GETTING STARTED GUIDE

Apollo3 Blue Security

Ultra-Low Power Apollo SoC Family

A-SOCA3B-GGGA01EN v1.3

Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING WITHOUT LIMITATION CONSEQUENTIAL OR INCIDENTAL DAMAGES. “TYPICAL” PARAMETERS WHICH MAY BE PROVIDED IN AMBIQ MICRO DATA SHEETS AND/OR SPECIFICATIONS CAN AND DO VARY IN DIFFERENT APPLICATIONS AND ACTUAL PERFORMANCE MAY VARY OVER TIME. ALL OPERATING PARAMETERS, INCLUDING “TYPICALS” MUST BE VALIDATED FOR EACH CUSTOMER APPLICATION BY CUSTOMER’S TECHNICAL EXPERTS. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.

Revision History

Revision	Date	Description
1.0	September 9, 2021	Initial Release.
1.1	December 16, 2021	▪ Added new Signature content ▪ Updated content in section KREVTRACK Register ▪ Updated content in section AREVTRACK Register ▪ Updated content in section CUSTKEKW* Register ▪ Updated content in section CUSTAUTHW* Register
1.2	February 8, 2022	▪ Updated Reference Document section ▪ Updated Apollo3 Plus and Apollo3 Blue Plus INFO0 Block ▪ Updated Apollo3 and Apollo3 Blue INFO0 Block
1.3	October 19, 2022	▪ Updated document part number.

Reference Documents

Document ID	Description
A-SOCA3B-UGGA02EN	Apollo3 Blue Secure Update Flow User’s Guide
DS-A3P-1p1p2	Apollo3 Blue Plus Datasheet
DS-A3-1p0p2	Apollo3 Blue Datasheet

Table of Contents

SECTION

1

Terminology

Table 1-1 describes the terminology used in this document.

Table 1-1: Terminology

Abbreviation	Definition
HMAC	Keyed-Hash Message Authentication Code
INFO0	Refers to the customer Flash INFO block which is used to store customer part-specific configuration.
KEK	Key Encryption Key
OTA	Over the Air. For simplification, the update flow will sometimes be referred to as just "OTA" even if a wired interface is used to download the image(s).
RoT	Root of Trust
SBL	Ambiq Secure Boot Loader (referenced either as SBL or Ambiq SBL)
SBL_CUST	Customer Secure Boot Loader or also referenced as Secondary Secure Boot Loader

SECTION

2

Overview

The security features on the Apollo3 Blue SoC Family enable a trusted firmware model which is critical for wearables and general Internet of Things (IoT) devices. Unless otherwise noted, the following information for the Apollo3 Blue is equally applicable to the Apollo3 and the Apollo3 Blue Plus.

The Apollo3 Blue establishes a Root-of-Trust leveraging hardware and secure firmware and maintains this secure foundation throughout the device boot/reset flow. This is based on the Ambiq® secureSPOT® technology. To ensure a full end-to-end security model, Apollo3 Blue not only supports a secure/trusted boot flow but also supports secure firmware updates (wired or OTA), secure key storage and provisioning, secure in-field feature upgrades, debug policy enforcement, Flash memory protection as well as inline scrambling/descrambling for external memory interface. Most of these features is described in this document. Note that the secure boot feature is only supported on secure boot enabled SKUs.

SECTION

3

Secure Life Cycles

The Apollo3 Blue has several secure life cycle states. These life cycle states define the security policy, how the policy is enforced (if applicable), and what resulting action(s) should take place based on certain features and any exceptions that may occur during execution.

