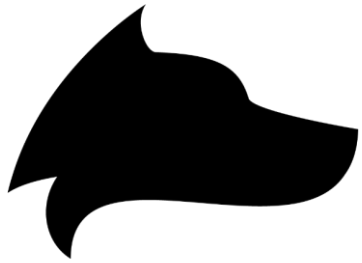


Privoro LLC

Privoro SafeCase Security Module

FIPS 140-3 Non-Proprietary Security Policy

Document Version RC3

July 8, 2026

Prepared for:

Privoro LLC
3100 W. Ray Road, Suite 201
Chandler, AZ 85226
privoro.com
+1 844.774.8676

Prepared by:

KeyPair Consulting Inc.
987 Osos Street
San Luis Obispo, CA 93401
keypair.us
+1 805.316.5024

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document defines the Security Policy for the Privoro SafeCase Security Module by Privoro LLC (Privoro), hereafter denoted the Module. The Module is a firmware-hybrid module with a single-chip embodiment, updated only by a complete image replacement by Privoro in the SafeCase environment.

The Module is validated to FIPS 140-3 overall Level 2 requirements with security levels as specified in Section 1.2.

1.2 Security Levels

Section	Title	Security Level
1	General	2
2	Cryptographic module specification	2
3	Cryptographic module interfaces	2
4	Roles, services, and authentication	2
5	Software/Firmware security	2
6	Operational environment	N/A
7	Physical security	3
8	Non-invasive security	N/A
9	Sensitive security parameter management	2
10	Self-tests	2
11	Life-cycle assurance	2
12	Mitigation of other attacks	2
	Overall Level	2

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Module provides cryptographic services for use in SafeCase mobile device management systems.

Module Type: Firmware-hybrid

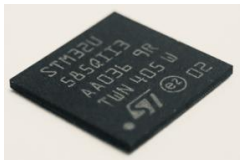
Module Embodiment: Single Chip

Cryptographic Boundary:

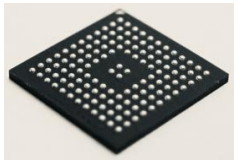
The Module's cryptographic boundary representing the logical functionality (outlined in red) is shown in Figure 2.

Tested Operational Environment's Physical Perimeter (TOEPP):

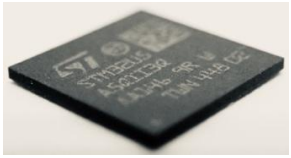
The TOEPP consists of the surfaces, edges and solder connections of the integrated circuit package shown in Figure 1. The hardware is the STM32U5 System on Chip (SoC) in two variants with the feature differences listed in the Tested Module Identification – Hybrid Disjoint Hardware Table.



