Application Note

Cybersecurity Enablers in MSPM0 MCUs

ABSTRACT

MSPM0Gxx and MSPM0Lxx microcontrollers provide a variety of security enabler technologies to help developers implement their security measures to protect assets such as code, data, and keys. This document describes the enablers provided in these devices, what their capabilities and limitations are, how they operate, and how to configure them for basic use cases.

Table of Contents

1 Introduction.....2
1.1 Goals of Cybersecurity.....2
1.2 Platform Security Enablers.....2
2 Device Security Model.....4
2.1 Initial Conditions at Boot.....4
2.2 Boot Configuration Routine (BCR).....4
2.3 Bootstrap Loader (BSL).....4
2.4 Boot Flow.....5
2.5 User-Specified Security Policies.....5
3 Secure Boot.....13
3.1 Secure Boot Authentication Flow.....13
3.2 Asymmetric vs. Symmetric Secure Boot.....13
4 Cryptographic Acceleration.....15
4.1 Hardware AES Acceleration.....15
4.2 Hardware True Random Number Generator (TRNG).....16
5 Device Identity.....16
6 Summary.....16
7 References.....17
8 Revision History.....17
A Security Enablers by Subfamily.....18

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

As industrial, automotive, and personal electronics applications become more connected, and as the tools available to attackers continues to grow, the importance of device security in embedded applications continues to increase. MSPM0 microcontrollers from TI include a variety of hardware and software security enabling technologies for engineers to leverage when developing an application with security in mind.

1.1 Goals of Cybersecurity

In general the key goals of cybersecurity in an embedded application are to protect critical assets as follows:

- Confidentiality (keeping secret data secret)
- Integrity (protecting data from modification)
- Authenticity (ensuring all parties are who they claim to be)
- Availability (ensuring that data and/or functionality is there when it is needed)
- Non-repudiation (origin and/or identity of data is provable to additional parties)

These key goals are often applicable for assets which can be in the following states:

- At rest (code, data, or keys on a microcontroller which are not actively being used)
- In use (code, data, or keys on a microcontroller which are being actively used in an application)
- In transit (code, data, or keys on a microcontroller which are moving between an MCU and another entity)

1.2 Platform Security Enablers

The security enablers included in MSPM0 devices are given in Table 1-1. A complete list of security enablers available across the broader range of TI products can be found at the TI security portal.

Table 1-1. MSPM0 MCU Platform Security Enablers

Security Enabler	Device Feature	MSPM0L	MSPM0G
Debugging security	Password authenticated debug access	All	All
	Password authenticated bootstrap loader access	All	All
	Password authenticated main flash memory mass erase	All	All
	Password authenticated complete factory reset	All	All
	TI failure analysis (FA) enable/disable	All	All
	Complete hardware disable of serial wire debug (SWD) interface	All	All
	Permanently lockable device configuration data	All	All
	Error resistant device configuration data	All	All
	Password memory contains hashes only (SHA2-256)	Future	Future
Secure boot	Permanently lockable main flash memory (static write protection)	All	All
	CRC-32 verified main flash region	All	All
	SHA2-256 verified main flash memory region	Future	Future
	Single point of entry to main flash application at boot	All	All
	Firmware image authentication routines (asymmetric or symmetric)	All	All
	Lockable flash for key revocation and rollback protection	Future	Future
	W^X (write-or-execute) SRAM boundary	All	All
Secure Storage	Static flash memory read/execute (RX) firewall	Future	Future
	IP protection (execute-only) firewall	Future	Future
	W^X (write-or-execute) enforcement on main flash banks	Future	Future
	AES volatile key store (up to four 128-bit keys plus a session key)	Future	Future
Cryptographic acceleration	Hardware AES accelerator (128-bit / 256-bit)	Future	Optional
	Hardware TRNG	Future	Optional

Table 1-1. MSPM0 MCU Platform Security Enablers (continued)

Security Enabler	Device Feature	MSPM0L	MSPM0G
Device identity	Unique device identifier (96-bit)	All	All
Physical security	Boot configuration routine fault injection attack countermeasures	Future	Future

2 Device Security Model

The foundation of the MSPM0 security model is the enforcement of a set of user-specified security policies at boot time. This section provides an overview of the device boot process and the user-specified policies which may be set to enable a wide variety of application use cases.

2.1 Initial Conditions at Boot

During a cold power up (POR), the device is reset to a secure state. The digital IO pins are in a high impedance configuration with all peripheral functions disconnected, the NRST pin is in NRST mode, and the serial wire debug (SWD) interface pins are in SWD mode. Following the release of the brown-out reset, the serial wire debug port (SW-DP) is initially enabled to allow a debug probe to establish an initial connection to the debug subsystem.

At this point in the boot process, the only debug access ports (DAPs) which are accessible by a debug probe are the configuration access point (CFG-AP) and secure access point (SEC-AP). The CFG-AP may be used by a connected debug probe to read generic device information (such as the device generic part number). The SEC-AP may be used to attempt to pass a command message to the boot configuration routine. Application debug access to device (through the AHB-AP, ET-AP, and PWR-AP DAPs) remains blocked by hardware firewalls. As a result, the device hardware does not permit any debug access to the processor, the EnergyTrace state, or the power configuration during device power-up.

Following a brown-out reset (BOR), a boot reset (BOOTRST) is always generated, which starts execution of the boot configuration routine.

