AxioCrypto-M235x v2.1.0

Non-propriety Security Policy

Version:	v2.7
Date:	March 21, 2023
Classification:	Public

Security Platform Inc.

Suite #771, 7F, Pangyo Start-up Zone,

815 Daewangpangyo-ro, Sujeong-gu,

Seongnam-si, Gyeonggi-do 13449 Korea

EDITOR

Author	Title
Kyung-mo Kim	CTO

Revision History

Version	Description	Date	By
1	Initial Version	Apr/04/2020	Kyung-mo Kim
1.1	Cryptographic boundary complemented	May/19/2020	Kyung-mo Kim
2.1	Axiocrypto-M235x v2	Jan/23/2021	Kyung-mo Kim
2.3	Additional CAVP	Jun/09/2021	Kyung-mo Kim
2.5	Final	Jul/23/2021	Kyung-mo Kim
2.6	NIST's comments	Jun/30/2022	Kyung-mo Kim
2.7	NIST's comments	Mar/21/2023	Kyung-mo Kim

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This is a non-proprietary security policy for the AxioCrypto-M235x v2.1.0 of Security Platform Inc. This Security Policy describes how the cryptographic module meets the requirements for a FIPS140-2 level 1 validation as specified in the FIPS140-2 standard. This Security Policy is part of the evidence documentation package to be submitted to the validation lab.

FIPS140-2 specifies the security requirements for a cryptographic module protecting sensitive information. Based on four security levels for cryptographic modules this standard identifies requirements in eleven sections. For more information about the standard, please visit http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

1.2 Scope

This Security Policy specifies the security rules under which the cryptographic module operates its major properties. It does not describe the requirements for the entire system, which makes use of the cryptographic module.

1.3 Security Level

The module meets the overall requirements applicable to FIPS140-2 Security Level 1. In the individual requirement sections of FIPS140-2 the following Security Level ratings are achieved:

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	1
6	Operational Environment	N/A
7	Cryptographic Key Management	1

Section	Section Title	Level
8	EMI/EMC	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

Table 1 Security Level per FIPS140-2 Section

2 Cryptographic Module Specification

The following section describes the cryptographic module and how it conforms to the FIPS140-2 specification in each of the required area.

2.1 Module Overview

The AxioCrypto-M235x cryptographic module (hereafter referred to as “the module”) is designed to provide foundational security services for the platform, including secure boot, secure lifecycle state, platform identity and key management. It offers high-throughput cryptography operations, suitable for diverse set of use cases, such as TLS link protection, key protection, sensor data encryption, device identification and more.

2.2 Module Embodiment

The module is a sub-chip cryptographic subsystem and firmware-hybrid module. AxioCrypto-M235x is integrated in Nuvoton M235x SoC. M235x SoC contains internal SRAM and flash memory. This flash memory is mapped in address space and the firmware code can execute directly in the flash memory. There is no need to copy code into RAM to execute it. SRAM is for stack, heap and global variables.

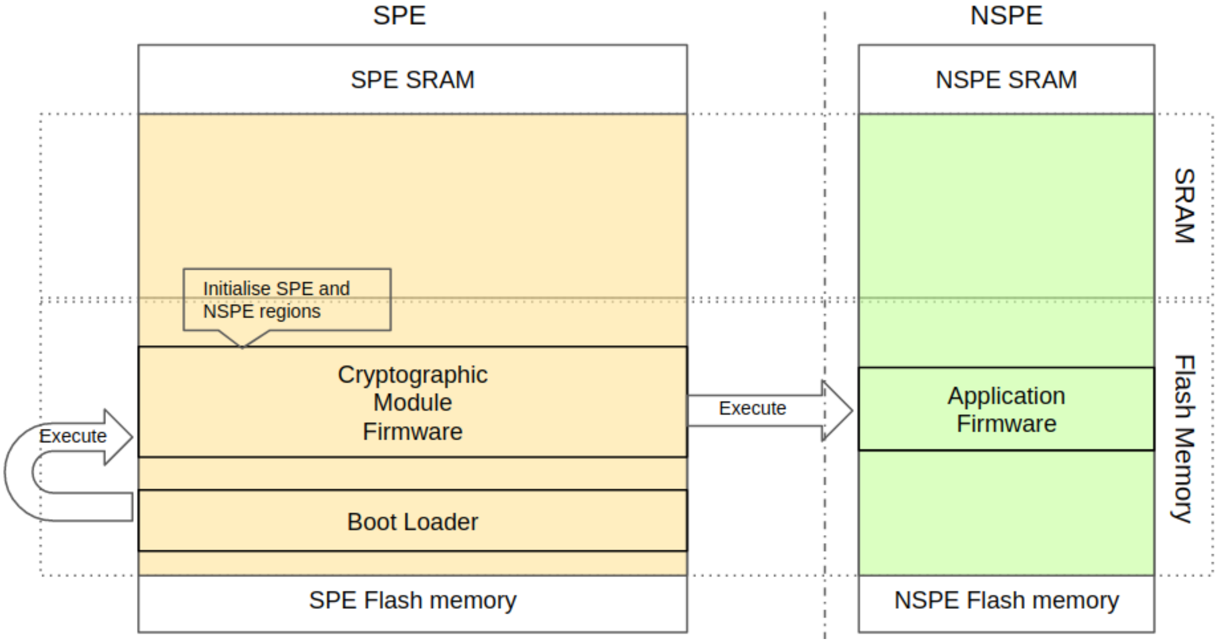

Figure 1 Structure of the memory and the Initialization of the module

Both SRAM and flash memory are divided to two environments: Secure Processing Environment (SPE) and Non-Secure Processing Environment (NSPE). When powered on, boot loader runs first and run SPE firmware. When SPE runs, it initializes the module and run NSPE firmware. Both of SRAM and flash memory are divided when SPE initializes the module and these regions are not allowed to change while the system continues operation.

When NSPE firmware runs, because SPE firmware doesn’t support task, the way to use SPE firmware’s service is using Non-Secure Callable (NSC) function. NSC function is located in SPE region of flash memory. When NSPE firmware calls an NSC function, the first instruction called is SG instruction. SG instruction means “Secure Gateway”. When branching to a secure gateway from Non-secure state, the SG instruction switches to the Secure state. The SG instruction executed in NSPE region is useless because it does not change the security state to “Secure state”. If applications in NSPE try to call functions in SPE directly, exception will occur and the system halts. All the sensitive information is protected in SPE, against NSPE.

The hardware isolation technology enforces separation of data and control between the two environments containing the hardware and the firmware components. The firmware in the SPE is the module firmware, which is compact and strictly controlled. The firmware in NSPE is application firmware, which manages and runs diverse peripherals and features.

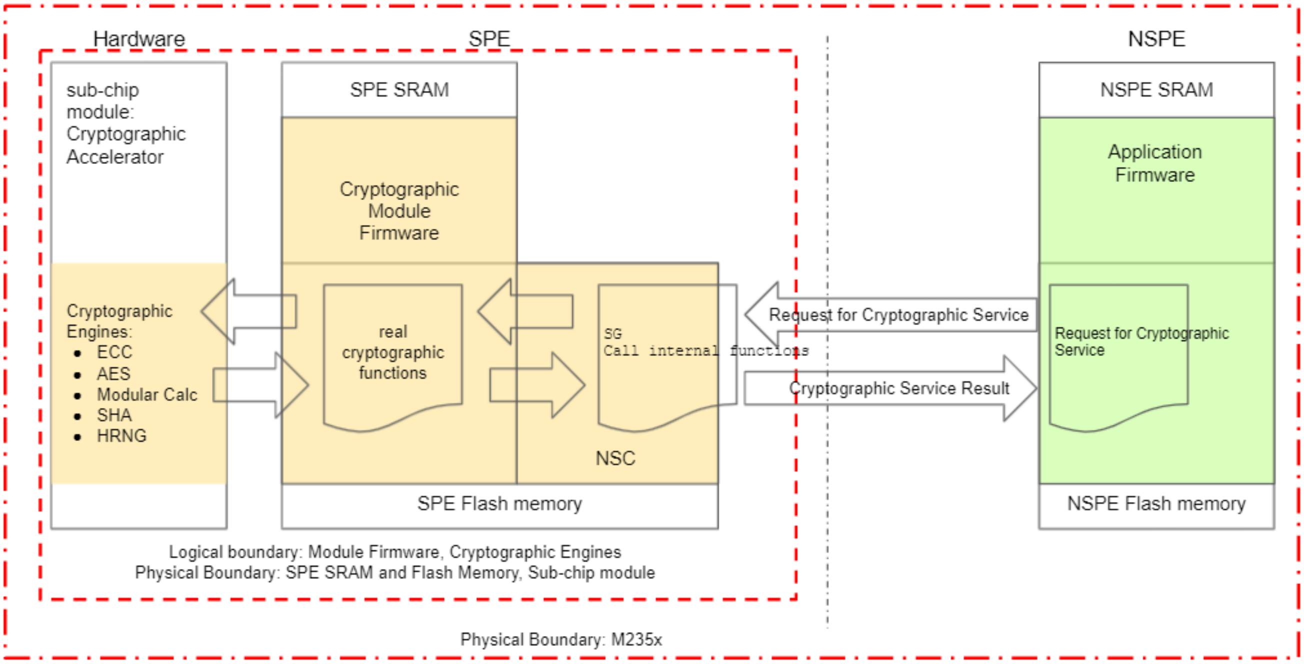
Hardware
SPE
NSPE
sub-chip module: Cryptographic Accelerator
SPE SRAM
Cryptographic Module Firmware
NSPE SRAM
Application Firmware
Cryptographic Engines:
• ECC
• AES
• Modular Calc
• SHA
• HRNG
real cryptographic functions
SG
Call internal functions
Request for Cryptographic Service
Request for Cryptographic Service
Cryptographic Service Result
NSC
SPE Flash memory
NSPE Flash memory
Logical boundary: Module Firmware, Cryptographic Engines
Physical Boundary: SPE SRAM and Flash Memory, Sub-chip module
Physical Boundary: M235x

