GigaDevice Semiconductor Inc.

GD32H73x_75x Security Introduction

Application Note
AN121

Revision 1.2

(Feb. 2026)

Table of Contents

List of Figures

List of Tables

1. Introduction

The GD32H73x_75x series MCU has a strong security architecture while providing high performance.

There are security resources for the whole life cycle management, including hardware, firmware resources in the chip and related software tools developed by Gigadevice.

This document helps user fully understand security architecture of the GD32H73x_75x series MCU, including secure features, secure install and update solution, secure boot and secure mode.

Table 1-1. Applicable product

Product series	Model
GD32H73x	GD32H737 series
GD32H75x	GD32H757, GD32H759, GD32H75E series

Note: This application note is for reference only. In case of any conflict with the user manual or datasheet, the user manual or datasheet shall prevail.

2. Secure features

GD32H73x_75x series MCU provides many secure features, some are secure peripherals, like CRC, TRNG, HAU, CAU, which are hardware accelerator for general encryption algorithm. Some are secure mechanism, like secure mode, RDP, PCROP, WRP. Others are secure firmware,like basic secure services (BSS), immutable secure boot code.

Internal flash is protected with FMC AES cryptography algorithms, RDP, PCROP, WRP, and the RTDEC for external OSPI memory.

These features all contribute to a very secure system.

2.1. Memory protection unit (MPU)

The MPU is a feature provided by Cortex-M of ARM, it allows specific access rights of all memory resource, this protection is dynamically managed by CPU at runtime. The MPU splits memory address space to several regions, and access right of each region can be set independently. E.g. Executable, not executable(XN), read-write (RW), read only (RO) or no access. Privileged and unprivileged modes.

2.2. Security protection (SPC)

The SPC is useful for protecting the software or firmware from illegal users. Security protection is global. Not only flash memory but also other secured regions are protected. Other secured regions include backup SRAM (BKPSRAM), RTC backup registers and encrypted regions protected by real-time decryption (RTDEC). There are 3 levels for protecting.

No protection: No protection performed.

Protection level low: The main flash can only be accessed by user code. Option bytes block is accessible by all operations in debug mode, boot from SRAM or boot from boot loader mode.

Protection level high: Debug mode, boot from SRAM or boot from boot loader mode are disabled. The option bytes block can be read but can not be modified. The user option bits can be read but can not be modified. And accesses to the other secured areas are also allowed. The SPC byte cannot be reprogrammed. It cannot move back to protection level low or no protection level.

2.3. Security protection for the internal Flash memory

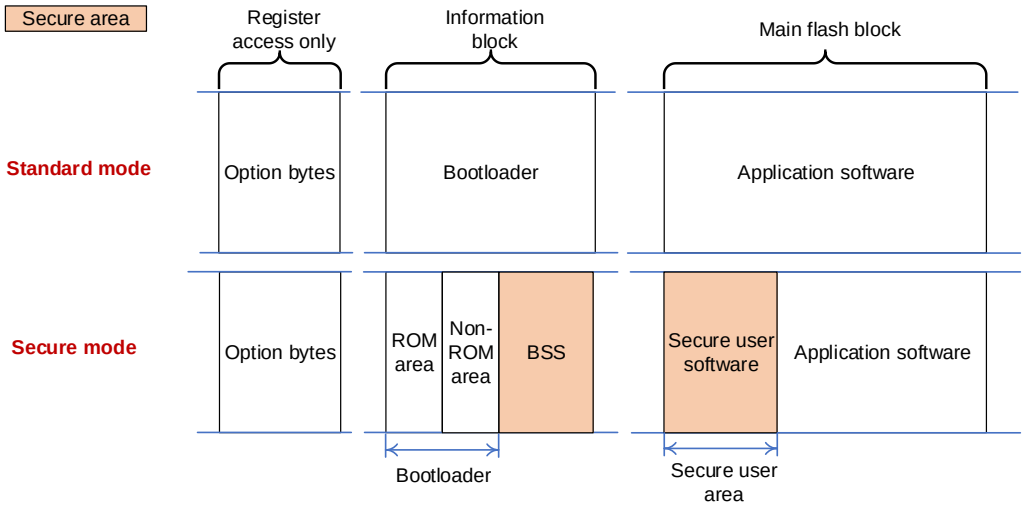
To protect various data and code, the GD32H73x_75x series MCU has standard mode and

secure mode.