2.2 Boot Configuration Routine (BCR)

MSPM0 devices contain an immutable root-of-trust boot configuration routine contained in read-only memory (ROM). The boot configuration routine (BCR) is always the first code to run on the Cortex-M0+ processor following a BOOTRST of the device. The BCR also runs upon software invocation of the bootstrap loader (BSL) as it is needed for authorizing the BSL entry. The core responsibilities of the BCR are to:

1. Load TI factory data needed for proper device operation from the FACTORY flash memory region into logic, and verify the integrity of the factory data (including device trim data) through CRC-32
2. Load the user-specified device configuration (including the security policies) from the NONMAIN flash memory region into logic, and verify the integrity of the user configuration data through CRC-32
3. Check for any boot commands sent over the serial wire debug (SWD) interface, authorize them (if applicable), and process them (if authorized)
4. Check for bootstrap loader (BSL) invocation conditions if the BSL is enabled, and start the BSL if a valid invocation occurred
5. Check the integrity of a portion of the MAIN flash memory region containing the user application code before starting the user application
6. Log any boot errors to the CFG-AP
7. Trigger hardware to start the application by fetching the stack pointer from 0x0000.0000 and the reset vector from address 0x0000.0004 in MAIN flash

During execution of the BCR, the AHB-AP, ET-AP, and PWR-AP DAPs remain inaccessible through the SWD interface. If the user specified security policy allows debug access to the device, then these DAPs will become available when the hardware starts the user application or the bootstrap loader.

2.3 Bootstrap Loader (BSL)

MSPM0 devices may also contain an immutable bootstrap loader (BSL) in read-only memory (ROM). The BSL provides a means to program and verify the contents of the device memory through a standard serial interface (UART or I2C), as opposed to the serial wire debug (SWD) interface.

The BSL can only be started by the BCR. The BCR checks for a valid BSL invoke condition (software invoke, IO pin invoke, blank device invoke) and validates that the BSL is enabled for use before starting the BSL. When the BSL exits, the BCR runs again to load the current device security policies and start the user application.

The BSL is always protected by a 256-bit user-specified password that must be passed to the BSL through the UART or I2C interface when starting a BSL session. The BSL can be disabled if it is not used (see the BSL enable/disable policy).

2.4 Boot Flow

The high level boot flow for MSPM0 devices is given in Figure 2-1.

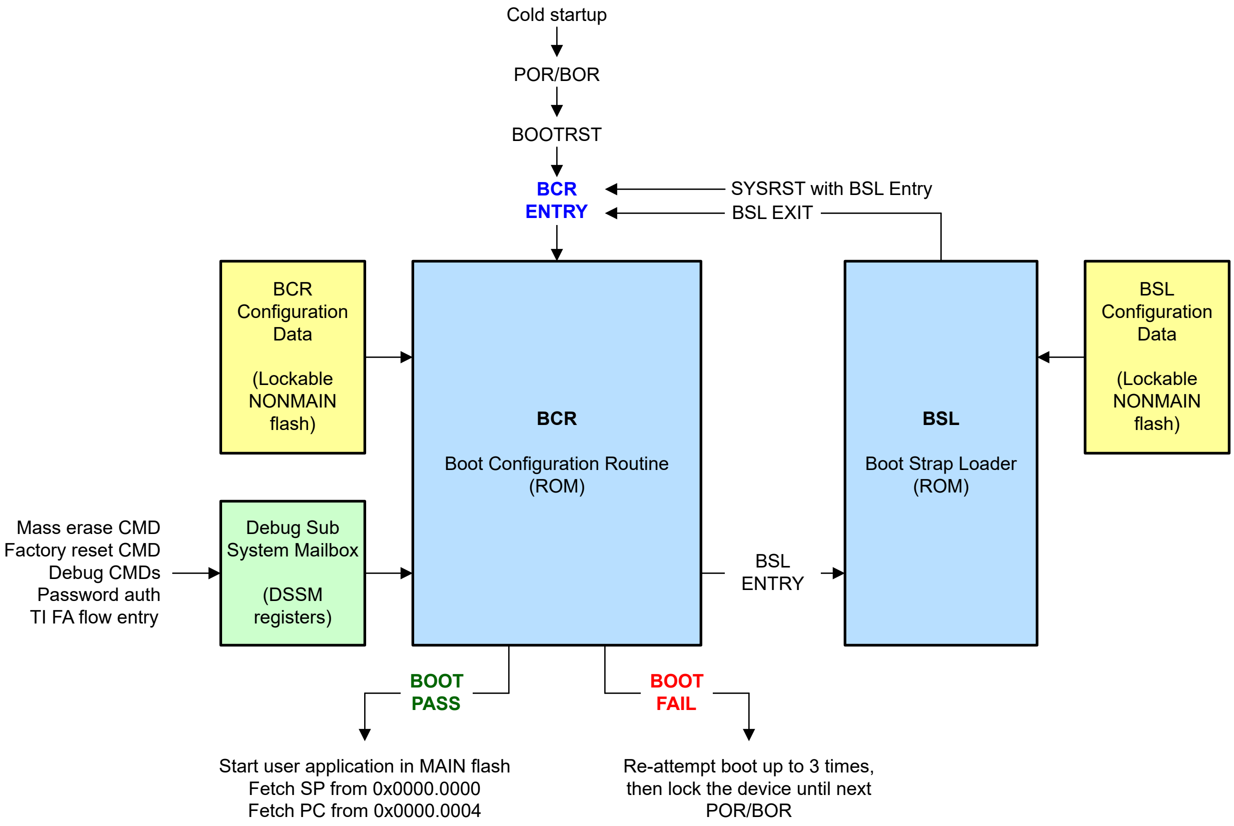

Figure 2-1. High Level Boot Flow

Note that the BCR and BSL both contain user-specified configuration data structures in the lockable NONMAIN flash memory region. These security policies which are specified through these data structures are described in Section 2.5.

2.5 User-Specified Security Policies

MSPM0 devices contain a dedicated region of flash memory for storing user-specified security and device configuration policies. This region is referred to as the NONMAIN flash region. The boot configuration routine (BCR) and bootstrap loader (BSL) reference the user-specified data stored in the NONMAIN flash region to configure the device for operation.

