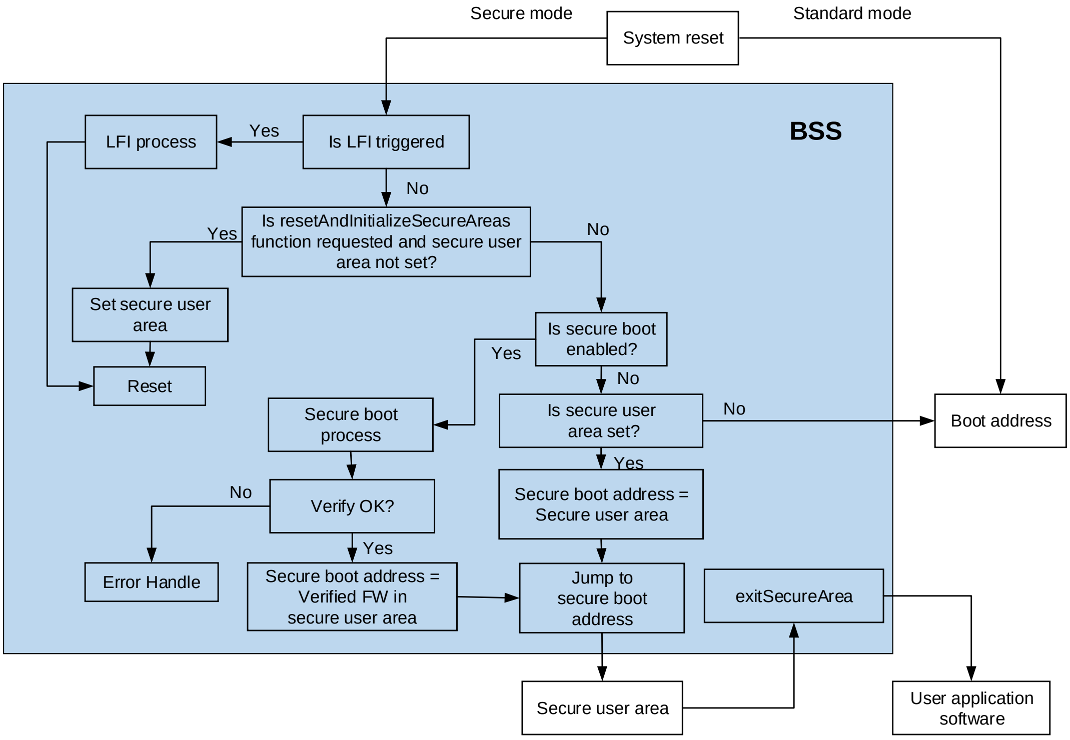
In secure mode, secure firmware store in information block to support boot. Secure user software is secure application code, data, or algorithms stored in main flash block.

If secure user area is not set, MCU will jump to the requested boot address which set by the BOOT_ADDR0[15:0] option bits in FMC_BTADDR_EFT register.

In secure mode, regardless of the startup configuration ( BOOT pins and boot addresses ), MCU will be forced to boot from secure ROM area.

The specific boot process is shown in *Figure 3-2. Secure boot flow*.

**Figure 3-2. Secure boot flow**



For more information on licensed firmware installation, please refer to the *AN118 GD32H7xx Licensed Firmware Install (LFI) Overview*.

For more information on licensed firmware update, please refer to the *AN119 GD32H7xx Licensed Firmware Update (LFU) Overview*.

For more information on secure boot, please refer to *AN120 GD32H7xx Secure Boot Overview*.

## 3.2. Secure service configure

Secure mode is enabled as long as one of the SCR bits in the option byte or efuse is set to 1. At this time, the SCR bit of the FMC_OBSTAT0_EFT register is 1. And after the SCR bit is enabled, a system reset is required to activate secure mode.

Unlike the efuse configuration, the SCR bit in the option byte can be modified. If there is no valid secure user area and no valid DCRP area, SCR bit in the option byte can be reset freely. If a valid DCRP area or a valid secure user area exist, the only way to reset SCR bit in the option byte is: to perform a SPC level low to no protection demotion when DCRP_EREN ( in FMC_DCRPADDR_EFT or FMC_DCRPADDR_MDF register ) is set to 1 and SCR_EREN ( in FMC_SCRADDR_EFT or FMC_SCRADDR_MDF register ) is set to 1 .Otherwise OBMERR flag is set.

Standard mode can be returned when the SCR bit in option byte is 1 and the SCR bit in efuse is 0. To return to standard mode, the secure user area and DCRP area need to be removed before or at the same time as the SCR option bit is cleared. For more details, please refer to the modify rules of the relevant option bytes.

# 4. Secure user software

## 4.1. Access authority

Secure user software is stored in secure user area and can only be accessed in secure mode. Secure user software is the secure code that is called and executed by the secure bootloader after the reset. After the system is reset, the secure user software builds the contents of the user safe startup, software license check, secure firmware update, and secure initialization.

After the secure user software is executed, if the code jump to the user's main application (non-secure), the content of the secure user area cannot be accessed. Before exiting secure area, user must call the exitSecureAreas security function to Implement jumping, with one parameter being the address of the main application to jump.

As soon as entering the application code and attempt to access the secure user area, a read protection error (RSERR) is set and the current read operation is aborted.

## 4.2. Secure user area configure

One secure user area can be defined by setting the SCR_AREA_END and SCR_AREA_START option bytes with a granularity of 4 Kbytes. This means that the actual secure user area size is defined by:

Secure user area size = [( SCR_AREA_END[10:0] - SCR_AREA_START[10:0] ) + 1] x 4Kbytes

If SCR_AREA_END[10:0] = SCR_AREA_START[10:0], whole main flash block is secure user area.