In standard mode, the main flash block of the internal flash could be split to user area and DCRP area, which allows only instruction transactions from the system, but blocks data access.

In secure mode, a secure access area can be defined within the main flash block of the internal flash. The secure area can then be isolated from the rest of the Flash.

Figure 2-1. Memory architecture in standard mode and secure mode



2.3.1. Sector erase/program protection

Sector erase/program protection functions can prevent inadvertent operations on the flash memory. The sector erase or program will not be accepted by the FMC on protected sectors. The sector protection function can be individually enabled by configuring the WP[21:0] bit field to 0 in the option bytes.

Table 2-1. WP bit for sectors protection

WP bit	Sectors protected
WP[0]	Sector 0~Sector 15
WP[1]	Sector 16~Sector 31
. . .	. . .
WP[14]	Sector 224~ Sector 239
WP[15]	Sector 240~ Sector 255
WP[16]	Sector 256~Sector 383
WP[17]	Sector 384~Sector 511
. .	. .

WP bit	Sectors protected
.	.
WP[20]	Sector 768~ Sector 895
WP[21]	Sector 896~Sector 959

Note: For WP [x] (x=0 … 15), 1 bit corresponds to 16 sectors, ranging from sector (x * 16) to sector (x * 16 + 15). For WP [x] (x=17 … 21), 1 bit corresponds to 128 sectors, from sector (x * 128-1792) ~ sector (x * 128-1665). For WP[21], 1 bit corresponds to 64 sectors, from sector 896 ~ sector 959.

Modifying the internal flash or option bytes can only be done through the FMC_CTL or FMC_OBCTL register. After reset, the FMC_CTL or FMC_OBCTL register is not accessible in write mode, and the LK bit in FMC_CTL or FMC_OBCTL register is 1. They can be modified only after they are unlocked correctly.

2.3.2. Dedicated code read protection (DCRP)

In the main flash block, a DCRP area can be specified, allowing only instruction transactions from the system, but not data access.

The DCRP area can be defined by setting the DCRP_AREA_END[10:0] and DCRP_AREA_START[10:0] option bytes with a granularity of 4 Kbytes. The DCRP area can also be configured by modifying the MCU reserved parameter in EFUSE macro with granularity of 32KB bytes.

When executing code in this area, the debug events will be ignored. Only CPU can access DCRP area, using only instruction fetch transactions. In all other cases, access to the DCRP area is illegal. The valid DCRP area is erase-protected.

2.3.3. Secure user area

In the main flash block, a secure user area can be defined which can only be accessed when the CPU executes a secure application.

Secure user areas can isolate secure code from application non-secure code. Secure user areas can be used to protect a custom secure boot library, firmware update code, or a third party secure library.

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. 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.

When executing code in this area, the debug events will be ignored. Only CPU can access it by executing GD secure library or user secure application. In all other cases, accessing the secure user area is illegal.

A valid secure user area is erase-protected. Cannot erase sectors located in this area.

For more information on secure memory management, please refer to the ***AN113 GD32H73x_75x Secure Memory Management***.

2.4. Security protection for the external Flash memory

GD32H73x_75x series MCU only provides security protection for OSPI memory.

2.4.1. Real-time decryption (RTDEC) for OSPI memory

GD32H73x_75x series MCU provides up to two RTDEC. Each RTDEC can configure four independent and different encrypted areas. Each RTDEC block is assigned to one of the two OSPI ports and the OSPI should be configured as memory map mode. When the encrypted data is read from OSPI flash, the real-time decryption is performed by RTDEC in AES-128 CTR mode.