The following life cycle states are defined for the Apollo3 Blue as follows:

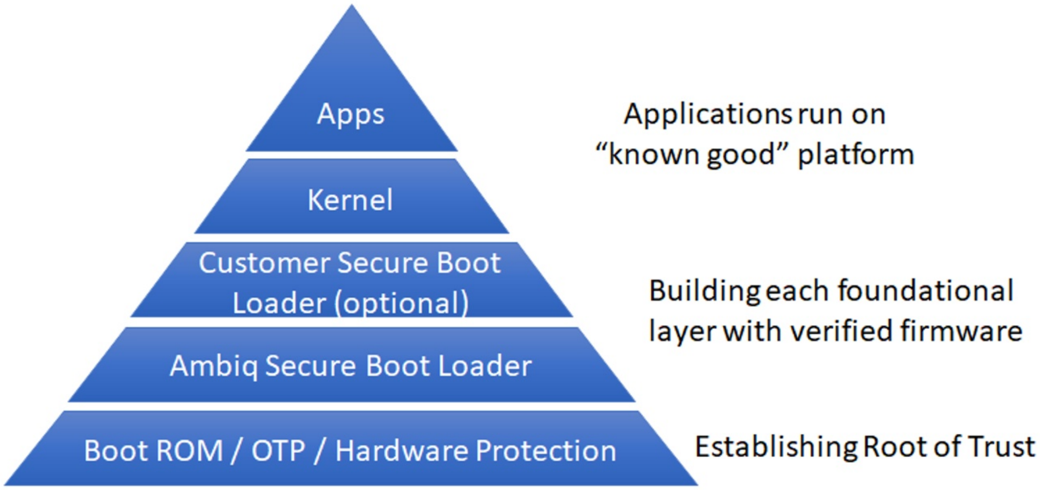
- Customer Manufacturing
  - Device is open to initial programming using SWD or other wired interface (UART, SPI/I^2C)
  - Customer debug is unlocked
  - While in this state, the device can be locked/partially locked/open based on customer security policy to support additional development prior to entering Production state.
- Customer Production
  - The device is in the final state, enforcing customer security policy
  - Customer debug is locked/unlocked based on customer security policy

SECTION 4

Secure Boot

Secure boot establishes a trusted execution foundation for the device. The secure boot flow is based on the following security model.

Figure 4-1: Apollo3 Blue Security Model


Apps
Kernel
Customer Secure Boot Loader (optional)
Ambiq Secure Boot Loader
Boot ROM / OTP / Hardware Protection
Applications run on "known good" platform
Building each foundational layer with verified firmware
Establishing Root of Trust