Figure 2 AxioCrypto-M235x High Level Diagram

The application firmware in NSPE can call NSC API to use cryptographic services. NSC API calls internal functions in SPE, which executes real cryptographic functions. In those internal functions, time-consuming functions such as ECC point and modular calculations and AES block operation are transferred to hardware cryptographic accelerators and returns the result.

2.2.1 Module Hardware

The hardware portion of the cryptographic module is the cryptographic engine of Nuvoton M235x SoC. Details are addressed in 2.3.3.2.

The hardware version is: 1.0.

2.2.2 Module Firmware

Axiocrypto-M235x has an API layer that provides consistent interfaces to the supported algorithms. Details are described in 2.3.3.1.

The firmware version supported by the module is: 2.1.0.

2.2.3 Test Configuration

The module has been tested on the following Configuration.

Module Name	Hardware Version / Test SoC	Test Platforms
AxioCrypto-M235x	v1.0 / M2354KJFAE	• Embedded proprietary OS running on Nuvoton M2351 SoC with Arm Cortex-M23; • Embedded proprietary OS running on Nuvoton M2354 SoC with Arm Cortex-M23.

Table 2 Test Platform of the module

2.3 Cryptographic Module Boundary and Components

The physical boundary of the module is the physical boundary of Nuvoton M235x SoC that contains the sub-chip hardware which implements the cryptographic engine and the host processor which executes the module firmware. The sub-chip hardware also has a sub-chip boundary which is defined as the set of hard circuitry cores that comprises the sub-chip cryptographic subsystem.

The physical boundary of the firmware is the platform on which the firmware and operating systems reside. The logical boundary of the firmware is the set of binary files that make up the firmware.

2.3.1 Software Block Diagram

SPE, Non-Secure Callable API, and interface to cryptographic accelerator make the module firmware. The SPE comprises of SPE regions of SRAM and Flash Memory. Module firmware resides in the SPE region of flash memory and uses SPE parts of SRAM for stack, heap and the space for global variables.

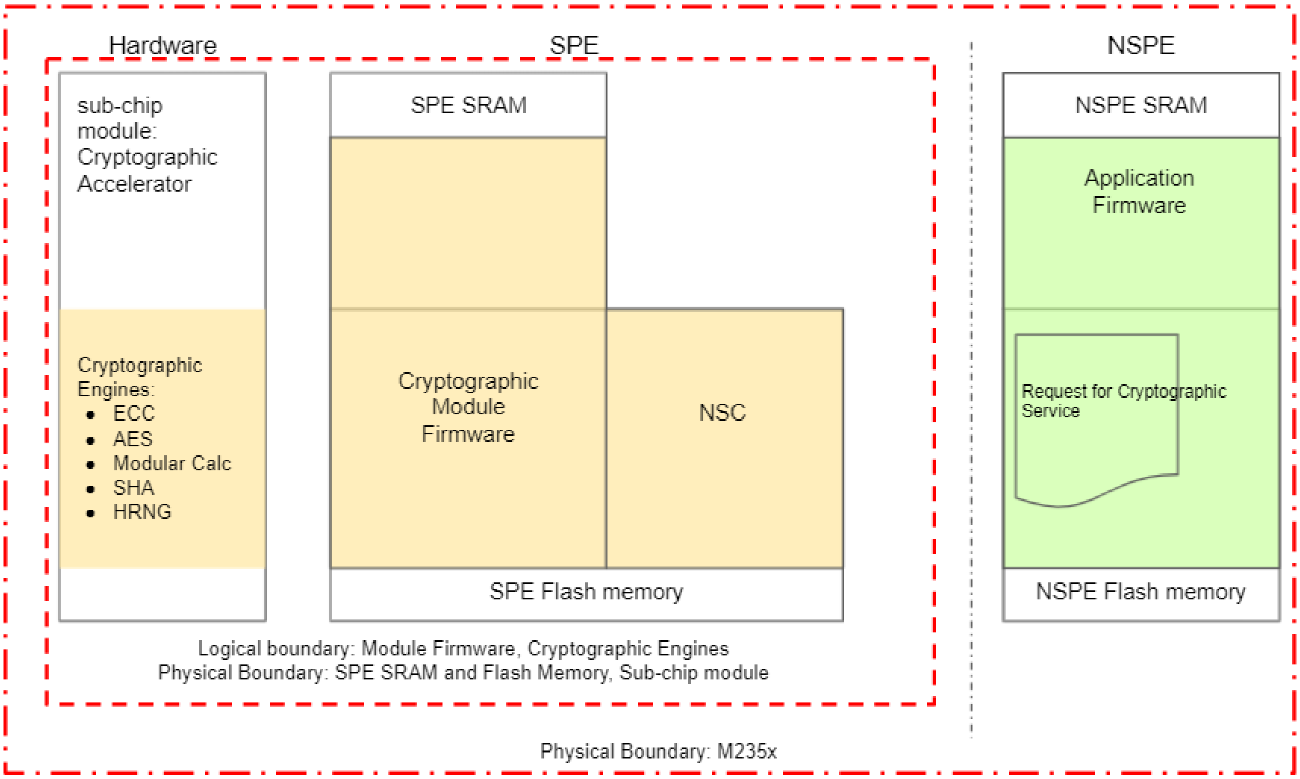
Hardware
SPE
NSPE
sub-chip module: Cryptographic Accelerator
SPE SRAM
NSPE SRAM
Application Firmware
Cryptographic Engines:
ECC
AES
Modular Calc
SHA
HRNG
Cryptographic Module Firmware
NSC
Request for Cryptographic Service
SPE Flash memory
NSPE Flash memory
Logical boundary: Module Firmware, Cryptographic Engines
Physical Boundary: SPE SRAM and Flash Memory, Sub-chip module
Physical Boundary: M235x

Figure 3 Cryptographic Module Logical Boundary

All the logical interfaces are in NSC. Control input, data input, data output and status output interfaces make NSC API. There is no other interface for date IO.

CSPs are generated in the module or provisioned from NSPE firmware. Any CSPs stored in the module are not exposed outside the module. All components which can handle plaintext CSPs are module firmware and cryptographic accelerator, all of these are inside the logical boundary of the module.

Once cryptographic key is stored, the key is referenced from NSPE by the number of slot where the key is stored. When NSC API is called, the key is decrypted from the storage and used inside the module. When the NSC API returns, CSPs in SRAM are zeroized.

The perimeter of the module forms the cryptographic boundary of this FIPS140-2 Security Level 1 compliant single-chip cryptographic module.

2.3.2 Hardware Block Diagram

This module hardware components are listed as following:

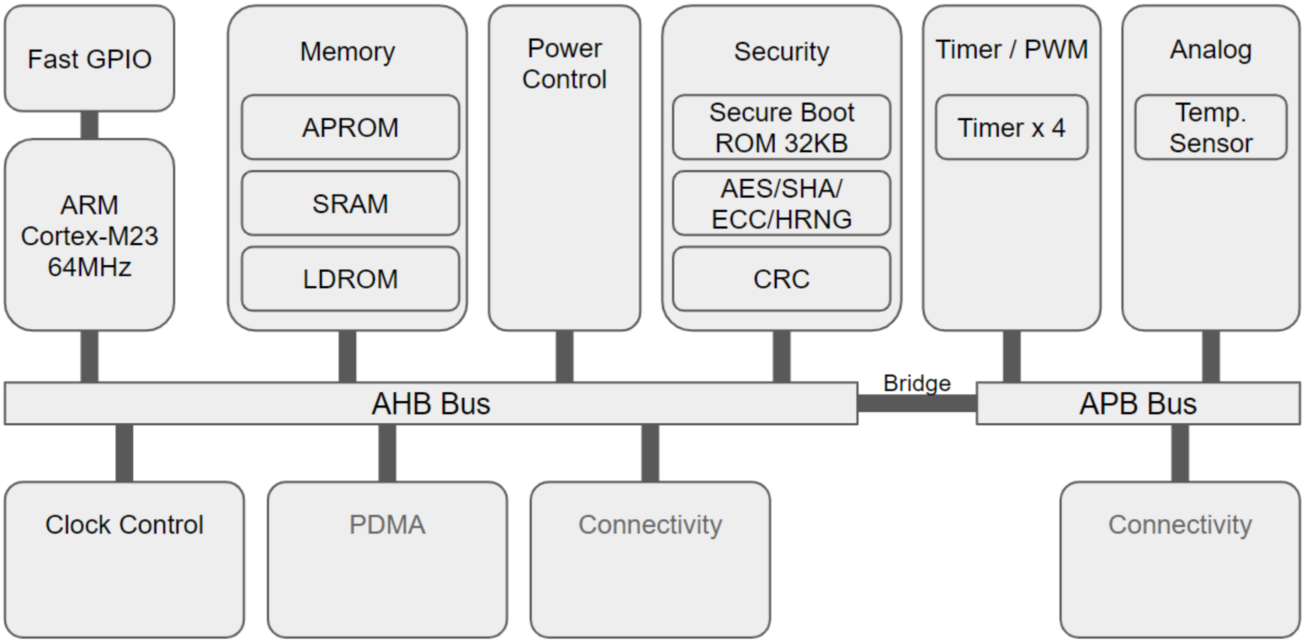

Figure 4 Hardware Components

“ARM Cortex-M23” core runs module firmware located in APROM. APROM is internal flash memory of M235x SoC. When the core runs firmware, it uses SRAM for stack, heap and spaces for global variables.

Power and Clock control is essential for SoC operation.

To accelerate cryptographic operations, the module firmware uses cryptographic engine, which is connected via AHB bus.

The values of Timer 0, “Temp. Sensor” and NDRNG is fed to DRBG function for seeding and reseeding.

Other peripherals are open to NSPE application firmware.

2.3.2.1 Trustzone Block

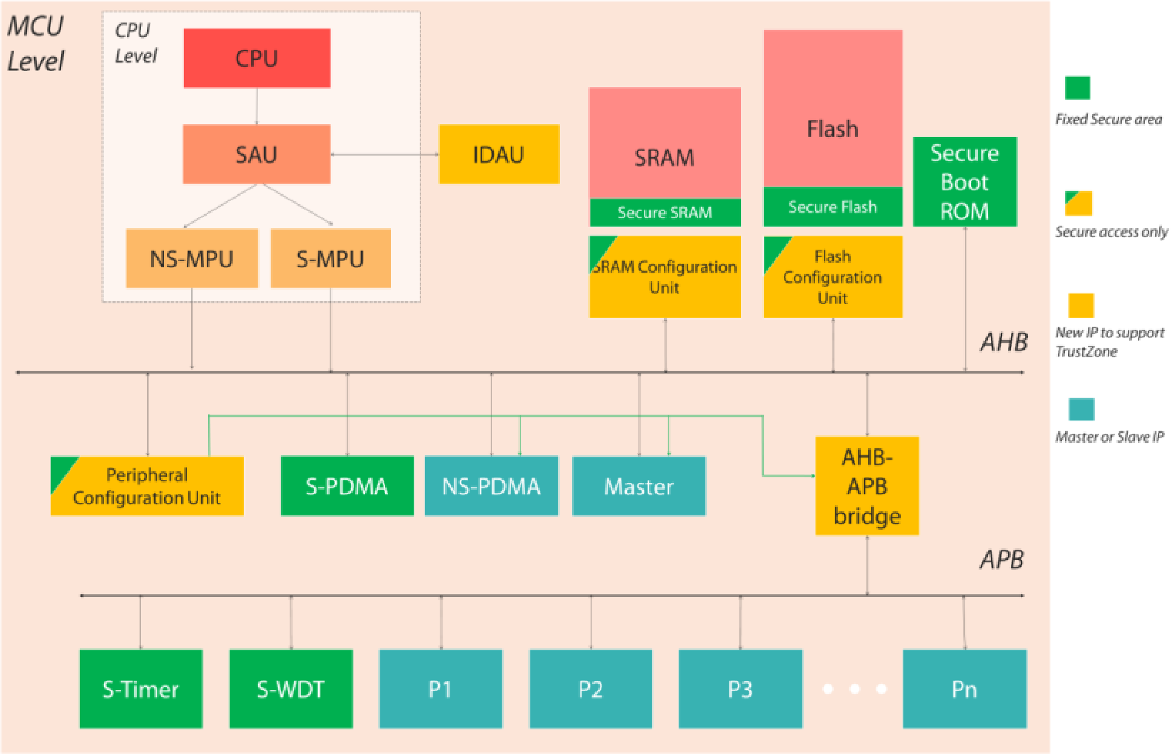

Figure 5 TrustZone Architecture Diagram

To enforce isolation, M235x SoC uses TrustZone architecture. In this architecture SRAM and flash memory is divided to SPE and NSPE regions. The green block of Figure 5 shows that this block is accessible only from SPE.

Followings are the specification of blocks related to isolation feature.

- Flash Configuration Unit
  - This unit divides SPE and NSPE region in Flash memory.
  - ABM and the module firmware are stored in SPE region.
- SRAM Configuration Unit
  - This unit divides SPE and NSPE region in SRAM.
  - Heap and stack memory for the firmware is in SPE region.
- Peripheral Configuration Unit
  - This unit enables or disables peripherals to access SPE.
- Secure Boot ROM

- Secure boot ROM supports secure boot function for boot code integrity and authenticity check.
- NuBL1 is the name of firmware doing secure boot function.

2.3.2.2 Sub-chip Security Boundary

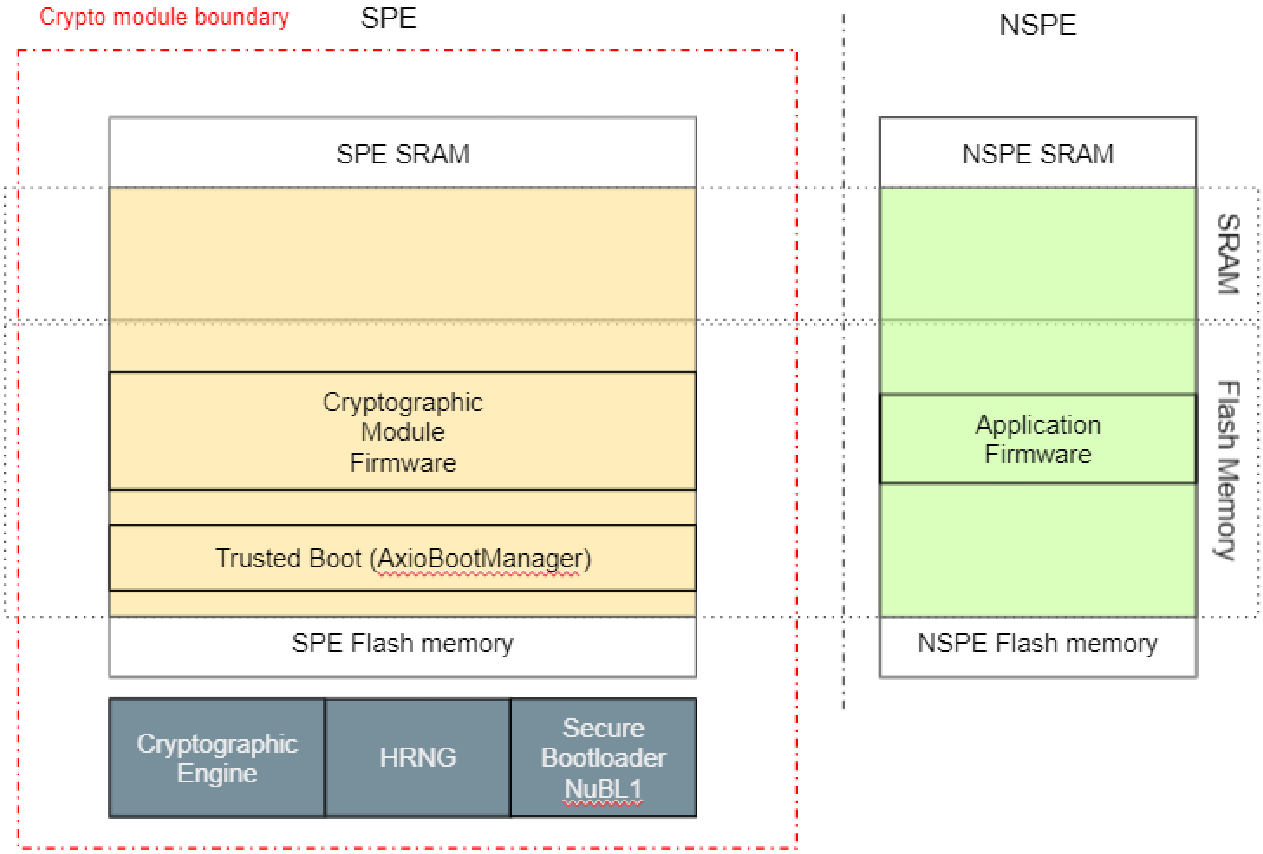

Figure 6 Security Boundary of AxioCrypto-M235x

The security boundary of AxioCrypto-M235x comprises of SPE region and cryptographic hardware. The hardware components are as next:

- Cryptographic Engine
- NDRNG
- NuBL1, Secure Bootloader in mask ROM

- SPE part of flash memory
- SPE part of SRAM

2.3.2.3 M235x SoC

The following figures show the outlet of the single-chip module. The figure below shows the external view of M235x SoC. Supported chips are listed in Table 3 below.