For more information on boot from OSPI flash, please refer to the ***AN122 Execution Environment of OSPI Flash of GD32H7 MCU User Guide***.

2.5. Unique device ID

Every device has a 96-bit unique identifier, which can't be modified. These bits can be read by the user and will not be repeated in all the GD32H73x_75x products. It can be used as serial numbers, or part of security keys, etc. This unique device ID can not be specified or changed by the user.

2.6. Hardware Unique Key (HUK)

Each device has a hardware unique key, which is a 128 bits private key. HUK provides the RoT (Root of Trust) for confidentiality. The HUK can not be accessed by the user directly but only by ROM code

2.7. Device certificate

Each device has a certificate used for exchanging encrypted messages. The certificate is signed by Gigadevice with our private key. It contains the public key, Signature and other information.

2.8. EFUSE

GD32H73x_75x series MCU supports 32*32 bits one-time programmable EFUSE, which are used for user data personalization or configuration. As a non-volatile unit of storage, the bit of efuse macro cannot be restored to 0 once it is programmed to 1. The EFUSE data structure as follow:

Table 2-2. EFUSE data structure

Parameter	Width/ bytes	Start address	Description
User control	4B	10'd0	User control parameter. For more details, refer to User control register (EFUSE_USER_CTL) in User_Manual.
MCU reserved	4B	10'd32	MCU reserved parameter. For more details, refer to MCU reserved register (EFUSE_MCU_RSV) in User_Manual.
Debug password	8B	10’d64	Debug password parameter. For more details, refer to Debug password register x (EFUSE_DPx) (x = 0,1) in User_Manual.
AES key	16B	10'd128	The AES key used to encrypt the firmware image. For more details, refer to Firmware AES key register x (EFUSE_AES_KEYx) (x = 0…3) in User_Manual.
User data	16B	10'd256	User defined data. For more details, refer to User data register x (EFUSE_USER_DATAx) (x = 0…3) in User_Manual.
User control	4B	10'd0	User control parameter. For more details, refer to User control register (EFUSE_USER_CTL) in User_Manual.

2.9. Cyclic redundancy checks management unit (CRC)

A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. The CRC calculation unit can be used to calculate 7/8/16/32 bit CRC code within user configurable polynomial.

2.10. True random number generator (TRNG)

The true random number generator (TRNG) module can generate a 32-bit random value by using continuous analog noise and it has been pre-certified NIST SP800-90B. 128-bit random value seed is generated from analog noise.

2.11. Cryptographic Acceleration Unit (CAU)

The cryptographic acceleration unit (CAU) is used to encipher and decipher data with DES, Triple-DES or AES (128, 192, or 256) algorithms. DES / TDES / AES algorithms with different key sizes are supported to perform data encryption and decryption in the CAU in multiple modes. The CAU is a 32-bit peripheral, DMA transfer is supported and data can be accessed in the input and output FIFO.

2.12. Hash Acceleration Unit (HAU)

The hash acceleration unit (HAU) is used for information security. The secure hash algorithm (SHA-1, SHA-224, SHA-256), the message-digest algorithm (MD5) and the keyed-hash message authentication code (HMAC) algorithm are supported for various applications. The digest will be computed and the length is 160 / 224 / 256 / 128 bits for a message up to (264 - 1) bits computed by SHA-1, SHA-224, SHA-256 and MD5 algorithms respectively. In HMAC algorithm, SHA-1, SHA-224, SHA-256 or MD5 will be called twice as hash functions and authenticating messages can be produced.

2.13. Secure JTAG

Secure JTAG function is only applicable to JTAG interfaces rather than SWD interfaces. The debugging interface is irreversibly switched from SWD interface to JTAG interface when this option is selected by programming the relevant EFUSE from 0 to 1.

For more information about Secure JTAG, please refer to the ***AN111 GD32H73x_75x Series Software Development Guide***.

2.14. System monitoring

The System monitoring feature can be used to detect incorrect operation and also help prevent some types of attack.