The user must provision the NONMAIN flash memory region of the device with the desired policies during production. This section will introduce the security policies which are user configurable through the NONMAIN configuration memory.

The NONMAIN flash region is partitioned into two distinct data structures:

- The BCR configuration, described in Section 2.5.1, which sets the boot configuration security policies
- The BSL configuration, described in Section 2.5.2, which sets the boot loader security policies

Both data structures are backed by their own 32-bit CRC digests, which are used as a part of the configuration data error resistance scheme.

Note

Additional parameters beyond those shown in this document are included in the BCR and BSL configuration structures; this document focuses on the parameters which are relevant for security. For a complete description of the BCR and BSL configuration structures in the NONMAIN flash memory region, see the boot configuration section of the architecture chapter in the corresponding technical reference manual.

2.5.1 Boot Configuration Routine (BCR) Security Policies

The BCR security policies are interpreted by the BCR, and include the following parameters:

- Serial wire debug related policies, described in Section 2.5.1.1
- Bootstrap loader enable and disable policy, described in Section 2.5.1.2
- Flash memory protection and integrity policies, described in Section 2.5.1.3

2.5.1.1 Serial Wire Debug Related Policies

The serial wire debug related policies configure the functionality which is available through the device's physical debug interface (SWD). By default, MSPM0 devices come from TI in an unrestricted state. This state allows for easy production programming, evaluation, and development. However, this unrestricted state is not recommended for mass production, as it leaves a large attack surface present. To accommodate a variety of needs while keeping the configuration process simple, MSPM0 devices support three generic security levels: no restrictions (Level 0), custom restrictions (Level 1), and fully restricted (Level 2). Table 2-1 shows the three generic security levels, from least restrictive to most restrictive.

There are 4 main uses of the SWD interface for which protection needs to be considered:

- Application debug access, which includes:
  - Full access to the processor, memory map, and peripherals through the AHB-AP
  - Access to the device EnergyTrace+ state information through the ET-AP
  - Access to the device power state controls for debug through the PWR-AP
- Mass erase access, which includes:
  - Ability to send a command through SWD to erase the MAIN memory region while leaving the NONMAIN device configuration memory intact
- Factory reset access, which includes:
  - Ability to send a command through SWD to erase the MAIN memory region and reset the NONMAIN device configuration memory to TI factory defaults (Level 0)
- TI failure analysis access, which includes:
  - Ability for TI to initiate a failure analysis return flow through SWD (note that the TI FA flow always forces a factory reset before FA access is given to TI; this ensures that TI does not have any mechanism to read proprietary customer information stored in the device flash memory when a failure analysis flow is initiated)

Table 2-1. Generic Security Levels

Level	Scenario	SW-DP Policy	App Debug Policy	Mass Erase Policy	Factory Reset Policy	TI FA Policy
0	No restrictions	EN	EN	EN	EN	EN
1	Custom restrictions	EN	EN, EN with PW, DIS	EN, EN with PW, DIS	EN, EN with PW, DIS	EN, DIS
2	Fully restricted	DIS	Don't care (access not possible with SW-DP disabled) [1]			

[1] When the SW-DP policy is **SW-DP disabled**, the mass erase and factory reset policies are a don't care from the point of view of the SWD interface. However, if the bootstrap loader (BSL) is enabled, the mass erase and factory reset policies do impact what functionality is available through the BSL. See the BSL security section for details on securing the BSL.

2.5.1.1.1 SWD Security Level 0

SWD security level 0 is the least restrictive SWD security state. This is the default state of a new device from TI, and it is also the state of a device following a successful factory reset. There are no restrictions on application debug access, mass erase, factory reset, for failure analysis in this state.

When to Use This State

Level 0 is well suited for prototyping and development, as it allows programming of the device memory and debug of the processor and peripherals.

When to Not Use this State

Level 0 should not be used in mass production. An attacker would have full freedom to read the contents of the device memory, manipulate the execution of the device, and possibly change the flash memory contents (depending on the flash memory write protection scheme).

2.5.1.1.2 SWD Security Level 1

SWD security level 1 allows for a customized security configuration. The physical debug port (SW-DP) is left enabled, and each function (application debug, mass erase command, factory reset command, and TI failure analysis) may be individually enabled, disabled, or (in some cases) enabled through password authentication, providing considerable flexibility to tailor the device behavior to specific use-cases.

When to Use This State

Level 1 is well suited for restricted prototyping/development scenarios and for mass production scenarios where the desire is to retain certain SWD functions (such as factory reset and TI failure analysis) while disabling other functions (such as application debug). Common examples of Level 1 customized configurations are given in Table 2-2.

Table 2-2. Examples of Level 1 Configurations

Level 1 Scenario	Configuration			
	App Debug	Mass Erase	Factory Reset	TI FA
This scenario restricts debug access with a user-specified password, but it leaves the factory reset and TI failure analysis available. This configuration allows field debug (with password), and it also allows the device to be brought back to the default "Level 0" state through factory reset.	EN with PW	DIS	EN	EN
This scenario does not allow debug. It does allow factory reset, but only with a user-specified password. This provides a way to open up a device in the field by clearing the MAIN memory contents and bringing the device back to a "Level 0" state if the password is known. Importantly, even if the factory reset password were compromised, it would not be possible for an attacker to read proprietary information in the MAIN flash memory.	DIS	DIS	EN with PW	EN
This scenario does not allow debug and it does not allow TI failure analysis. This prevents TI from performing a factory reset and further FA activities on the device, unless the user executes a factory reset with their user-specified password before returning the devices to TI for FA.	DIS	DIS	EN with PW	DIS

Note

Level 1 is the recommended configuration for most standard production use-cases. For applications which do not require secure boot, TI recommends using Level 1 in production with factory reset left enabled (with password) and TI failure analysis left enabled. In such a configuration, the device may be recovered to a less restrictive state after provisioning either by the user (with password) or by TI (through the failure analysis return flow). In use-cases requiring maximum secure boot assurance, a more restrictive Level 1 or Level 2 may be used for production, with the trade-off that devices may not be recoverable to a less restrictive state once provisioned.