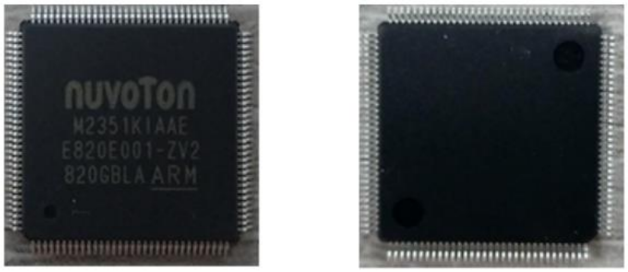

Figure 7 External view of Nuvoton M235x SoC

Supported Chip	Part Number
M2351	M2351ZIAAE
	M2351SIAAE
	M2351KIAAE
M2354	M2354LJFAE
	M2354SJFAE
	M2354KJFAE

Table 3 Part numbers supported with this module.

2.3.3 Module Components

2.3.3.1 Firmware components

The module contains firmware that resides in SPE region flash memory of the module, with key storage and future application storage functionality in the same physical flash memory. This firmware is implemented using high level language C. The customer using the module will be able to load or update an application to the NSPE region of flash memory. Figure 8 depicts the firmware components.

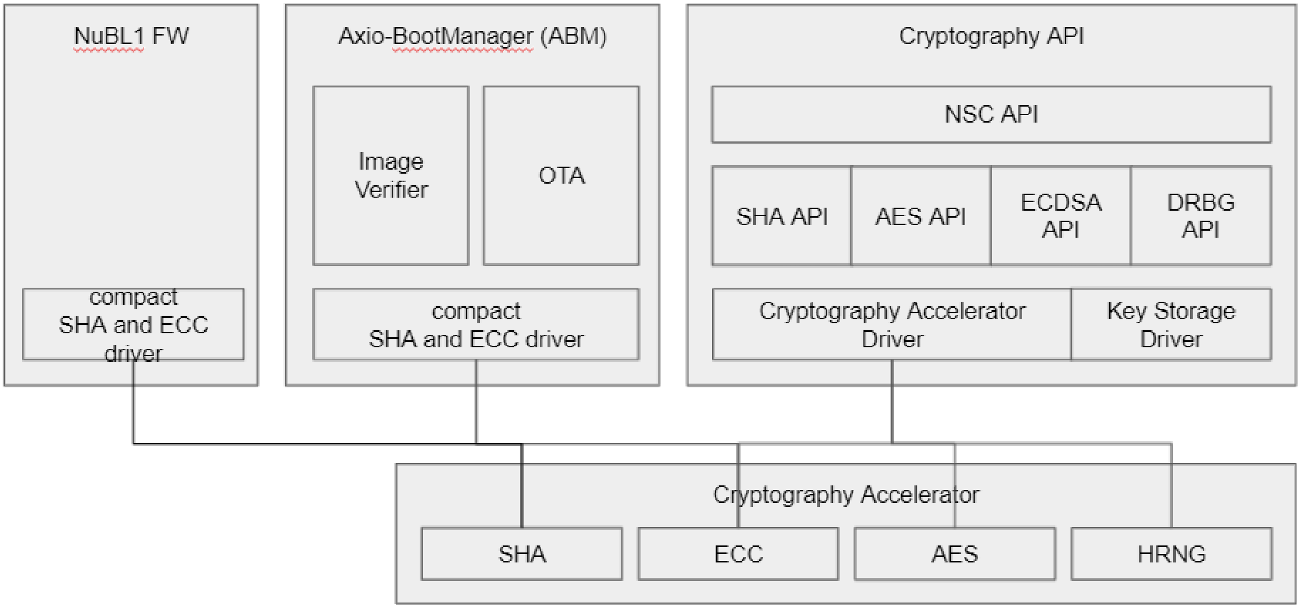

Figure 8 Firmware Block Diagram

Followings are module firmware components in SPE region.

- NuBL1 FW
  - Built-in bootloader in the M235x Mask ROM that acts as the root-of-trust.
  - NuBL1 FW supports SB(Secure Boot) verification for validating the identity and integrity of the next stage firmware.
  - The verification is based on the ECDSA.
  - NuBL1 FW uses public key hash in OTP 0~3 to identify SB root public key in ABM's meta info.
  - NuBL1 FW uses internal HIRC-48M as system clock.
- ABM
  - Axio-BootManager
  - ABM is the first mutable firmware code running after power-up.
  - NuBL1 verifies the integrity of this firmware.
  - ABM forms trust chain verifying and jumping to SPE firmware.
  - The verification is based on the ECDSA with P-256 curve and SHA2-256.
  - ABM uses SB root public key to verify SB certificate in meta info of firmware

image.
    - ABM uses SB public key in SB certificate to verify the signature of cryptography module image or application image.
- AxioCrypto-M235x module firmware
    - The module firmware provides the symmetric, asymmetric cryptography and hash service that utilizing the cryptographic accelerator.
    - The module firmware provides Deterministic Random Bit Generator (DRBG) which follows NIST SP800-90A Hash_DRBG and with seed from NDRNG.
    - The module firmware provides services for key management.
    - NSC API provides ways for NSPE application to use cryptography services in SPE.

2.3.3.2 Hardware Components

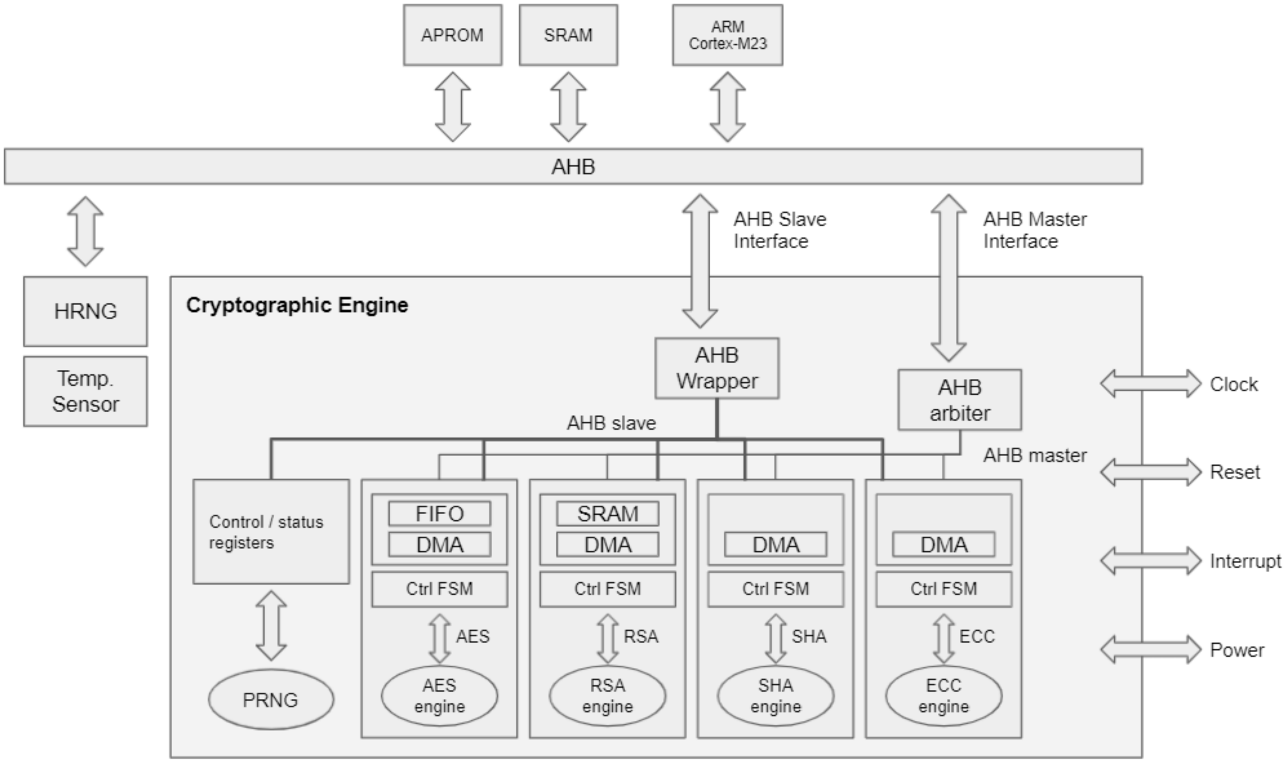

Figure 9 Cryptographic engine in M235x

Figure 9 shows the cryptographic engine which makes the module.