2.14.1. Tamper protection

Tamper detection is a feature dedicated to security, it can detect physical tampering attempts on the system level by an MCU pin, for example the opening of the product's enclosure. This feature can be powered from the RTC backup battery, it means that can always detect and trigger protection, even if the VDD is off. When a tamper attempt is detected, correlated registers may be reset, memory could be erased by user.

2.14.2. Clock monitor

HXTAL clock monitor (CKM) and LXTAL clock monitor (LCKM) are used to protect against external oscillator failures. Once the external oscillator failure is detected, the corresponding clock will be automatically switch to internal clock. And a clock failure event or interrupt will be generated. This failure interrupt is connected to the Non-Maskable Interrupt.

If a clock fault injection, freezing or glitch, used as part of an attack, it will be detected and prevented.

2.14.3. Power supply supervision

The abnormally-low voltage level will be detected, to ensure the MCU works in the certain voltage range. To guarantee the behavior of MCU, prevent a fault injection attack.

2.14.4. Temperature sensor

The internal temperature sensor can measure the device temperature on-going. Change the ambient temperature may be part of a fault injection attack or an on-line disassembly. The user can continuously monitor the device temperature and take appropriate actions.

2.15. Secure firmware

There is firmware within the GD32H73x_75x series that can only be used in Secure Mode but can not be read, even in secure mode.

2.15.1. Basic secure services (BSS)

Basic secure services (BSS) is a software provided by Gigadevice for security services configurations, it is located at the secure area of information block in the memory map.

For more information about BSS, please refer to the ***AN115 GD32H73x_75xGD32H7 series MCU Base secure services (BSS) user guide.***

BSS provides security services for boot-time and run-time.

BSS functions are listed below, the related secure services are called through bss.h:

Table 2-3. BSS function

Functions	Descriptions
BSS_get_version	get bss version
BSS_get_status	get bss status

Functions	Descriptions
BSS_get_certificate	get bss certificate
BSS_get_certificate_size	get bss certificate size
exitSecureArea	exit from the secure user area and jump to the user application
resetAndInitializeSecureAreas	Set the range of the secure user area based on the SCR_AREA_START and SCR_AREA_END option bytes

2.15.2. Immutable secure boot code

Secure boot code helps to implement system secure boot, it will execute after boot from ROM, and is responsible for application authentication and Integrity check before execution.

Those codes are hidden after boot, and can't be read by user.

3. Secure install and update

Firmware install and update is a very important part of the device lifecycle management. Firmware is signed using asymmetric keys and will be validated before installation on a GD32H73x_75xGD32H7 series MCU.

3.1. Licensed Firmware Install (LFI)

LFI is a secure mechanism implemented in GD32 microcontrollers that allows secure and counted installation of OEM firmware in untrusted production environment (such as OEM contract manufacturer). SFI is implemented in a secure bootloader. Licensed Firmware Install (LFI) are used to help securely install OEM firmware to the internal flash memory. LFI is only accessible in Secure Mode.

LFI allows OEM firmware code and configuration data being delivered and programmed in the form of encrypted files, then decrypted and installed inside the mcu, so to reduce the risk of firmware disclosure.

LFI must use HSM (Hardware Secure Module) provided by Gigadevice to implement authentication and integrity checks, OEM must use GD32 LFI Creator tool to program security firmware with its own AES secret key.

For more information about LFI, please refer to ***AN118 GD32H73x_75xGD32H7 series MCU Licensed Firmware Install (LFI) overview***.

3.2. Licensed Firmware Install X (LFIx)

Licensed Firmware Install X (LFIx) are used to help securely install OEM firmware to OSPI flash memory. LFIx can be used in Standard or Secure Mode to program external OSPI Flash memory with an encrypted image. This keeps external firmware and data secure throughout the life time of the product.

The user's program and data are always encrypted in life cycle.

For more information about LFIx, please refer to ***AN122 GD32H7GD32H73x_75x series MCU OSPI flash execution environment user guide***.

For more information about software operation, please refer to ***GD Licensed Data Creator User ManualAN133_GD32H7 MCU Security Firmware Generator***.