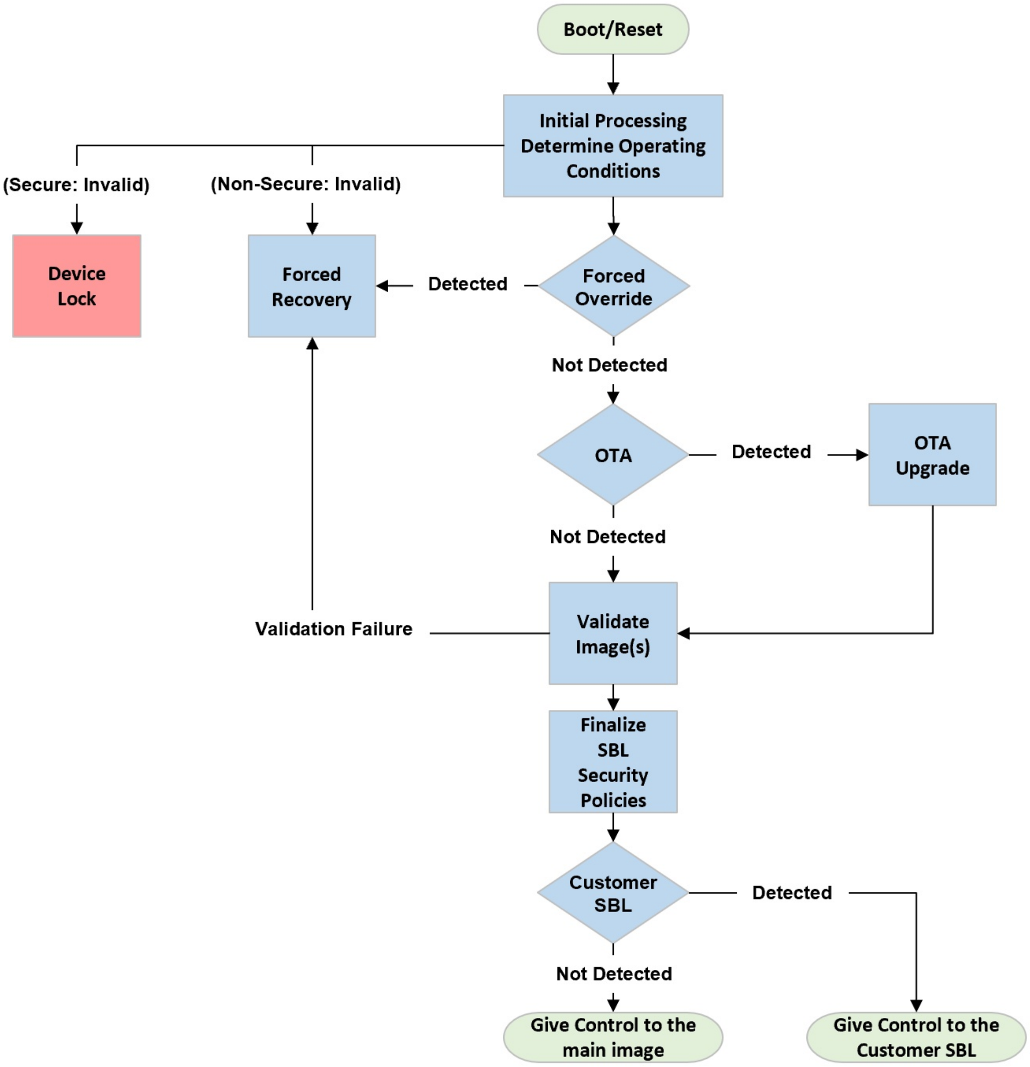
At the base of the security model is the Root-of-Trust (RoT), which is based on physical hardware in the SoC. This foundation is "known good" and trusted. From this foundational RoT, there is a trusted embedded firmware/hardware layer known as the Ambiq Secure Boot Loader (SBL). The SBL is provisioned at manufacturing and hardened/locked. The boot ROM within the hardware layer is responsible for ensuring the integrity of SBL and initiating the next level of the secure boot flow. SBL is responsible for verification of the subsequent firmware layers to complete the security model pyramid. An optional customer secure boot loader (SBL_CUST) a.k.a Secondary Secure Boot Loader can be installed, allowing a customer to customize their particular secure boot process as needed. This allows the customer to implement custom, proprietary and/or stronger cryptographic functions as well as custom protocols for firmware updates as/if needed to meet their security requirements. The SBL_CUST can also be used to enable access to external storage for expanded firmware validation or secure firmware update (wired or OTA).

The Ambiq Secure Boot Loader provides several functions at boot/reset with respect to secure boot (if permitted): forced recovery (customer boot loader or main firmware only), secure firmware update (wired or OTA) and firmware validation/installation. The security policy and policy enforcement are driven by several parameters via INFO0 configuration and embedded image header fields to provide flexibility for the required security enforcement for the product.

There are two security levels depending on the particular Apollo3 Blue SKU.

- **Non-Secure**: Secure boot and some other secure services are not supported. For these parts, INFO0:SECBOOT has no effect. Note that certain security capabilities like Secure OTA, and debugger lockout can still be availed.
- **Secure**: Secure boot and other secure services are supported. SBL can be in-field updated using secure OTA and recovered using secure recovery flow.

Figure 4-2: Secure Boot Loader High Level Flow



For Secure SKUs, secure boot is enabled by programming the INFO0:SECURITY:SECBOOT field in the customer Flash INFO0 block. This field must be set to 0x2 to enable the secure boot feature. If set to 0x5, the feature is disabled and the part will boot in non-secure mode. All other encodings will generate an exception. Note for this particular SKU if customer INFO0 space is uninitialized (e.g., Customer Manufacturing state), SBL will treat this as a non-secure part to enable programming of the device. Once INFO0 is programmed, the device will be treated as secure or non-secure based on the INFO0:SECURITY:SECBOOT field.