When to Not Use this State

Level 1 should not be used during prototyping if complete access to the device is desired; in such a case, Level 0 should be used instead.

Level 1 should also not be used in a mass production scenario where a maximally restrictive state is desired and no SWD functions are to be enabled; in such a case, Level 2 should be used instead as it directly disables the complete SWD physical interface and minimizes the possibility of misconfiguration.

Note

If a device is configured with application debug and factory reset disabled, the only way for a user to restore debug access to the device is if the user application code provides a mechanism to change the NONMAIN configuration to a less restrictive state. If the NONMAIN is locked through static write protection then the state is not reversible and there is no way for a user to re-gain debug access.

2.5.1.1.3 SWD Security Level 2

SWD security level 2 configures the device in a maximally restrictive state. The physical debug port (SW-DP) is completely disabled, and all of the SWD-accessible functions (application debug, mass erase, factory reset, and TI failure analysis) are not accessible through SWD, regardless of their individual configuration.

When level 2 is selected (SW-DP disabled), the application debug configuration and TI failure analysis configuration fields are don't care fields which do not impact the device configuration.

If the BSL is disabled, then the mass erase and factory reset configuration fields are also don't care fields. However, if the BSL is enabled, then the mass erase and factory reset configuration fields are still used by the BSL to authorize mass erase or factory reset commands originating from the BSL interface.

When to Use This State

Level 2 should only be used for mass production when no further access to any SWD functions is required and a maximally secure state is desired for the device.

When to Not Use this State

Level 2 should not be used in the following cases:

- Future application debug and/or re-programming through SWD may be required
- The user would like TI to be able to perform failure analysis on the device
- The user would like to be able to remove proprietary information from the flash memory by sending a mass erase or factory reset command through SWD

Note

Once a device is configured for level 2 (SW-DP disabled), further access to the device through SWD **is not possible**. The only way to bring a device back to a level 0 or level 1 state with SWD access restored is if the BSL and factory reset are both enabled (allowing a BSL factory reset command to be sent), or a mechanism in the user application code is included which can change the NONMAIN configuration to a less restrictive state. In either scenario, if the NONMAIN is locked through static write protection then the level 2 state is not reversible and there is no way to re-gain SWD access.

2.5.1.2 Bootstrap Loader (BSL) Enable/Disable Policy

The bootstrap loader (BSL) provides a means to program and verify the device memory through a standard serial interface (UART or I2C), as opposed to the serial wire debug interface. The BSL has its own configuration policy, but the BCR determines if the BSL is enabled to be invoked, or if it is to be disabled (non-invokable).

Since the BSL presents an additional attack surface, if it is not used in an application it may be disabled in the user-specified boot security policies. If the BSL is used in an application, then the BSL security settings (including the BSL access password) are managed in the BSL configuration policy.

2.5.1.3 Flash Memory Protection and Integrity Related Policies

The flash memory protection and integrity policies specify which sectors of flash memory are locked from modification, as well as which sectors are to be checked for integrity during the boot process before the user application is started.

2.5.1.3.1 Locking the Application (MAIN) Flash Memory

MSPM0 MCUs implement a static write protection scheme to lock out user defined sectors in the MAIN flash region from any program/erase operations at runtime. The desired static write protection scheme is configured as a part of the boot security policies in the NONMAIN flash region.

Purpose

Static write protection enables placement of a fixed, user-defined, application in the flash memory which has the following characteristics:

- Once programmed and locked, it is not modifiable by the application code or ROM bootloader
- If placed at the beginning of the flash memory, it is guaranteed to always be the first code which executes when the ROM boot configuration routine transfers execution to the user application

MSPM0 static write protection supports both characteristics, which must be satisfied to implement a secure boot image manager.

Capabilities

Any sector which is configured in the NONMAIN to be write-locked will be functionally immutable when the boot configuration routine transfers execution to either the bootstrap loader or the user application code in MAIN flash. Any attempt to program or erase a statically protected sector by the application code or the bootstrap loader will result in a hardware flash operation error, and the sector will not be modified.

While static write protection prevents any modification by application code or the boot loader, a mass erase or factory reset command sent through the SWD interface would be honored. If this behavior is not desired, the mass erase and/or factory reset SWD commands may be protected with unique passwords or disabled altogether (see the SWD policies). To completely remove any means of modifying statically write protected MAIN flash sectors, the mass erase and factory reset commands (or the SW-DP) must be disabled, and the NONMAIN boot configuration memory must also be statically write protected to prevent application code from changing the underling write protection scheme by modifying the contents NONMAIN region. This is discussed in the following section.

2.5.1.3.2 Locking the Configuration (NONMAIN) Flash Memory

MSPM0 MCUs implement a static write protection scheme to lock out the NONMAIN flash region from any program/erase operations at runtime. The write protection scheme is configured as a part of the boot security policies in the NONMAIN flash region.

Purpose

By default from TI, the NONMAIN configuration memory (which contains the user-specified boot security policies and bootstrap loader policies) is not write protected. This enables the NONMAIN to be erased by the user during provisioning and re-programmed with the user-specified policies which will be used in mass production.

In many cases, it is desirable for the configuration memory to be locked once it has been provisioned. Locking the configuration memory has the benefit of preventing any unauthorized modification of the security policies, bootstrap loader policies, and static write protection policies by either the bootstrap loader or the application code itself. In most applications, devices in mass production do not require modification of the configuration memory, even when the device firmware is updated.