3.3. Licensed Firmware Update (LFU)

After the product is delivered to the end-user, OEM often needs to update firmware. The environment of update may be untrusted.

Licensed Firmware Update (LFU) helps to support secure updates and can use various communication interface, e.g. USB disk, SD card, serial port DFU and Ethernet.

LFU is a series of demos provided by Gigadevice. Users can redevelop according to their own needs, and freely combine.

For more information on secure memory, please refer to the ***AN119 GD32H73x_75xGD32H7 series MCU Licensed Firmware Update (LFU) overview***.

4. Secure boot

GD32H73x_75x series MCU provides a boot first ROM, according to the related security standard, e.g. PSA of ARM. This allows the system to implement a secure boot to an internal secure area of flash memory.

For more information on secure memory, please refer to the ***AN130 GD32H73x_75xGD32H7 series MCU Secure boot overview***.

The security goal of a secure boot is to verify the integrity and authentication of the next stage firmware. Secure boot code, also called immutable bootloader(IBL), is burnt in ROM or ROM-like storage at the time of chip manufacture. Digital signature verification algorithm ECDSA-secp256r1-SHA256 is used. The high 16 bytes of the public key hash value is stored by the user in EFUSE as the trust root. If the verification is successful, the next stage of firmware is executed. If validation fails, the code is in a loop waiting for a reset.

4.1. Boot ROM

In secure mode, following a system reset, the mcu boots into secure state and always executes the ROM code first. The code in the ROM verifies that the next stage boot code is permitted to execute.

4.2. Immutable Secure Boot code

The Secure Boot code(IBL) is set to be executed once and it can not be access from outside ROM. The following table ***Table 4-1. Secure boot hardware features*** lists the hardware features of the MCU. IBL supports signature verification by ECDSA algorithm. Thus, user can customize their own secure bootloader code, which is called main bootloader (MBL).

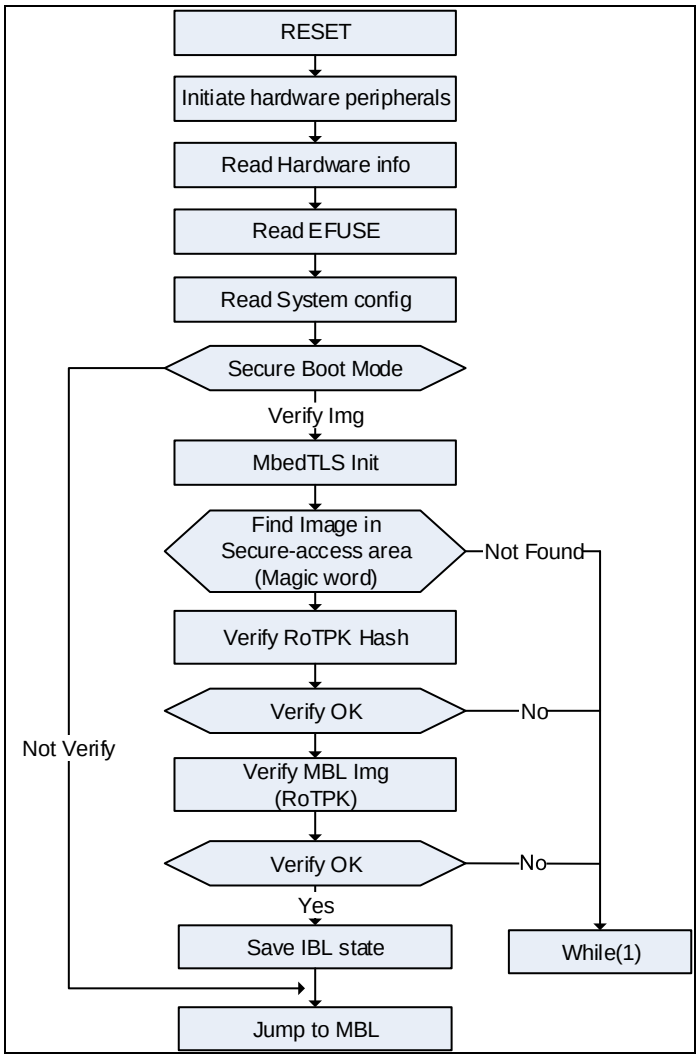
Table 4-1. Secure boot hardware features