If SCR_AREA_END[10:0] < SCR_AREA_START[10:0], protection is invalid.

Besides the option bytes, the secure user area can also be configured by modifying the user control parameter in efuse macro with granularity of 32KB bytes.

**Note:** The secure configuration priority of efuse is higher than the flash option byte, so in the product, if configure the secure user area by the option byte, the start and end address of the secure user area in efuse should be both set to 0, and the SCRLK bit in efuse should be set to 1, otherwise there may be vulnerability in the secure user area.

When a secure user area is configured by option bytes and its area address is valid, the code in the secure user area can update the size of the secure user area.

# 5. Basic security service

BSS provides the secure area setting function and secure area exiting function.

## 5.1. Secure area setting function

Secure area setting function is provided by Gigadevice to perform the initialization of secure user area. In standard mode, user can directly call function (resetAndInitializeSecureAreas) to set the secure area, and other basic security services are not allowed to access.

The description of ***Table 5-1. Function resetAndInitializeSecureAreas*** is shown as below:

**Table 5-1. Function resetAndInitializeSecureAreas**

| **Function name** | resetAndInitializeSecureAreas |
|---|---|
| **Function prototype** | void resetAndInitializeSecureAreas(BSS_secure_area_struct area); |
| **Function descriptions** | Set the range of the secure user area based on the SCR_AREA_START and SCR_AREA_END option bytes. |
| **Precondition** | - |
| **The called functions** | - |
| **Input parameter{in}** | |
| **area** | secure user area start address and end address |
| **Output parameter{out}** | |
| **-** | - |
| **Return value** | |
| **-** | - |

**Note:** After the function is completed, a system reset is generated. This function is available only when the secure user area is first set up. User must ensure that the correct secure programs is exist in the target secure area to make it can exit to the standard programs, otherwise it will cause chip scrap.

## 5.2. Secure area exiting function

Gigadevice provides a function ( exitSecureArea ) to exit from secure user software and jump to user application. It can close secure user area to ensure that the content in secure user area is no longer accessed.

The description of ***Table 5-2. Function exitSecureArea*** is shown as below:

**Table 5-2. Function exitSecureArea**

| **Function name** | exitSecureArea |
|---|---|
| **Function prototype** | void exitSecureArea(unsigned int vectors, unsigned int jtagState); |
| **Function descriptions** | exit from the secure user area and jump to the user application. |

| | |
|---|---|
| **Precondition** | - |
| **The called functions** | - |
| **Input parameter{in}** | |
| **vectors** | address of application vector to jump after exit secure user area. |
| **Input parameter{in}** | |
| jtag_state | status of the JTAG after exit secure user area. |
| *BSS_EXIT_SCR_JTAG_ENABLE* | enable the JTAG after exiting |
| *BSS_EXIT_SCR_JTAG_DISABLE* | Disable the JTAG after exiting |
| **Output parameter{out}** | |
| **-** | - |
| **Return value** | |
| **-** | - |

**Note:** After the function is completed, no system reset is generated. For security reasons, users should disable cache before calling this function in a secure user area.

# 6. Flash protection summary

*Figure 6-1. Flash protection areas* show the relationships among protected areas.

**Figure 6-1. Flash protection areas**

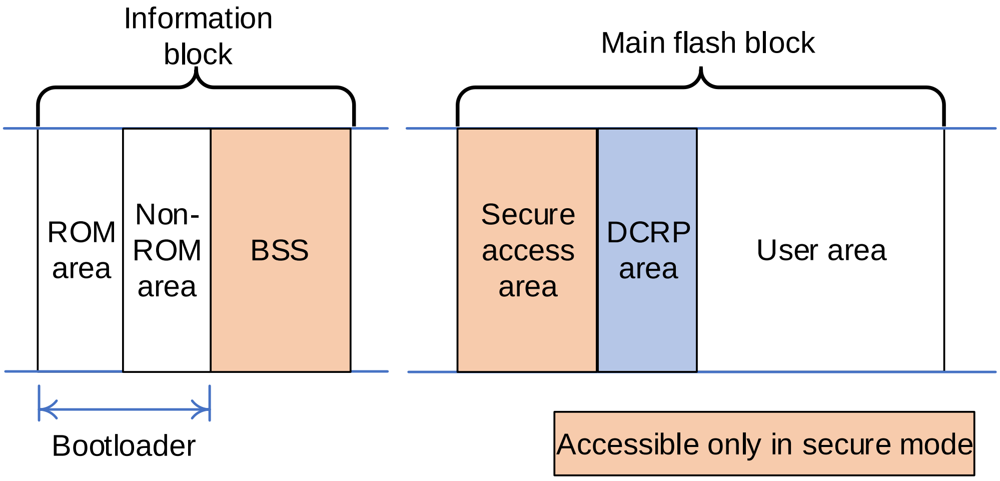


*Table 6-1. Secure memory protection access authority* summarizes the access rights of each protected area.

**Table 6-1. Secure memory protection access authority**

| Area | Mode | Operation | Accessible |
|---|---|---|---|
| DCRP | Standard mode / secure mode | Execution | Yes |
| | | Read | No |
| | | Debug | No |
| Secure user area | Secure mode | Execution | Has permission after reset until the code has finished executing |
| | | Read | Has permission after reset until the code has finished executing |
| | | Debug | No |
| BSS | Secure mode | Execution | Has permission after reset until the code has finished executing |
| | | Read | Has permission after reset until the code has finished executing |
| | | Debug | No |

# 7. Revision history

Table 7-1. Revision history

| Revision No. | Description | Date |
|---|---|---|
| 1.0 | Initial Release | Apr.18, 2023 |

## Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.