Followings are the specification of cryptographic engine in M235x:

- Cryptographic Accelerator
  - Elliptic Curve Cryptography (ECC)
    - ✓ This supports both prime field GF(p) and binary filed GF(2^m).
    - ✓ This supports NIST P-192, P-224, P-256, P-384, and P-521.
    - ✓ This supports NIST B-163, B-233, B-283, B-409, and B-571.
    - ✓ This supports NIST K-163, K-233, K-283, K-409, and K-571.
    - ✓ This supports point multiplication, addition and doubling operations in GF(p) and GF(2^m).
    - ✓ This supports modulus division, multiplication, addition and subtraction operations in GF(p) and GF(2^m).
  - AES 128, 192, and 256 bits
    - ✓ ECB, CBC, CFB, OFB, CTR, CBC-CS1, CBC-CS2, and CBC-CS3 mode
  - SHA-1, SHA2-224/256/384
- Random Number Generator
  - NDRNG (HRNG) for key generation
  - NDRNG collects random bits from a set of LFSR (Linear Feedback Shift Register) with guaranteed entropy.

2.4 FIPS Approved mode of operation

The module contains FIPS Approved mode of operation and a non-FIPS Approved mode of operation. The module operates in a FIPS Approved mode of operation by default, comprising all services described in section “2.5 FIPS Approved Security Functions”

For the module to operate in Non-Approved mode of operation, user must call axiocrypto_set_mode(). In axiocrypto_set_mode(), the module follows the procedures as next:

- Clear all stored CSPs.

- Change salt value used in Key Encryption Key calculation.

- Change mode of operation to given value: approved or non-approved
- reboot and initialize the module

As described above, when the mode of operation changes all CSPs are cleared and encryption key is changed for all CSPs not available after mode change.

The module does not implement bypass or maintenance modes.

The module will enter FIPS Approved mode following on a successful power up self-tests.

2.5 FIPS Approved Security Functions

The following table gives the list of FIPS Approved security functions that are provided by the module. The following notes and caveats apply:

Security Function	Details	CAVP Cert. #
AES	ECB (e/d; 128, 192, 256); CBC (e/d; 128, 192, 256); CTR (e; 128, 192, 256);	#C1603
AES-GCM	Direction: Decrypt, Encrypt IV Generation: External Key Length: 128, 192, 256 Tag Length: 32, 64, 96, 104, 112, 120, 128 IV Length: 8, 96, 1024 Payload Length: 16, 128, 256, 1000 AAD Length: 16, 128, 256, 1000	#C1603
ECDSA KeyGen	Curve: P-256 Secret Generation Mode: Testing Candidates	#C1603
ECDSA KeyVer	Curve: P-256	#C1603

Security Function	Details	CAVP Cert. #
ECDSA SigGen	Capabilities: Curve: P-256 Hash Algorithm: SHA2-256	#C1603
ECDSA SigVer	Capabilities: Curve: P-256 Hash Algorithm: SHA2-256	#C1603
Hash_DRBG	Prediction Resistance: Yes Supports Reseed Capabilities: Mode: SHA2-256 Entropy Input: 256 Nonce: 128 Personalization String Length: 0-256 Additional Input: 0-256 Returned Bits: 1024	#C1603
SHA2-256	Message Length: 8-51200 Increment 8	#C1603
HMAC-SHA2-256	MAC: 256 Key Length: 256-768 Increment 128	#A951
KAS-ECC-SSC SP800-56Ar3	(Cofactor) Ephemeral Unified Model, C(2e, 0s, ECC CDH) Domain Parameter Generation Methods: P-256 Scheme: ephemeralUnified: KAS Role: responder Note: Key establishment methodology provides 128 bits of encryption strength	#A951
KAS-KDF HKDF SP800-56Cr1 (KDA)	Two-step Concatenation KDF Fixed Info Pattern: literal[475530f22f799d1e3f3e9b2cb912e30e]||uPartyInfo||vPartyInfo Fixed Info Encoding: concatenation Derived Key Length: 256	#A951

Security Function	Details	CAVP Cert. #
	Shared Secret Length: 256-512 Increment 64 HMAC Algorithm: SHA2-256	
CKG (Crypto Key Generation)	Standard: SP800-133r2 Key Length: 256 Note: Using unmodified output from an approved DRBG	Vendor-affirmed

Note: Not all of the tested algorithms/modes are used by the module.

Table 4 FIPS Approved Security Functions

2.5.1 Non-Approved security functions but allowed in FIPS Mode

The module implements the following non-Approved but allowed algorithms in the FIPS140-2 mode of operations:

non-Approved security function	details
NDRNG	internal entropy source providing 384 bits of entropy to the DRBG

Table 5 Non-Approved security functions but allowed in FIPS mode

2.5.2 Internal IV Generator

The AES-GCM IV generation method from each of AES #C1603 is in compliance with IG A.5, scenario #2. The DRBG Cert. #C1603 is called to generate the IV inside the module and the IV length is 96 bits. The module generates new AES-GCM keys if the module loses power.

- For that purpose, the parameter of API should be set properly. Details are available in “AxioCrypto-M235x Runtime Library API Documentation”.

2.6 Non-Approved Security Functions

Functions in following table are not available in FIPS Approved mode.

Algorithm	Details	Use

ARIA	CBC (e/d; 128, 192, 256);
	CTR (e; 128, 192, 256);
		encryption

ARIA-GCM	Direction: Decrypt, Encrypt IV Generation: External Key Length: 128, 192, 256 Tag Length: 32, 64, 96, 104, 112, 120, 128 IV Length: 128 Payload Length: 0, 1024 AAD Length: 0, 1024	Authenticated encryption

Table 6 Non-Approved Security Functions

2.7 Life Cycle State and Operational State

The module life cycle and operational states follow a Finite State Model.

A module in the field is normally in the Operational life cycle state, which means it has all system keys in place and security functions enabled. The following is a summary outlining the life cycle states (LCS) and the operational states (OPS) comprising the module. Details of finite state model is in proprietary "AxioCrypto-M235x Finite State Model.docx"

Life Cycle State	Operational State	Description
Initialized crypto module	-	Bootloader ABM and crypto module firmware are written in this state associated with the independent Chip Vendor keys such as SB root public key hash, SB root public key and SB public key. In case any pre-issued keys are to be provisioned, they are done in this state.
Device Manufacturing	-	This state is associated with the Original Equipment Manufacturer (OEM), who packages the chip into a finished device.

Life Cycle State	**Operational State**	**Description**
Operational	Power-off	Power is off
	Cold Power-on	System Power on
	Self-Test	Performs FIPS power-on tests
	User	States in which authorized users obtain security services, perform cryptographic operations, or perform other approved or non-Approved functions
	Error	The FIPS error state is entered from the Self-Test state or through on of the conditional tests in case of error
RMA		The Return Merchandise Authorization state is for devices that return to the manufacturer for failure analysis

Table 7 Life Cycle and Operational States

3 Cryptographic Module Ports and Interfaces

The module supports a number of physical ports and logical interfaces, as shown in Table 8 Ports and Interfaces

below.

Type	Interfaces
Power	Power
Control Input	Clock
	Reset
	Interrupt
	AHB
	NSC API calls
Status Output	NSC API return values
Data Input	AHB
	NSC API inputs
Data Output	AHB
	NSC API outputs

Table 8 Ports and Interfaces

3.1 Hardware Physical Ports

This module has physical ports as following:

- Power: When CPU is powered on, the module starts operation and the state transfers to power on state before self-test. When powered down the module finished the operation and state transfers to power-down
  The single-chip module is provided with power via Power pin and doesn't provide power to external device.

- Reset: Reset Signal resets CPU and states.

- Clock: The clock controller in CPU generates clocks for the whole chip, including system clocks and all peripheral clocks based on external clock input.

- AHB: The AHB is used to pass data between SRAM, Flash memory group and cortex-M23 CPU.
  AHB is shared physical port among NSC API calls, NSC API return values, NSC API data inputs and NSC API data outputs. AHB is used for ARM core and Cryptographic Engine to access APROM and SRAM. In this case AHB is memory interface. The AHB operation follows memory reference in function call, which is the same as function call in software operation.

- Interrupt: peripheral devices configured in SPE have an interrupt signal to inform the Cortex-M23 Core for status change.

The block diagram is in “Figure 4 Hardware Components”

There is no cover, opening, manual control, nor physical status indicator in single-chip module.

3.2 Firmware Logical Interfaces

This module’s data input, data output, control input and status output all use NSC API calls. Data input is in input buffer delivered by NSC API and Data output is in output buffer. Control input is a specific NSC API for a job desired. Status output is the return value of NSC API.

NSC API in NSC which specify control input is called from user application firmware and transfers data input to cryptographic module in SPE. When the cryptographic module returns, NSC API delivers data output in output buffer and returns status output to user application firmware.

logical interface	data input	data output	control input	status output
place in NSC API call	input buffer	output buffer	each NSC API, flags in parameter	return value

Table 9 Logical interfaces

There is no external device for data input, data output, control input nor status output.