Item	Requirement
EFUSE	1. EFUSE AES Key is write once 2. Secure boot is enabled by the a bit in EFUSE.
ROM	1. Close the ROM before jumping to MBL. Close the ROM is that IBL can be executed again only after the system is reset. 2. The IBL can access sram and secure-access area, while debug is turned off.
SRAM	1. After the system is reset, the sram area used by the IBL is automatically cleared.
Peripheral	1. TRNG, CAU, HAU engine data registers are cleared automatically after system reset.

Figure 4-1. Secure Boot code Flow shows the secure boot process. After the system is

reset, the IBL configures peripherals for the subsequent signature authentication process. If the verification passes, MCU jumps to MBL, otherwise, it goes into an infinite loop and waits for the next reset.

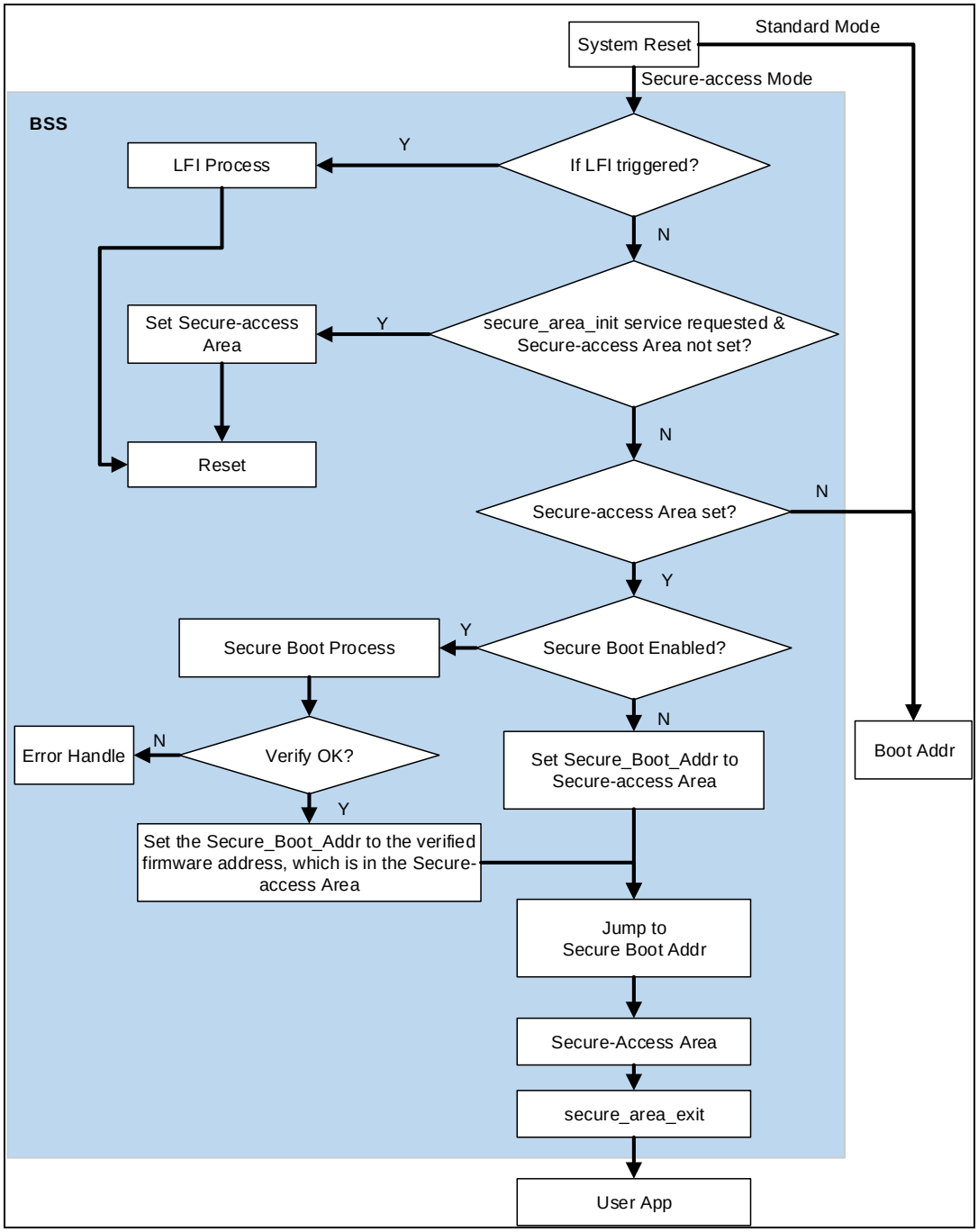
Figure 4-1. Secure Boot code Flow



4.3. Secure boot from the internal Flash memory

When the user application is running in internal flash memory, secure boot checks the code and jumps to the code when it is correct. The specific boot process is shown in ***Figure 4-2. Secure boot flow from the internal flash memory***.

Figure 4-2. Secure boot flow from the internal flash memory



The user needs to modify the start address and interrupt vector table of the code to ensure the correct jump and interrupt response of the code. LFI Creator and ALL IN ONE are used for code encapsulation and the MCU secure boot function is activated. The usage of GD32 All-In-One Programmer and GD LFI Creator can refer to chapter ***GD32AllInOneProgrammer*** and ***GDLicensedDataCreator***/Programmer