4.1 Ambiq Secure Boot Loader

The Ambiq Secure Boot Loader is preinstalled at Ambiq device manufacturing and comes preinstalled on all Apollo3 Blue devices. The firmware is verified at manufacturing utilizing a part-specific key, the Ambiq Hardware Unique Key and a CRC32 hash of the firmware image loaded. A unique signature is computed on chip in hardware and then verified as part of an automated integrated function within the test program. This ensures the required image has been loaded correctly and is valid. Once the image is loaded on the device, the sections of Flash where the SBL partitions are programmed are locked.

At device cold boot or any reset, the integrity of SBL is reverified using the same CRC32 hash. Additionally, the SBL image is checked to ensure it matches with the Flash and/or SRAM memory bounds (depending on the particular device SKU).

Once hardware verification of the SBL is complete and successful, SBL execution begins. If any verification check fails at this stage, the part is locked and permanently unrecoverable.

The SBL includes all functionality to support secure and non-secure boot loading, secure firmware update (wired or OTA), secure in-field feature upgrades (if permitted) and firmware recovery. The pre-installed SBL can also be updated (if permitted) via a secure firmware update (wired or OTA) described in *Section 6 Secure Firmware Update on page 16*.

At initial reset/boot, SBL registers the secure lifecycle state (described in *Section 3 Secure Life Cycles on page 7*) and customer security policy to direct the secure boot flow accordingly. The security policy configurations that direct the Ambiq Secure Boot Loader are listed in *Section 13 Apollo3 Plus and Apollo3 Blue Plus INFO0 Block on page 24*.

The Ambiq Secure Boot Loader checks if a valid firmware update (OTA or wired) is pending or if a recovery is required. For firmware updates, see *Section 6 Secure Firmware Update on page 16* for details. For recovery flow, see *Section 10 Firmware Recovery and Override on page 21* for details.

4.1.1 Non-Secure Boot

This is the boot mode for all nonsecure SKU devices. In addition, even for secure SKU, if the life cycle state is **Customer Manufacturing** (INFO0 has not yet been programmed by customer), or If the INFO0:SECURITY:SECBOOT is set to **disabled** (this might be the case during early development in "Customer Development" Stage), the Ambiq Secure Boot Loader defaults to non-secure boot.

When booting in non-secure boot mode, SBL will look for a valid firmware image loaded at INFO0:MAIN_PTR0 (defaults to 0xC000 for uninitialized INFO0 case). If a valid firmware image is detected, SBL will verify the Stack Pointer and Reset Vector values to ensure they reference valid locations within the Apollo3 Blue memory space based on the device configuration. SBL jumps to the specified firmware image and begins execution. If the image is invalid, a valid firmware image needs to be installed using either a debugger or supported wired interface for update.

4.1.2 Secure Boot (For Secure SKU only)

For a secure SKU device, once the INFO0:SECURITY:SECBOOT is set to **enabled** (and INFO0 signature is initialized to be Valid), the firmware validation flow is executed. There is an additional policy configuration that can direct the SBL to skip the firmware validation step on a soft reset. If the INFO0:SECURITY:SECBOOTONRST is set to **disabled**, SBL will skip the validation sequence for soft reset. This configuration should be used with caution but can be used if reset latency is a concern.

The Ambiq Secure Boot Loader supports AES128-CBC for decryption, SHA-256 HMAC for authentication and CRC32 for integrity check. Additional cryptographic algorithms can be supported as part of the Customer (or Secondary) Secure Boot Loader (SBL_CUST). The key storage has provisions to support up to 2048b of public key storage with a 256b public key hash as well as 1024 bits of symmetric key encryption keys and 1024 bits of authentication keys.

Also note that all firmware images processed by the Ambiq Secure Boot Loader must reside within embedded/on-chip Flash. External Flash (via the MSPI or SPI interface) is not supported by SBL. Support for external non-volatile memory requires a Customer Secure Boot Loader.