When the module is in Error state, data input and data output is ignored and status output contains the type of error.

When the module is in Self-test state, SPE blocks the scheduler and NSPE application cannot run until SPE releases the scheduler. In the state, there is no data output.

The module doesn’t provide any API which allows to output key components or CSPs.

At any time, only single instance of task / thread can use NSC API service. It is limited at the entrance of NSC API call. As a result, output data path is disconnected from the tasks / threads performing key generation, key entry or key zeroization because that data output path is in progress means that other API calls are all blocked from the start.

Documentation for the APIs is available in the proprietary documentation package.

4 Roles, Services and Authentication

4.1 Roles

AxioCrypto-M235x offers high-level cryptographic services which operate on CSPs and user inputs, as well as platform security services which can be used by the incorporating platform to establish a root of trust. In addition, the module requires Initialization at manufacturing time, and offers special states for recovery.

The following two roles are defined:

- User Role
  The user is defined as a set of firmware applications running in the NSPE, when the module is in Operational state. This role accesses cryptographic services, including Approved security functions, as well as platform security services.

- Crypto Officer Role
  The Crypto Officer is defined as the platform’s manufacturer. The Crypto Officer is responsible for initializing the module’s non-volatile memory and OTP, in order to make the module operational. This role has exclusive access to the services that change the module’s life cycle state. Life cycle states that are designated for the Crypto Officer Role are Chip Manufacturing, Device Manufacturing, and RMA (Return Merchandise Authorization).

The module does not define a Maintenance Role. While the module does enable the operator to enter the RMA states after the module has already been operational, those states do not provide additional access to CSPs in the module. RMA states are limited only to holders of diagnostics firmware. On update CSPs are erased because update applies to the whole area of SPE.

The module doesn’t support multiple concurrent operators. When the module is running NSPE application is blocked waiting for the result from the module.

4.2 Identification and Authentication

Role authentication is implicit. The Crypto Officer Role is defined by access to the initialization services. Specific Life Cycle states are designated for the Crypto Officer role. At manufacturing time, the module assigns the operator to the Crypto Officer role until the OTP is initialized, putting the device into the Operational Life Cycle State. At that point, the module assigns the operator to the User role. Before OTP Initialization, no firmware is recorded in internal storage and no firmware can run on the hardware.

The module remains assigned to the User Role permanently, unless the operator moves the module into special Life Cycle States intended for fault discover (RMA), which are again assigned to the Crypto Officer role.

4.3 Services

The services provided by the module to each role are specified in the table below.

Access to keys is denoted with a single letter according to the following notation:

- I – input key(s) from the user
- O – output key(s) to the user
- R – read key(s) from internal storage
- W – write key(s) to internal storage
- C – zeroized key(s) in internal SRAM
- Z – zeroize key(s) in internal storage

Approved Service (Function)	**Purpose**	**Security Functions**	**Keys and CSPs**	**User Role**	**CO Role**
AES Key input	Put AES keys into the module's internal storage		Input: User Keys (I,W) (Note: all keys indicated as user key in [SP, Table 12])	✓	

Approved Service (Function)	**Purpose**	**Security Functions**	**Keys and CSPs**	**User Role**	**CO Role**
AES	Encrypt / Decrypt data via approved security function AES	AES-128,192,256 modes: ECB, CBC, CTR, GCM	Input: User keys(R)	✓	
Hash	Calculates a message digest via approved hash function (SHS)	SHA2-256		✓	
ECDSA Key Generation	ECDSA key generation	Curves: P-256 Uses DRBG	Output: User Keys(W), ECDSA Public Key(O) Sizes: 256	✓	
ECDSA Key input	put User keys into the module's internal storage	Curves: P-256	Input: User Keys (I,W) Sizes: 256	✓	
ECDSA Public Key Validation	Public Key Validation	Curves: P-256	Input: User Keys(R)	✓	
Sign Signature	Generates a ECDSA digital signature with a previously loaded private key	Curves: P-256 Uses DRBG SHA2-256	Input: ECDSA private key(R) Sizes: 256	✓	
Verify Signature	Verifies a ECDSA digital signature with a previously loaded public key	Curves: P-256 SHA2-256	ECDSA public key (R) Sizes: 256	✓	
Change Key	Change User keys		User Keys (W) ECDSA Keys (W)	✓	
DRBG Instantiation	DRBG Context Instantiation	Creates a DRBG context using	Output: DRBG context (V, C) (W)	✓	

Approved Service (Function)	**Purpose**	**Security Functions**	**Keys and CSPs**	**User Role**	**CO Role**
		NDRNG internally for seeding			
DRBG Generation	Generate Random Vector Generate Random Vector in Range [SP800-90A]	Hash-DRBG(SHA2-256)	Input/Output: DRBG context (V, C) (R, W)	✓	
DRBG Reseeding	DRBG Reseeding with optional Additional Input [SP800-90A]		Input/Output: DRBG context (V, C) (R, W)	✓	
Erase Key storage in Flash Memory	Clear all User keys and Key Salt. Regenerate Key Salt		Zeroize: User Keys(Z), Key Encryption Key(Z), Key Salt(Z)	✓	
RMA (Return Merchandise Authorization)	Run “Erase Key storage in Flash Memory”, change Life Cycle State (LCS) to RMA		Zeroize: User Keys(Z), Key Encryption Key(Z) Output: Key Salt (Z, W), OEM public key(W) Input: SB root public key(R), SB public key (R)	✓	
Reset	Warm reset Perform self-tests	Clear keys stored in internal SRAM	zeroize: User keys(C)	✓	
Secure Boot – Firmware Certificate Verification	Certificate Chain verification Firmware image verification	Uses ECDSA, SHA2-256 to verify boot certificate and firmware contents	Input: SB root public key hash(R) SB root public key(R) SB public key(R) Sizes: 256	✓	

Approved Service (Function)	Purpose	Security Functions	Keys and CSPs	User Role	CO Role
Firmware Update	Module firmware update as an integrated part of device firmware update	Uses ECDSA, SHA2-256 to verify firmware image certificate and firmware contents	Input: SB root public key(R) SB public key(R) sizes: 256	✓	
Module firmware provisioning	Write bootloader and cryptography module firmware to SoC		Output: SB root public key hash(W)		✓
OEM Asset Provisioning	Write OEM firmware image to internal storage	ECDSA verify	Input: SB root public key hash(R) SB root public key(R) SB public(R) OEM public key(R)		✓
Provide information	Indicates the FIPS status: Approved, or Error State (include error code)			✓	
Set Mode	Sets operation mode: approved or non-approved.		Zeroize: User Keys(Z), Key Salt(Z)	✓	
HMAC	Message authentication	HMAC SHA2-256	Input: User Keys(I)	✓	
ECDH Key Exchange	Key Agreement [SP800-56A]	EC Diffie-Hellman Curves: P-256	Input: User Keys(I) Output: User Keys(O) Size: 256	✓	

Table 10 Approved Services

Non-Approved Service (Function)	Purpose	Security Functions	Keys and CSPs	User Role	CO Role
ARIA Key input Non-Approved	Put ARIA keys into the module’s internal storage		Input: User Keys (I,W)	✓	
ARIA Non-Approved	Encrypt / Decrypt data via non-approved security function ARIA	ARIA-128,192,256 modes: ECB, CBC, CTR, GCM	Input: User keys(R)	✓	

Table 11 Non-Approved Services

5 Physical Security

The SoC containing the sub-chip module and the memory chips is a production-grade component with standard passivation protection. The final device which contains the SoC and memory chips are entirely contained within a hard-plastic production-grade enclosure. The module is designed to meet FIPS140-2 Level 1.

6 Operational Environment

The module hardware which is integrated in the SoC and the module firmware which resides in internal flash memory are all contained in a commercial mobile device manufactured by the device OEM. The module uses integrity techniques to enforce the non-modification of its firmware. The module firmware is disconnected from all the I/O peripherals in the SoC. Therefore, the operational environment is considered non-modifiable.

The FIPS140-2 Area 6 Operational Environment requirements do not apply to the module in this validation because the module does not contain a modifiable operational environment.