The GDLicensedDataCreator supports firmware encryption functionality.

For more information about this tools, please refer to ***GD Licensed Data Creator User Manual***.

The GDLicensedDataProgrammer supports encrypted firmware installation functionality. For more ***information about this tool, please*** refer to the ***GD Licensed Data Programmer User Manual***.

.

5. Boot from the OSPI memory

GD32H73x_75x series MCU provides a boot capability from external OSPI memory in standard mode, where the Secure Boot must boot works in Secure mode

Similar to secure boot, when boot from the OSPI memory, Immutable secure boot code will configure RTDEC and OSPI host before executing program in the OSPI memory. The code and data in the OSPI memory are encrypted by tools before download. So those can effectivity protect the firmware intellectual property of equipment manufacturer, even end-user private data(e.g. comtom initial login password) in the device.

For more information on boot from OSPI flash, please refer to the ***AN122 Execution Environment of OSPI Flash of GD32H7 MCU User Guide***.

6. Secure mode

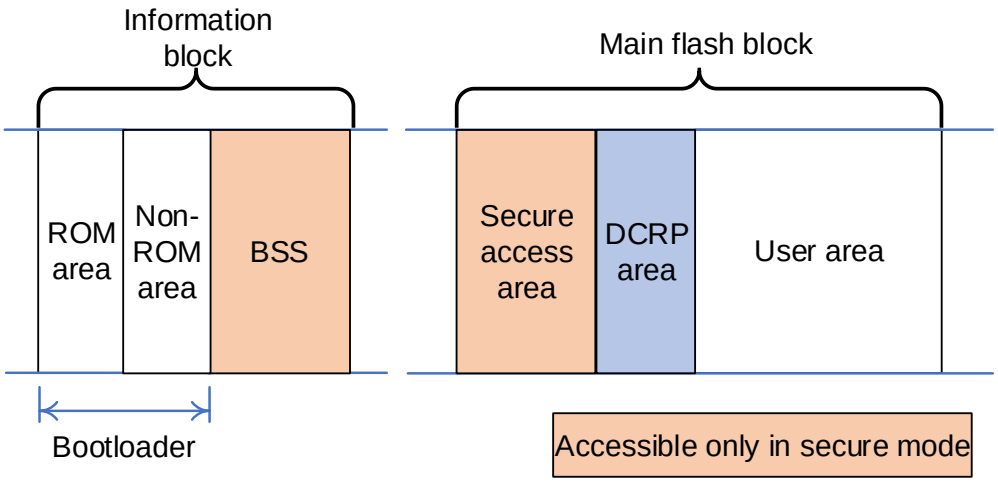
Secure mode is a specialized mode to implement secure boot, secure storage and secure access.