Capabilities

When configured to be protected, the entire NONMAIN region will be write-locked and will be functionally immutable when the boot configuration routine transfers execution to either the bootstrap loader or the user application code in MAIN flash. Any attempt to program or erase the NONMAIN by the application code or the bootstrap loader will result in a hardware flash operation error, and the sector will not be modified.

While static write protection prevents any modification by application code or the boot loader, a factory reset command sent through the SWD interface would still be honored. If this behavior is not desired, the factory reset SWD command may be protected with a unique password or disabled altogether (see the SWD policies). To completely remove any means of modifying the NONMAIN configuration memory, the factory reset command and TI FA (or the SW-DP) must be disabled.

Note

When the NONMAIN is statically write protected, and the factory reset command and TI FA (or the SW-DP) are disabled, the NONMAIN is equivalent to immutable read-only memory, and it is no longer possible to change the device configuration by any means. Further, if any MAIN memory region sectors are configured with static protection, these sectors also can not be modified by any means and may be considered as immutable.

2.5.1.3.3 Verifying Integrity of Application (MAIN) Flash Memory

The BCR supports checking the data integrity of a user-specified address range in the MAIN flash memory before transferring execution from the BCR (in ROM) to the user application (in MAIN flash memory).

Purpose

The integrity check may be used as an additional step to ensure the code which runs first after the boot ROM (usually the secure boot image manager) has a CRC digest that matches the expected value. This integrity check reduces the likelihood that any unexpected corruption of critical code in the flash memory (which may be responsible for authenticating the remaining user application software image) can create a security vulnerability.

Capabilities

A start address, length, and ISO-3309 CRC-32 digest may be provisioned into the NONMAIN configuration memory. During the boot process, the BCR will compute the CRC-32 digest of the specified range in the MAIN flash memory, and verify the computed digest against the provisioned (expected) digest. If the values match, the user application is started. If the values do not match, the user application is not started and the result is a catastrophic boot error.

2.5.2 Bootstrap Loader (BSL) Security Policies

The BSL security policies are interpreted by the boot loader when it is invoked, and include the following parameters:

- BSL access password, described in Section 2.5.2.1
- BSL read-out policy, described in Section 2.5.2.2
- BSL security alert policy (tamper detection), described in Section 2.5.2.3

2.5.2.1 BSL Access Password

Access to the BSL is always protected by a 256-bit user-specified password. There is no option to disable the password. The password must be provided to the BSL after invocation for access to most BSL functions to be granted. When the password is not provided, the only BSL commands allowed are *Get Identity* and *Start Application*.

If a wrong password is provided to the BSL, the BSL halts for 2 seconds, after which an additional attempt can be made to send the correct password. After three failed password attempts, the security alert function is activated (see Section 2.5.2.3).

2.5.2.2 BSL Read-out Policy

The BSL optionally supports read-out of the device memory for debug and/or diagnostic purposes (after access to the BSL has been granted with a correct password match). By default, this capability is disabled for security to prevent extraction of sensitive code and/or data from the device. When the BSL read-out policy is disabled, the only information which may be provided to a host through the BSL interface is a CRC32 digest of a memory segment with a minimum segment length of 1 kilobyte. If direct read-out of the device memory is desired, it may be enabled in the BSL configuration.

2.5.2.3 BSL Security Alert Policy

The BSL provides an alert mechanism for taking action when tampering is suspected. Specifically, if an incorrect password is passed to the BSL 3 times during one BSL session, the security alert is activated and the BSL may respond in one of three different ways based on the specified security alert policy:

1. Issue a factory reset (erasing the MAIN flash and resetting the NONMAIN flash regions)
2. Disable the BSL (leaves the MAIN flash intact but re-configures the NONMAIN to block BSL access)
3. Ignore (do not modify the configuration and allow password attempts to continue)

Note

Options 1 and 2 require that the NONMAIN flash region not be statically write protected (see Section 2.5.1.3.2).

When option 1 is selected, any MAIN memory region which is configured to be statically write protected (see Section 2.5.1.3.1) will not be erased during the factory reset.

2.5.3 Configuration Data Error Resistance

MSPM0 devices employ several mechanisms to reduce the possibility of data errors in the NONMAIN configuration memory from leading to a loss of security.

2.5.3.1 CRC-Backed Configuration Data

The BCR configuration data and BSL configuration data structures in the NONMAIN memory each include a CRC32 value corresponding to the CRC32 digest of the respective structure. During the device boot process, the BCR will compute the CRC digest of the data structures and compare it with the stored CRC values before the data contained within the structures is trusted for use.

BCR Configuration CRC Fail Handling

In the event that the BCR configuration data (which contains the SWD policies, BSL enable/disable policy, and flash memory protection and integrity check policies) fails its CRC check during boot, a catastrophic boot error results and the following limitations are imposed:

- The error cause will be logged in the CFG-AP as a boot diagnostic
- The BSL will not be invoked, even if it was configured to be enabled
- The user application is not started
- No application debug access is enabled
- A pending SWD factory reset command, if enabled or enabled with password, is honored
- A pending TI failure analysis flow entry, if enabled, is honored
- The boot process will re-attempt up to 3 times
  - If the 2nd or 3rd attempt pass, the device boots normally
  - If the 3rd attempt does not pass, no further boot attempts are made until the next BOR or POR

The benefit of the this CRC check is that any bit flips in configuration data, such as the static write protection configuration (which is a pillar of secure boot), may be detected with high confidence during the boot process. The fail handling procedure explicitly prevents the BSL and user application from running, and the only supported options (SWD factory reset and TI FA) are protected by 16-bit pattern-match fields.