Each firmware image that is validated on secure boot has a boot-time (or at-reset) security policy configuration that informs the SBL what security operations need to be performed. These configuration bits are described in the respective firmware header descriptions in the *Section 10 Firmware Recovery and Override on page 21*. In addition, the customer can configure a mandatory security policy to be enforced when updating the images to ensure secure boot chain and policy is not compromised when doing updates. This is defined in the INFO0:SECURITY:SECPOL field.

Each firmware image header specifies the specific algorithm to be performed as well as the respective authentication key and/or Key Encryption Key (KEK) index. An additional version field is also provided which can be checked for anti-rollback protection. This is described in more detail in the *Section 11 Secure Updates on page 22*. If any firmware validation steps fail, SBL will go into a firmware recovery step. The firmware recovery process is described in the *Section 10 Firmware Recovery and Override on page 21*.

Prior to hand off to either a Customer Secure Boot Loader or the main image, the SBL memory region is both write and copy protected in hardware.

4.2 Customer Secure Boot Loader

An optional Customer Secure Boot Loader is supported, which allows the customer to implement custom secure boot loader functions. An example of usage would be if a public key or proprietary cryptographic algorithm is required or if external Flash storage is used to store firmware to be validated. The Customer Secure Boot Loader has access to the customer key bank as needed for key storage and verification.

For firmware updates, the Ambiq Secure Boot Loader supports various image types as listed in the *Section 12 Image Header Formats on page 23*. For customer-specific images to be validated exclusively by the Customer Secure Boot Loader, the magic numbers allocated for these images must not conflict with those reserved for the Ambiq Secure Boot Loader. Customers can use 0xC1-0xCA, 0xCE-0xCF. All others are currently reserved for Ambiq use.

SECTION

5

Firmware Provisioning

Customer firmware provisioning is performed in a few different ways depending on the life cycle state of the part and the security policy configuration.

5.1 Customer Manufacturing

In this life cycle state, the customer INFO0 block is not yet programmed. The subsequent programming steps depend on whether the device is secure boot enabled (for Secure SKU only) or non-secure boot enabled.

Caution should be taken as some of the configuration settings, when set, will prevent further updates to INFO0 block and debugger capabilities. Specific fields to treat carefully are INFO0:SECURITY[8:0]. It is recommended that INFO0 protection, and debugger locking, if required – be implemented as the last step, if provisioning in stages.

5.1.1 Non-Secure Boot Enabled

In Customer Manufacturing state, INFO0 block is by default set to uninitialized state and, as such, the device is treated as "non-secure" by default. In this configuration, the INFO0 block is not required to be programmed. Customer firmware can be installed either by using a debugger port (SWD) from the Ambiq provided Flash Helper functions, or by a supported wired interface (see *Section 1 Firmware Recovery and Override on page 21* for details). The firmware is expected to be installed at offset 0xC000 in internal Flash. The Ambiq Secure Boot Loader provides the loader support to install the firmware and begin execution at the installation offset. No additional programming is required.

During Customer Development phase, this configuration can be used, which will allow firmware programming and re-programming as needed, and is also more suitable for developer debugging.

If the device is to be programmed as a "non-secure boot" part by initializing the INFO0 block, care must be taken to ensure the device is properly initialized. Although the device may be "non-secure boot", other security features can/will still be leveraged such as secure OTA.

- INFO0:SECURITY field must be programmed to configure the respective security policy per the customer's requirements.
  - SECBOOT must be set to 0x5 to disable secure boot.
  - SECPOL and KEYWRAP must be programmed per the customer's security requirements.
  - SECURITY:SECURE_LOCK MUST NOT be enabled, this bit should remain 1.
  - All other fields can be left unprogrammed to '1' for safe values.
- INFO0:CUSTOMER_TRIM field must be programmed to a valid configuration based on the customer's board requirements. Incorrect programming of this field may lead to an unstable part.
- INFO0:MAINPTR0 field must be programmed to a valid page aligned address in internal flash to point to the start of the main image.
- INFO0 signature should be programmed last or at the same time as the security and trim fields to avoid mis-programming and locking the part.