7 Cryptographic Key Management

7.1 Critical Security Parameters and Public Keys

The module works with two types of keys: system keys and user keys. The cryptographic module includes system keys for administration purposes. System keys are limited by hardware to use in a few specific operations. User keys are generated inside the module or explicitly inserted to the module. In both cases user keys are stored inside SPE region and user can access these keys indirectly by the index of slot in which they are stored. Once User keys are inserted to the module, the contents of user keys are not exposed outside the module. The following table provides a list and description of all CSPs and public keys managed by the module. Keys are marked in columns of “System Key” or “User Key”

CSP/ **Public Key**	**Type**	**Generate/** **Input**	**Output**	**Storage**	**Use**	**System Key**	**User Key**
SP800-90A DRBG Seed	256-bit random number	Input from the NDRNG of the module	None	RAM	Used to seed the SP800-90A DRBG.		✓
SP800-90A Nonce	80-bit increasing counter	Input from 16MHz Timer	None	RAM	Used to provide nonce on SP800-90A DRBG Initialization		✓
SP800-90A personalization string	96-bit number	Input from CPU Unique ID	None	CPU Register	Used to provide personalization string on SP800-90A DRBG Initialization		✓
SP800-90A additional Input	12-bit number	Input from Temperature Sensor in SoC	None	CPU Register	Used to provide additional input on SP800-90A DRBG Initialization		✓

CSP/ Public Key	Type	Generate/ Input	Output	Storage	Use	System Key	User Key
SP800-90A internal state V	440-bit number	Calculated on DRBG Init	None	RAM	SP800-90A DRBG Internal state		✓
SP800-90A internal state C	440-bit number	Calculated on DRBG Init	None	RAM	SP800-90A DRBG Internal state		✓
SB root public key hash	SHA2-256 hash	Load on boot time	None	OTP	CPU checks if SB root public key is registered.	✓	
SB root public key	256-bit ECDSA public key	Load on boot time	None	Flash Memory meta info of ABM	used to verify the integrity of ABM	✓	
SB public key	256-bit ECDSA public key	Load from meta info of the module	None	Flash Memory meta info of module firmware. Included in SB Certificate.	used to verify the integrity of AxioCrypto module firmware image	✓	
OEM public key	256-bit ECDSA public key	Load from meta info of OEM firmware	None	Flash Memory meta info of module firmware. Included in OEM Certificate.	used to verify the integrity of OEM firmware image	✓	
Key Salt	256-bit number	Obtained directly from DRBG on key storage initialization / Load on key storage access	None	Flash Memory SPE	Used to diversify Key Encryption Key with fixed CPU ID.	✓	
Key Encryption key	256-bit AES keys	Derived from XORing CPU ID and Key Salt	None	RAM	Used to encrypt and decrypt User keys and ECDSA keys	✓	

CSP/ Public Key	Type	Generate/ Input	Output	Storage	Use	System Key	User Key
ECDSA private key	256-bit ECDSA key	Generated using SP800-90A DRBG / Input from outside of the module / Load from flash memory in encrypted form	None	Flash Memory SPE	Used to generate signature during ECDSA generation.		✓
ECDSA public key	256-bit ECDSA key	Generated inside module / Input from outside of the module / Load from flash memory in encrypted form	axiocrypto_asym_getky()	Flash Memory SPE	Used to verify signature during ECDSA verification.		✓
AES key	256-bit AES keys	Input from outside of the module / Load from flash memory in encrypted form	None	Flash Memory SPE	Encryption / Decryption		✓
ECDH private key	256-bit ECDH key	Generated using SP800-90A DRBG / Input from outside of the module / Load from flash memory in encrypted form	None	Flash Memory SPE	Used to generate share key during Key Agreement		✓
ECDH public key	256-bit ECDH key	Generated inside module / Input from outside of the module / Load from flash memory in encrypted form	axiocrypto_ecdh_getkey()	Flash Memory SPE	Key Token used in Key Agreement		✓

CSP/ Public Key	Type	Generate/ Input	Output	Storage	Use	System Key	User Key
ECDH shared secret	256-bit AES key	Generated inside module	None	RAM	Shared secret computation		✓
Session key derive from HKDF	256-bit AES key	Generated inside module	None	RAM	Key derivation		✓
HMAC key	512-bit HMAC key	Load from outside of the module / Load from flash memory in encrypted form	None	Flash Memory SPE	Message authentication generation, message authentication verification		✓

Table 12 Critical Security Parameters and Public Keys

7.2 Key Generation and Diversification

The module uses a DRBG compliant with the NIST SP800-90A standard [SP800-90A], seeded with 384 bits from a NDRNG. See “2.3.3.1 Firmware components” and “2.3.3.2 Hardware Components” for details on the NDRNG and DRBG implementations.

Asymmetric key generation services offered by the module are defined by the [FIPS186-4] standard, using the DRBG service for random inputs, and are detailed in Section 4.3. No dedicated service is offered for symmetric key generation; the user is instructed to directly use the output of the DRBG service.

Internally, the module uses the DRBG to provide random inputs to ECDSA, ECDH, and HMAC cryptographic services that use randomness. There is no intermediate information output during or upon completion of key generation process.

Based on CPU ID, Key Encryption Key is generated along with Key Salt. Key Salt is obtained using unmodified output from the approved DRBG and stored in flash memory. It remains unchanged until the flash memory area is re-initialized. The flash memory area is in SPE. Key Encryption Key is generated using the formula: (Key Salt $\oplus$ CPU ID).

7.3 Key Establishment

The module offers key establishment service detailed in 4.3. The key agreement is based on Ephemeral Unified Model scheme [SP800-56A, 6.1.2.2]. Session key derivation is based on HKDF [SP800-56C].

7.4 Key Entry and Output

The module supports electronic key entry methods only. The module does not support manual key import or export methods. User keys are input and output electronically in plaintext, as parameters to the module's NSC APIs. AES keys, ECDSA private keys and ECDH private keys can only be input with destination slot's index. ECDSA public key and ECDH public key can be input or generated from corresponding private key and output.

The module does not receive seeds from the outside. The DRBG is only seeded internally from the NDRNG; the DRBG does support a service to receive additional data from the user, according to the definitions in NIST SP800-90A.

7.5 Key Storage

The seed values in DRBG is transient, not stored in the module.

The module stores up to 20 entries including ECDSA, EC Diffie-Hellman, HMAC and AES keys. These keys and CSPs are stored in dedicated space of Flash Memory. Before CSPs stored into memory they are encrypted using approved AES-GCM. The keys are derived from device specific hardware id and 32 bytes salt value. Salt value is generated from Random() when the storage is initialized. These values are restricted from reading outside SPE.

7.6 Key Zeroization

The user keys in volatile memory can be zeroized at any time by putting the module through a power-on reset. The reset clears hardware registers and SRAM contents. The module specifically erases the portions of SRAM which may hold keys, CSPs and intermediate values: the SRAM and the work buffer described in Section 7.1 are cleared on every operation, and following reset.

For user keys in persistent memory, user can trigger the key zeroization via following services:

a) Erase Key storage in Flash Memory

b) Firmware Update

c) Set Mode (approved mode in non-approved mode, or non-approved mode in approved mode)

For system keys, user can zeroize all the keys but ‘SB root pubic key hash’ using a PC tool to send command to device. When PC tool sends command to device, the device firmware interprets it and enables RMA state.

‘SB root pubic key hash’ is in OTP and not zeroized because OTP is not modifiable after lock.

7.7 RNG Seed Values

During power up initialization, the module uses NDRNG to compute new DRBG Seed values. Any old seed values (which were randomized) are then overwritten with the new computed values. These seed values temporarily exist in volatile memory and are zeroized by power cycling the module. These values are not accessible to any user. The NDRNG internal entropy source providing 384 bits of entropy to the SP 800-90A DRBG.

8 Electromagnetic Interference/Compatibility (EMI/EMC)

According to 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Chapter 15.101, “No authorization is required for a peripheral device or a subassembly that is sold to an equipment manufacturer for further fabrication; that manufacturer is responsible for obtaining the necessary authorization prior to further marketing to a vendor or to a user.” As this product is a subassembly sold for further fabrication, it is exempt from part 15 of the FCC rules.

9 Self-Tests

According to FIPS140-2 requirements, the module performs a number of self-tests on power-up, conditionally on selected events.

9.1 Power-up Tests

The following tests are performed on module initialization, before the operator can request any of the module's cryptographic services. The tests are performed automatically, without needing any action from an operator.

A failure of any of these tests will cause the module to enter the Error state, which is indicated by the status output. On success, the module will enter the Approved state and will change its status output accordingly. In the Error state, the module no longer responds to further commands, and output any data. Users can retry self-test by rebooting the device.

The module does not support a dedicated service or API to invoke those tests on demand; the operator can initiate a power-on reset, to have the module perform the tests.

9.1.1 Cryptography Test

A cryptographic algorithm test using a known answer are conducted for all cryptographic functions of each Approved cryptographic algorithm implemented by a cryptographic module. Known input data and answers are stored in Flash Memory. Tests include following:

- AES encryption and decryption operations, in ECB, CBC, CTR and GCM with 128, 192 and 256-bit key sizes.
- Hash generation for SHS, with SHA2-256 tested.
- Message authentication using HMAC with SHA2-256
- ECDSA signature generation and verification using P-256 curve, with the SHA2-256 message digests.
- KAS-ECC-SSC: primitive “Z” Computation KAT

- KAS-KDF HKDF: HKDF KAT
- DRBG instantiate, Reseed, Additional Data, random vector generation.

KATs function by encrypting/decrypting, hashing or signing a string for which the calculated output is known and stored within the cryptographic module. An encryption, hashing or signature test passes when the calculated output matches the expected value. The test fails when the calculated output does not match the expected value. Because there is single implementation for each algorithm, the calculated output is only compared to stored value in internal flash memory.