BSL Configuration CRC Fail Handling

If the BSL configuration data (which contains the BSL password and BSL policies) fails the CRC check during BSL invocation, a catastrophic boot error results and the following limitations are imposed:

- The error cause is logged in the CFG-AP as a boot diagnostic
- The BSL is not invoked, even if it was configured to be enabled
- The user application is not started
- No application debug access is enabled
- The boot process re-attempts up to 3 times
  - If the 2nd or 3rd attempt pass, the device boots normally
  - If the 3rd attempt does not pass, no further boot attempts are made until the next BOR or POR

The benefit of this CRC check s that any bit flips in the BSL configuration data may be detected with high confidence during the invoke process. The failure handling procedure prevents the BSL from starting with invalid data which could lead to a loss of security.

TI Factory Trim Data CRC Fail Handling

In addition to the user-specified configuration data, if the TI factory trim fails its CRC check during boot, a catastrophic boot error will also result with the following limitations:

- The error cause will be logged in the CFG-AP as a boot diagnostic
- The BSL will not be invoked, even if it was configured to be enabled
- The user application is not started
- No application debug access is enabled
- A pending TI failure analysis flow entry, if enabled, is honored
- The boot process will re-attempt up to 3 times
  - If the 2nd or 3rd attempt pass, the device boots normally
  - If the 3rd attempt does not pass, no further boot attempts are made until the next BOR or POR

2.5.3.2 16-bit Pattern Match for Critical Fields

Critical policies in the BCR configuration memory, such as the SWD security policies, are implemented as 16-bit pattern-match fields in the NONMAIN memory, with the following characteristics:

- An exact pattern match is required to enable lower security states
- Any value in the 16-bit field not matching the exact defined patterns results in a maximally secure state for the respective parameter

This behavior prevents single bit flips from causing the device to enter a lower security state than that which was originally specified.

3 Secure Boot

The MSPM0 devices support authentication of application software (secure boot) through a combination of hardware and software features. Asymmetric and symmetric based authentication schemes are supported, although not all MSPM0 devices provide secure storage to protect symmetric keys from software exploits.

The MSPM0 architecture includes several key hardware features needed to enable secure boot:

- Lockable flash memory for storing fixed authentication firmware and authentication keys
- Single point of entry during boot, ensuring that the secure boot image manager is always the first application to run after the BCR

The MSPM0 software development kit (SDK) includes a boot image manager (BIM) reference application for implementing secure boot on MSPM0 MCUs. This reference application may be easily configured and provisioned into MSPM0 devices.

3.1 Secure Boot Authentication Flow

The following provisioning steps are required to prepare a device to support secure boot:

1. The boot image manager firmware must be configured and programmed into the MAIN flash memory, with the reset vector at 0x0000.0004 pointing to the start of the boot image manager
2. Any authentication key material needed by the boot image manager must be programmed into the MAIN flash memory, adjacent to the boot image manager
3. The device NONMAIN configuration memory must be programmed with the following characteristics:
    a. The MAIN flash sectors containing the boot image manager firmware and key material must be configured as statically write protected to prevent modification.
    b. The NONMAIN flash sector must be configured as statically write protected to prevent modification.
    c. The mass erase and factory reset commands are recommended to be password protected or disabled (disabling factory reset with the above configuration settings will result in the NONMAIN configuration becoming permanently locked, together with the sectors containing the boot image manager and authentication keys.
    d. The MAIN flash memory integrity check is recommended to be enabled, with the address range set to include the boot image manager and authentication keys.

Once provisioned, and once signed firmware is programmed into the device, the secure boot flow from device power-up is as follows:

1. During power-up, the device is in a maximally secure state. The BCR will check the integrity of the device configuration memory and load the user-specified policies accordingly if the device configuration is valid.
2. The BCR will compute the CRC value corresponding to the MAIN flash memory containing the BIM and key material. If the CRC check passes, the BCR will transition execution to the first user code (the boot image manager).
3. The boot image manager will compute the digest of the remaining application code:
    a. In the case of asymmetric authentication, the secure hash (SHA2-256) digest of the application code will be computed in software
    b. In the case of symmetric authentication, the CMAC message authentication code corresponding to the application code will be computed using the authentication key
4. The boot image manager will validate the digest against the provided signature:
    a. In the case of asymmetric authentication, the digital signature will be decrypted in software using the elliptic curve digital signature algorithm (ECDSA), and the result will be compared with the computed hash
    b. In the case of symmetric authentication, the computed CMAC will be compared with the CMAC in the digital signature
5. If the application code digest matches the signature, the application code is started, else a user-specified failure handler will be invoked.

3.2 Asymmetric vs. Symmetric Secure Boot

While the boot image manager provided in the MSPM0 SDK supports both asymmetric and symmetric secure boot, there are trade-offs between the two implementations which should be carefully weighed for a given application. Table 3-1 gives the trade-offs between the two alternatives.

Table 3-1. Secure Boot Algorithm Comparison

Parameter	Asymmetric (SHA2+ECDSA)	Symmetric (CMAC)
Authentication time	Longer, due to software hash computation and public key arithmetic	Shorter, due to simplicity of algorithm and ability to leverage hardware AES acceleration when available
Code size	Larger, due to SHA and ECDSA algorithms	Smaller, especially if AES acceleration is available on the target device
Key integrity	Public keys must be provisioned into the device and must be immutable	Shared keys must be provisioned into the device and must be immutable
Key confidentiality	Public keys have no confidentiality requirement and there is no need for protecting the public key from vulnerabilities in application code	Shared keys must be kept confidential, and should be wrapped when not in use and secured with a static read firewall (if supported by the target device) to protect the shared key from vulnerabilities in application code

TI recommends the asymmetric implementation in most situations. In cases where code size is limited and/or authentication time must be kept to a minimum, the symmetric implementation may be used, with the trade-off that the shared key must be managed carefully. Not all devices provide secure storage to protect shared symmetric keys from software vulnerabilities.