5.1.2 Secure Boot Enabled (Secure SKU Only)

For this configuration, the INFO0 block **must** be initialized as follows. Note that once this step is executed, the part will be enabled for secure boot, which means all firmware loading will require secure firmware update (wired or OTA) and caution should be taken to ensure INFO0 block is programmed correctly. Incorrect programming at this stage may result in a locked out part.

- INFO0:SECURITY field must be programmed to configure the respective security policy per the customer's requirements.
  - SECBOOT must be set to 0x2 and SECBOOTONRST must be set to a valid encoding.
  - SECPOL and KEYWRAP must be programmed per the customer's security requirements
  - SECURITY:SECURE_LOCK MUST NOT be enabled, this bit should remain 1.
  - All other fields can be left unprogrammed to '1' for safe values
- INFO0:CUSTOMER_TRIM field must be programmed to a valid configuration based on the customer's board requirements. Incorrect programming of this field may lead to an unstable part.
- INFO0:MAINPTR0 field must be programmed to a valid page aligned address in internal flash to point to the start of the signed main image.
- INFO0 signature should be programmed last or at the same time as the security and trim fields to avoid mis-programming and locking the part.

Once INFO0 block is programmed with a valid programming sequence (e.g., INFO0 signature is valid), the device is now controlled under secure boot. Only appropriately formatted signed images will be accepted by the SBL. All subsequent updates will need to be done using secure firmware update (wired or OTA) (see *Section 1 Secure Firmware Update on page 16*) using appropriate image types. Secure Recovery (see *Section 1 Firmware Recovery and Override on page 21*) needs to be used to factory reset the device.

5.2 Customer Production

Once in the Customer Production state, if the device is secure boot enabled, only the secure firmware update flow can be used to provision firmware as described in the *Section 1 Secure Firmware Update on page 16*. Only exception is for firmware recovery, which needs to follow the flow detailed in the *Section 1 Firmware Recovery and Override on page 21*.

In this state, customers can choose the appropriate security configuration depending on their production flow and security requirements. For secure boot enabled devices, it is recommended that the debugger lock be set to disable any further debugger access.

SECTION

6

Secure Firmware Update

See *Apollo3 Blue Secure Update Flow User's Guide A-SOCA3B-UGGA02EN* document for details.

SECTION

7

Secure Key Management

The Apollo3 Blue SoC provides secure key banks for Ambiq and/or Customer use. The Customer key bank is an 2kB block in the Flash INFO0 sector. The key bank is partitioned to allocate regions for Authentication keys, Key Encryption Keys, public key, and a public key hash. Although the number of keys is specific to the customer implementation, the region of INFO0 sector that has been allocated to each function is fixed. This is to allow appropriate indexing and revocation as/if needed.

The Customer Key Bank is available to the Ambiq Secure Boot Loader as well as the Customer Secure Boot Loader for secure boot authentication and decryption. After secure boot completes, this key bank can only be accessed with the Customer KEYBANK Key. This key is provisioned by the customer. Access to the key bank can be unlocked after secure boot completes by writing the KEYBANK key to the MCU_CTRL:CUSTOMERKEY0-3 register. Writing any other value to this register relocks the keybank. The KEYBANK key must be controlled by customer secure firmware.

All keys stored in the key bank can be wrapped. This is to provide additional security of the keys. The algorithm used for key wrapping (and subsequent unwrapping) is specified in the INFO0:SECURITY:KEYWRAP field.

Multiple keys can be stored in the Authentication key sector or the KEK sector. The key to be used for the respective operation is referenced via a key index provided as part of the image header. Ambiq specific key indices are 0-7. Customer-specific key indices are 8-15.

The Apollo3 Blue also supports key revocation. This allows the customer to restrict/retract specific keys from being used. See *Section 7.3 Key Revocation on page 18* for details.