KATs for DRBG function by seeding the DRBG with known values and checking that the output matches the pre-calculated value stored within the cryptographic module.

If any test fails, the module will abort test procedures and return a value indicating Error state.

9.1.2 Firmware Integrity Test

The module performs firmware integrity tests as part of the boot sequence. Instead of EDC, approved digital signature algorithm using ECDSA P-256 with SHA2-256 hash function is used.

When the firmware is updated, the firmware image is externally loaded. In that case, the integrity of the image is checked using ECDSA P-256 with SHA2-256 hash function.

The firmware integrity test operates as in the following figure:

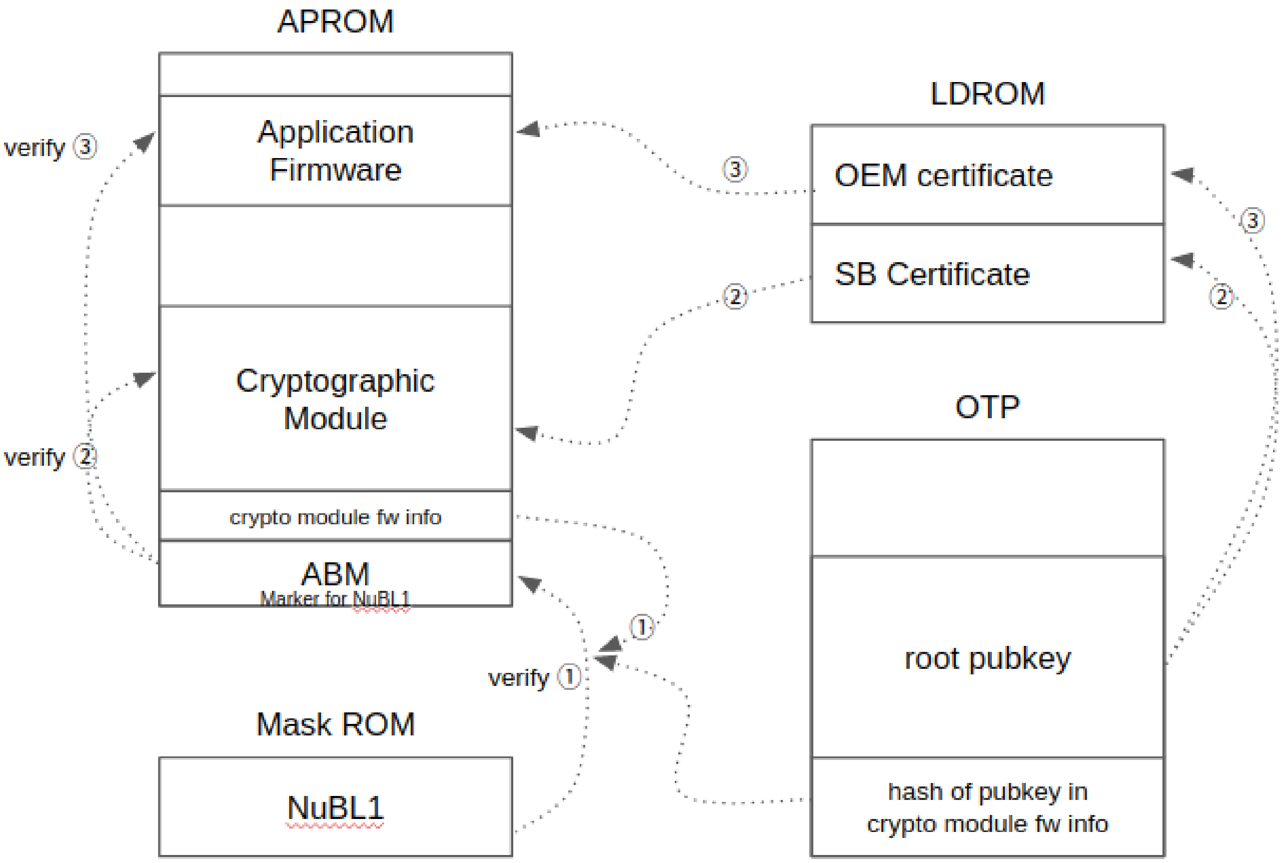

Figure 10 Firmware Integrity Test Procedure

There are a few storage peripherals in M235x SoC:

- APROM: internal flash memory in M235x SoC, for code and data
- Mask rom: storage for NuBL1, secure boot loader. The contents are immutable after fusing at manufacturing.
- OTP: One-Time Programmable area. Applications can write in OTP and lock it. After locking, the contents are preserved.
- LDROM: 4KB sized space for data or boot-time code. In AxioCrypto-M235x, LDROM is used as the storage for certificates.

NuBL1, the first bootloader after power-up is in Mask ROM. When power is provided and M235x SoC starts running, NuBL1 starts running before any other firmware. NuBL1 uses

OTP area for storage of public key for next-level firmware verification. NuBL1 knows where the next level firmware is – it's at the lowest area of internal flash memory. And, at the offset of 0x10, there is "Marker for NuBL1", which contains the address of next level "fw info". In fw info, there are size of the firmware, public key for verifying it, and the signature.

When NuBL1 finds the public key for verification, it calculates the SHA2-256 hash and compares it to that in OTP. If the match, NuBL1 gives control to next-level firmware – now ABM. The arrows with ① show this process.

ABM uses the hierarchical structure of certificates to verify the integrity of cryptographic module and application firmware from different makers. ABM stores the root public key in OTP, next to "hash of public key" used by NuBL1. With the root public key, it verifies the integrity of certificates in LDROM. With each of certificates in LDROM, it verifies the cryptographic module and application firmware. Then, it jumps and gives control to cryptographic module. ② shows the verification process for cryptographic module and ③ shows it for application firmware.

The cryptographic module runs necessary operations and jumps to application firmware. When application firmware requests the cryptographic services from cryptographic module, it provides the service.

The developer of application firmware can install his/her own certificate if only the certificate was signed with private key which pair with "root public key" in OTP.

The certificate used here is compatible to X.509 public key certificate.

9.1.3 Critical Function Test

No other critical function test is performed on power up

9.2 Conditional Self-Tests

The pairwise consistency test is run whenever the module generates an asymmetric key pair.

Continuous RNG tests are implemented in the module's two random number generators —

the NDRNG and the DRBG. In both, each block of output generated is compared for equality with the previous block (exempting the very first block generated). The NDRNG uses a block size of 16 bits. The DRBG uses a block size of 256 bits. A failure of each Continuous RNG Test will put the entire module into the Error State.

10 Design Assurance

10.1 Configuration Management

Git is used for software source control and configuration management. It supports full history and version tracking.

10.2 Delivery and Operation

For the bootloader and crypto module firmware, they are delivered in binary format, embedded in the M235x SoC. All of the images are signed and protected by secure boot. The cryptographic model is built-in and can’t be disabled by any feature option.

10.3 Guidance Documents

The guidance for the operator is listed below:

- The operator is instructed to follow the additional caveats on approved functions.

The documents are provided in a link for download.

10.4 Proprietary Document

The following proprietary documents are available to the customers for this module:

- AxioCrypto Runtime Library API Documentation
- AxioCrypto Software Integration Guidelines
- Nuvoton-M235x Technical Reference Manual

The documents are provided in a link for download.

11 Mitigation of Other Attacks

The cryptographic module is not designed to mitigate specific attacks.

12 References

[FIPS140-2] Security Requirements for Cryptographic modules

[FIPS186-4] Digital Signature Standard (DSS)

[SP800-90A] Recommendation for Random Number Generation Using Deterministic Random Bit Generators

[IG] Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program

[SP800-56A] Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography

[SP800-56C] Recommendation for Key-Derivation Methods in Key-Establishment Schemes

[SP800-133] Recommendation for Cryptographic Key Generation

13 Acronyms

AES	Advanced Encryption Standard
AHB	AMBA High-Performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APROM	Application ROM
CBC	Cipher Block Chaining
CDH	Cofactor Diffie-Hellman
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
DES	Data Encryption Standard
DPA	Differential Power Analysis
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
HMAC	Keyed-Hash Message Authentication Code
HIRC	High-speed Internal RC oscillator
HKDF	HMAC-based Extract-and-Expand Key Derivation Function
HRNG	Hardware Random Number Generator
KAT	Known Answer Test
KAS	Key Agreement Scheme
KDA	Key Derivation Algorithm
LDROM	Loader ROM
LFSR	Linear Feedback Shift Register
LIRC	Low-speed Internal RC oscillator
NDRNG	Non-Deterministic Random Number Generator
NSC	Non-Secure Callable

NSPE	Non-Secure Processing Environment
OTP	One Time Programmable
RAM	Random Access Memory
RMA	Return Merchandise Authorization
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest Shamir and Adleman Public Key Algorithm
SB	Secure Boot
SG	Secure Gateway (Cortex-M23 Instruction)
SHA	Secure Hash Algorithm
SOC	System on Chip
SPE	Secure Processing Environment
SSC	Shared Secret Computation