4 Cryptographic Acceleration

Certain MSPM0 MCUs offer hardware acceleration for the advanced encryption standard (AES), as well as hardware for generating true random numbers for cryptographic purposes (TRNG). See the device specific data sheet to determine if a device has an AES accelerator or TRNG, or refer to Appendix A.

4.1 Hardware AES Acceleration

Certain MSPM0 devices include hardware acceleration for the advanced encryption standard (AES). See the device-specific data sheet to determine if a particular device includes hardware AES acceleration.

4.1.1 Overview

The AES accelerator module performs encryption and decryption of 128-bit data blocks with a 128-bit or 256-bit key in hardware according to the advanced encryption standard (AES). AES is a symmetric-key block cipher algorithm specified in FIPS PUB 197.

The AES accelerator features include:

- AES 128-bit block encryption and decryption
- DMA trigger support for automating ECB, CBC, OFB, and CFB block cipher modes as defined in NIST SP 800-38
- Support for accelerating CTR cipher mode by encrypting precalculated (nonce || counter) blocks and accelerating XOR of plaintext with the generated key stream
- Support for accelerating CBC-MAC tag computation (CBC DMA mode with zero initialization vector)
- On-the-fly key expansion for encryption and decryption
- Offline key generation for decryption
- Shadow register storing the initial key for all key lengths
- 8-bit byte or 32-bit word access to provide key data, input data, and output data
- AES ready interrupt
- Supported in RUN and SLEEP (see the *Operating Modes* section of the device technical reference manual)

The AES accelerator hardware consists of the 128-bit state memory and associated input/output registers, the AES encryption/decryption core and control logic, and the 256-bit AES key memory and associated input register. The AES hardware is shown in Figure 4-1.

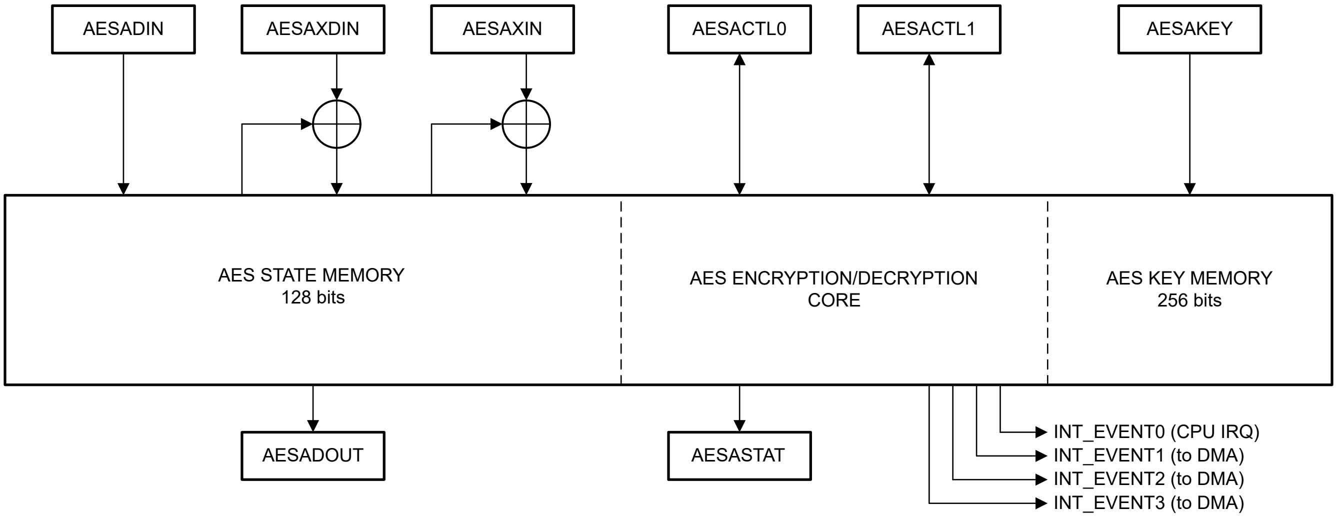

Figure 4-1. AES Accelerator Block Diagram

4.1.2 AES Performance

The AES accelerator provides fast encryption and decryption of 128-bit blocks. AES accelerator performance in both cycles and execution time for block encryption and block decryption (with pregenerated decryption key) is given in Table 4-1.

Table 4-1. AES Hardware Accelerator Key Performance Metrics

AES Key Length	Encryption (OPx==0x0)			Decryption (OPx==0x3)		
	Cycles	Time (32 MHz)	Time (80 MHz)	Cycles	Time (32 MHz)	Time (80 MHz)
128-bit	168	5.25 µs	2.10 µs	168	5.25 µs	2.10 µs
256-bit	234	7.31 µs	2.93 µs	234	7.31 µs	2.93 µs

4.2 Hardware True Random Number Generator (TRNG)

Certain MSPM0 devices include a hardware true random number generator (TRNG) block. The TRNG may be used to easily generate true random seed values which may be used to seed a deterministic random bit generator (DRBG).

The TRNG module provides 32-bit true random outputs based on a delta-sigma modulation based analog entropy source inside the device. A dedicated regulator is provided local to the TRNG to protect against power manipulation attacks.

Integrated heath tests provide power-on self-test of the analog and digital components of the TRNG, and continuous monitoring is provided through statistical self-tests.

The TRNG is suitable for use in creating a TRNG + DRBG system which can pass the NIST SP800-22 statistical test suite for cryptographic random number generators. A block diagram of the TRNG is given in Figure 4-2.