7.1 Key Provisioning

The Customer keys must be programmed before secure boot is enabled (e.g., when INFO0:SECURITY:SECBOOT is programmed). Keys can be provisioned via a Serial Wire Debug connection using the Ambiq Flash Helper functions or the wired loader interface using the flow specified in *Section 10 Firmware Recovery and Override on page 21*. This depends on the life cycle state of the part and the preferred provisioning method. The keys must be wrapped prior to being loaded onto the device. The wrap method is up to customer. ChipID information can be read from the device to enable device binding of the key wrap if desired. If a custom wrap algorithm is used (e.g., a method other than that specified/supported by the Ambiq Secure Boot Loader), these keys can only be referenced by the Customer Secure Boot Loader as the Ambiq SBL will not be able to unwrap.

Once keys are provisioned, it is advised to lock the key bank modifications by clearing the corresponding SECURITY:EN_CUST_INFO_PROG bit. This field has 4 bits where each bit represents a quadrant of the INFO0 partition. The key bank is in the upper quadrant controlled by bit3 of this field (e.g., set SECURITY:EN_CUST_INFO_PROG = 4’b0xxx).

7.2 Key Wrapping

Apollo3 Blue supports chip specific wrapping of the keybank in INFO0 to further enhance security and reduce potential exposure and risk on unauthorized access. Three keywrap modes are supported, as configured in SECURITY:KEYWRAP.

- None (0): No keywrapping, Actual keys are programmed as-is in INFO0
- XOR (1): The keys are derived as follows:
  - Key = INFOKey ^ CustomerKey ^ ChipIDKey
  - Here ChipIDKey is generated as 64b ChipID repeated to the key length
- AES128 (2): The keys are derived as follows:
  - Key = FAES128-CBC(INFOKey, CustomerKey, IV)
  - Here IV is generated as 64b ChipID duplicated to 128b

7.3 Key Revocation

Key revocation is supported using two monotonic counters in nonvolatile Memory, KREVTRACK and AREVTRACK. The Secure Boot Loader checks these counters to ensure the KEK index or Auth Key index are not referencing values marked “invalid” by these counters. To update a counter (revoke a key), a secure update would be required.

Example use of these counters:

For a 128-bit KEK,
KREVTRACK = 0x7FFFFFFF
KEK index = 0 would be invalid as the msb of the KEK bank is marked invalid.
KEK index = 1 would be the first valid KEK
To then revoke KEK1, the KREVTRACK would need to be updated to 0x3FFFFFFF.

Each bit (starting with the msb) represents a key index. Updates to enforce key revocation are performed using the same firmware update flow specified in *Section 7 Secure Key Management on page 17*. After a key is revoked, all references to the revoked key index/indices will fail.

SECTION

8

Flash Protection

There are several protection mechanisms to support various customer security policies. Write and Copy protection configs can be set in INFO0 (using SECURITY:WRITE_PROTECT_L/H, COPY_PROTECT_L/H, WRITE_PROTECT_SBL_L/H and COPY_PROTECT_SBL_L/H in Apollo3 and Apollo3 Blue, and using SECURITY:WRITE_PROTECT*, COPY_PROTECT_*, WRITE_PROTECT*_SBL and COPY_PROTECT*_SBL in Apollo3 Plus and Apollo3 Blue Plus).

Each protection bit represents a 16kB chunk of Flash. Clearing a particular bit in these fields restricts the appropriate action (write or read) to the corresponding 16kB portion of Flash. The WRITE/COPY protections are applied always before SBL execution. This allows for certain blocks within Flash to be unconditionally protected if needed. The WRITE/COPY_PROTECT_SBL protections are applied by SBL during the secure boot execution. These protections get applied as needed based on any OTA updates since SBL needs access to certain regions of Flash to perform these updates.

There are several other security policy configs defined in the customer INFO0 partition (see the INFO0 section of this document for details). The INFO0 partition can also be protected to prevent updates to security policies, keys, etc., once the part is provisioned/secured. The SECURITY:EN_CUST_INFO_ERASE and SECURITY:EN_CUST_INFO_PROG fields allow the customer to lock specific portions/access to the INFO0 partition as needed. When SECURITY:EN_CUST_INFO_ERASE is set (programmed to ‘0’), the INFO0 partition can no longer be erased. Only an Ambiq supported device recovery will allow the device to be re-provisioned.