6.1. Isolated secure area in the internal Flash memory

User can split a secure area, which is isolated with normal area, secure area can access only in secure mode.

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

Figure 6-1. Flash protection areas



For more information on secure memory, please refer to the *AN113 GD32H73x_75x Secure Memory Management*

6.2. Secure mode rules

The secure mode follows the following rules:

- Regardless of the boot configuration (BOOT pins and boot addresses), MCU will be forced to boot from secure ROM area.
- Cannot jump back to secure areas after jumping from secure areas to application software.
- Basic security service (BSS) can only be called in secure mode.
- To return to standard mode, the secure area and DCRP area need to be removed.
- 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.

For more information on secure memory, please refer to the ***AN113 GD32H73x_75x Secure Memory Management***.

7. Secure ecosystem

7.1. Hardware security module (HSM)

HSM is a security device to negotiate KEK with mcu and generate license. HSM is in charge of:

1. Securely storing OEM key.
2. Verifying certificate that is used to authenticate GD mcu.
3. Negotiating KEK and providing license.
4. Counting number of produced GD devices.

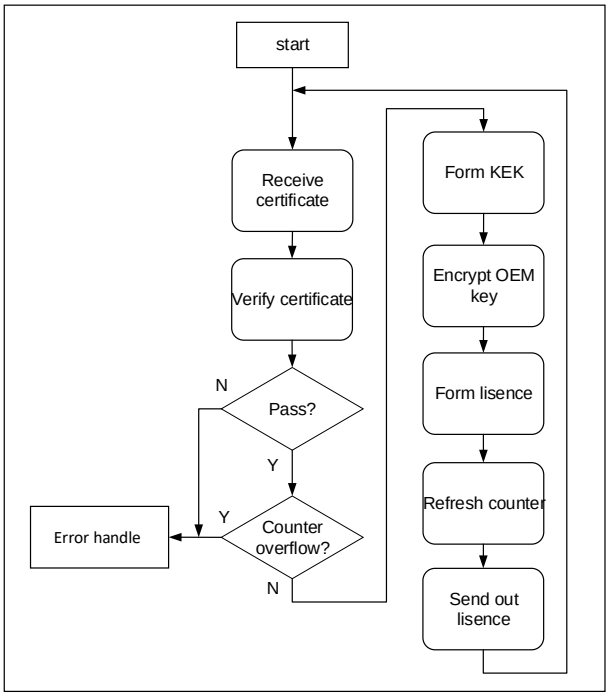
For more information about HSM, please refer to the ***AN116 GD32 MCU Hardware security module (HSM) overview.***

The interaction between HSM and PC is in USB-HID protocol.

Gigadevice is not responsible for developing HSM. However, an HSM example is developed for demonstration.

An example HSM work flowchart is described as below.

Figure 7-1. HSM workflow example



7.2. GD32AllInOneProgrammer

GD32 All-In-One Programmer support encrypted programmer base EFUSE key and OSPI programmer functions.

For more information about this tools, please refer to ***GigaDevice All-In-One Programmer User Manual***.

7.3. GDLicensedDataCreator/Programmer

The GDLicensedDataCreator supports firmware encryption functionality.

For more information about this tools, please refer to ***GD Licensed Data Creator User Manual***.

The GDLicensedDataProgrammer supports encrypted firmware installation functionality. For more information about this tool, please refer to the ***GD Licensed Data Programmer User Manual***.

8. Revision history

Table 8-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Apr.18, 2023
1.1	Partial statement modification	Sep.13, 2023
1.2	The chip series has been changed to GD32H73x75x	Feb.10, 2026

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 according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. 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.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its 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 Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.