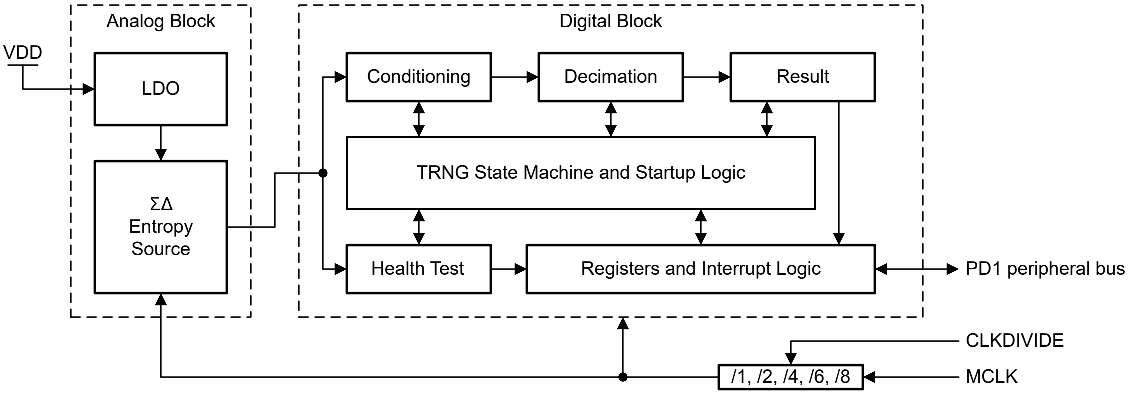

Figure 4-2. TRNG Block Diagram

For more information on the operation of the TRNG, refer to the device family technical reference manual.

5 Device Identity

All MSPM0 devices include a 96-bit unit-specific identification code (device ID), which can be read by application software. See the technical reference manual and device data sheet for more information on the device ID.

The device ID is designed by TI to be unique for each unit which is shipped, and as such it can be used to identify or distinguish a particular unit from any other unit. While the device ID is unique, it is not cryptographically random, as some of the bits correspond to device characteristics such as the part number and product revision.

6 Summary

The security enablers offered in MSPM0 MCUs offer a unique blend of capability and value for MCU customers looking to add more cybersecurity capabilities to new applications. Distinctive features at the price point (such as password authenticated application debug, mass erase, and factory reset) enable a variety of development and production use-cases while keeping configuration simple and straightforward.

7 References

- TI Security E-book (SWPB021)
- TI security portal (link)
- MSPM0G technical reference manual (SLAU846)
- MSPM0L technical reference manual (SLAU847)

8 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
January 2023	*	Initial Release

A Security Enablers by Subfamily

The security enablers including in a given MSPM0 subfamily are listed in Table A-1. Note that certain features are planned for future MSPM0 devices and may not be included in the devices families shown in the table.

Table A-1. Security Enablers by MSPM0 Subfamily

<table>
<tr><th>Security Enabler</th><th>Security Enabler</th><th>MSPM0L110x</th><th>MSPM0L13xx</th><th>MSPM0G110x</th><th>MSPM0G150x</th><th>MSPM0G3x0x</th></tr>
<tr><td rowspan="9">Debugging security</td><td>Password authenticated debug access</td><td colspan="5">Yes</td></tr>
<tr><td>Password authenticated boot strap loader access</td><td colspan="5">Yes</td></tr>
<tr><td>Password authenticated main flash memory mass erase</td><td colspan="5">Yes</td></tr>
<tr><td>Password authenticated complete factory reset</td><td colspan="5">Yes</td></tr>
<tr><td>TI failure analysis (FA) enable/disable</td><td colspan="5">Yes</td></tr>
<tr><td>Complete hardware disable of serial wire debug (SWD) interface</td><td colspan="5">Yes</td></tr>
<tr><td>Permanently lockable device configuration data</td><td colspan="5">Yes</td></tr>
<tr><td>Error resistant device configuration data</td><td colspan="5">Yes</td></tr>
<tr><td>Password memory contains hashes only (SHA2-256)</td><td colspan="5">No</td></tr>
<tr><td rowspan="7">Secure boot</td><td>Permanently lockable main flash memory (static write protection)</td><td colspan="5">Yes</td></tr>
<tr><td>CRC-32 verified main flash region</td><td colspan="5">Yes</td></tr>
<tr><td>SHA2-256 verified main flash memory region</td><td colspan="5">No</td></tr>
<tr><td>Single point of entry to main flash application at boot</td><td colspan="5">Yes</td></tr>
<tr><td>Firmware image authentication routines (asymmetric or symmetric)</td><td colspan="5">Yes</td></tr>
<tr><td>Lockable flash for key revocation and rollback protection</td><td colspan="5">No</td></tr>
<tr><td>SRAM W^X (write-or-execute) boundary enforcement</td><td colspan="5">Yes</td></tr>
<tr><td rowspan="4">Secure Storage</td><td>Static flash memory read/execute (RX) firewall</td><td colspan="5">No</td></tr>
<tr><td>IP protection (execute-only) firewall</td><td colspan="5">No</td></tr>
<tr><td>W^X (write-or-execute) enforcement on main flash banks</td><td colspan="5">No</td></tr>
<tr><td>AES volatile key store (up to four 128-bit keys plus a session key)</td><td colspan="5">No</td></tr>
<tr><td rowspan="2">Cryptographic acceleration</td><td>Hardware AES accelerator (128-bit / 256-bit)</td><td colspan="3">No</td><td colspan="2">Yes</td></tr>
<tr><td>Hardware TRNG</td><td colspan="3">No</td><td colspan="2">Yes</td></tr>
<tr><td>Device identity</td><td>Unique device identifier (96-bit)</td><td colspan="5">Yes</td></tr>
<tr><td>Physical security</td><td>Boot configuration routine fault injection attack countermeasures</td><td colspan="5">No</td></tr>
</table>

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265