The SECURITY:EN_CUST_INFO_PROG has 4 bits where each bit represents a quadrant of the INFO0 partition. Whenever one of the bits is set (e.g., programmed to ‘0’), that particular quadrant of INFO0 cannot be programmed further. If SECURITY:EN_CUST_IN- FO_ERASE is also set (programmed to '0'), this essentially locks the INFO0 quadrant preventing any further modifications. Only an Ambiq supported device recovery will allow the device to be re-provisioned.

SECTION

9

Debug

There are multiple debug disable options supported to assist with debugging at different stages of production. The SECURITY:SDBG bit, when set (e.g., programmed to ‘0’), disables debugger access during secure boot until the MCU_CTRL:BOOTLOADER:PROTLOCK is set. This allows customers to enable debug while still in development of the secondary secure bootloader. Once ready for production, this bit should be set disabling debug during this portion of secure boot. Note that the debugger is always disabled during Ambiq SBL execution.

The SECURITY:DEBUG_PROT bit, when set (e.g., programmed to ‘0’), disables debugger access completely. This bit should be set for production.

Ambiq Secure Bootloader wipes all the SRAM on exit by default. There is an additional field (SECURITY_SRAM_RESV), allowing the customer to allocate a portion of SRAM that will not be subject to the wipe operation. This allows customers to preserve certain information in SRAM across soft reboots (e.g., for debugging failures).

SECTION 10

Firmware Recovery and Override

Firmware recovery is required whenever the device has encountered an unrecoverable error. This can occur due to a validation error during secure boot flow. Recovery can be performed via a wired interface or via SWD, if enabled.

The wired interface used is configured using INFO0:SECURITY_WIRED_CFG and INFO0: SECURITY_WIRED_IFC_CFG0-5 fields. UART, SPI and I^2C are supported as wired interface. At initial programming, the default wired interface is available. By default (uninitialized INFO0, and flash), wired interface is enabled using UART0. External host should be connected to Apollo3 Blue using UART-RX pin 49, and UART-TX pin 48, using baudrate 115200 with no flow control.

A forced override can also be initiated, if enabled, using the specified override GPIO configured in INFO0:SECURITY_OVR. If this field is 0x3F, the override feature is disabled.

Upon boot or reset, the Ambiq Secure Boot Loader detects if a recovery or a forced override is to be initiated.

The SBL provides support for an external host to connect during boot up to upgrade images or to recover a failing device. SBL initiates recovery if any of the boot-time validations fail or if there is no valid image to boot. Override is a feature provided by SBL where a forced image upgrade can be initiated using specific GPIO settings during the boot up.

In either case, an external host needs to follow a predefined messaging protocol to instruct SBL to upgrade assets on the device as part of the wired update process. For wired updates, the SBL will poll on the specified wired interface until the timeout specified in INFO0:SECURITY_WIRED_CFG:TIMEOUT. Upon detection, the recovery update flow is followed.

See *Apollo3-Blue Secure Update Flow* document for more details.

SECTION

11

Secure Updates

See *Apollo3 Blue Secure Update Flow User's Guide A-SOCA3B-UGGA02EN* document for more details.

SECTION

12

Image Header Formats

See *Apollo3 Blue Secure Update Flow User's Guide A-SOCA3B-UGGA02EN* document for details on image header formats.

SECTION

13

Apollo3 Plus and Apollo3 Blue Plus INFO0 Block

See *Apollo3 Blue Plus Datasheet DS-A3P-1p1p2* for more information on the Apollo3 Plus and Apollo3 Blue Plus INFO0 block.

SECTION

14

Apollo3 and Apollo3 Blue INFO0 Block

See *Apollo3 Blue Datasheet DS-A3-1p0p2* for more information on Apollo3 and Apollo3 Blue INFO0 block.

6500 River Place Boulevard, Building 7, Suite 200, Austin, TX 78730
www.ambiq.com
sales@ambiq.com
+1 (512) 879-2850