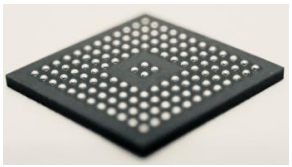
STM32U5 (85) Top



STM32U5 (85) Bottom



STM32U5 (A5) Top



STM32U5 (A5) Bottom

Figure 1: Module Physical Perimeter

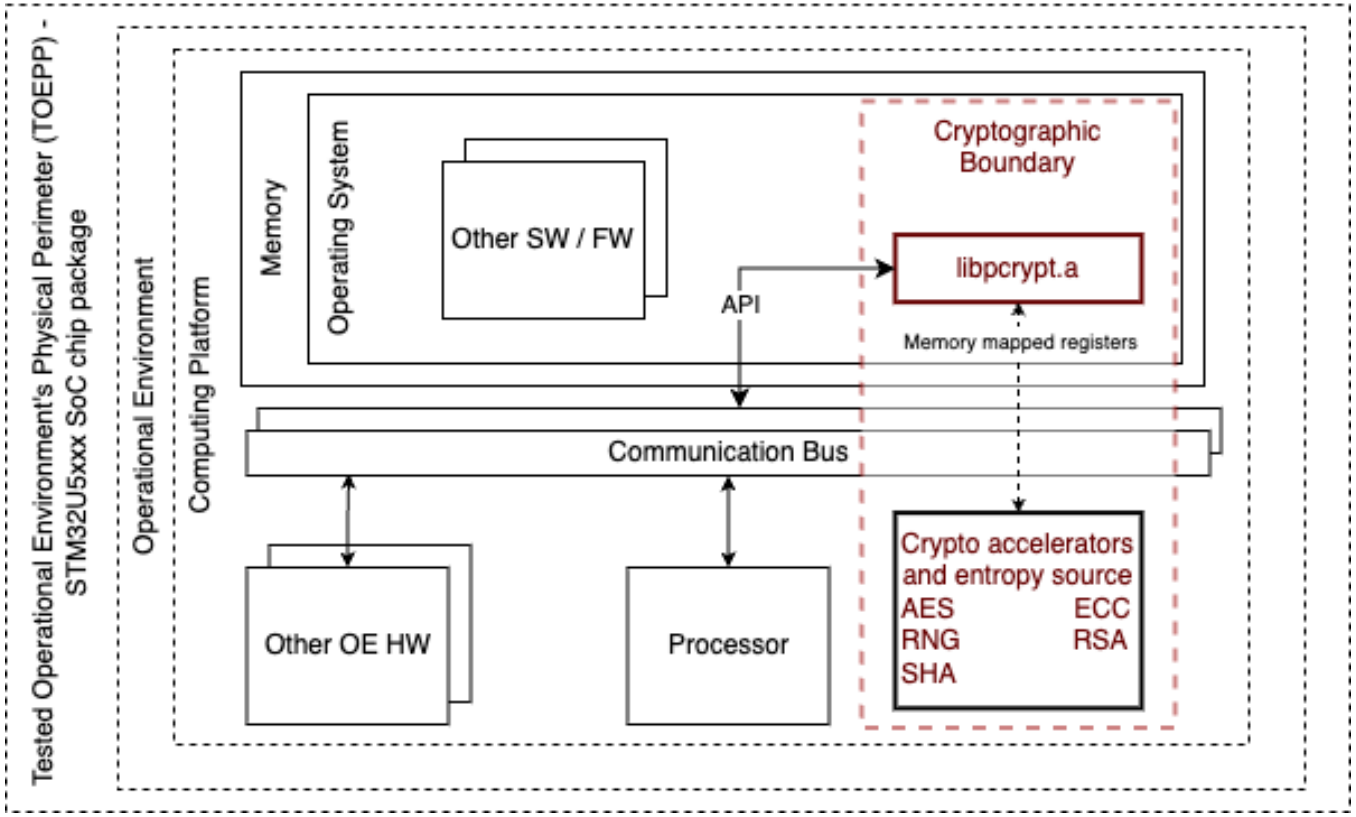


Figure 2: Block Diagram

The Module consists of two components which are disjoint relative to one another; hereafter, the qualifying term 'disjoint' is correctly associated with each component but omitted for the sake of brevity. The components are:

- libpcrypt.a - the firmware component, as shown in Figure 2.
- STM32U5_CHW: STM32U5 SoC cryptographic hardware elements providing AES, ECC, RNG, RSA and SHA functions, as shown in Figure 2.

STM32U5_CHW refers to the set of hardware accelerators utilized by the module; FIPS 140-3 draft IG 1.B describes this scenario as "…an agglomeration of hardware parts/circuits forming a hardware component boundary."

All interfaces between libpcrypt.a and the hardware accelerators are memory mapped registers which communicate over a bus governed by a controller that enforces isolated communications between the STM32M5_CHW register set and libpcrypt.a, such that no external component can access information on the passing between the components over the bus.

For all cryptographic functionality with the exception of SHA2-384, the libpcrypt.a component interacts with the associated accelerator register set to achieve the cryptographic services implemented by the Module. The libpcrypt.a API provides all entry points for all module services; in turn, libpcrypt.a interacts through cryptographic library and hardware abstraction layers, which translate the API into the memory mapped register values, invoke the hardware functions through control register writes, and obtain results when the operations are done. For example, the AES-HW provides the capability to perform the cryptographic primitives but requires libpcrypt.a functions to fulfill the complete algorithm inclusive of initialization and error checking. The SHA2-384 functionality is implemented fully in libpcrypt.a firmware.

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

Package or File Name	Software/ Firmware Version	Features	Integrity Test
libpcrypt.a	1.2.0	N/A	ECDSA signature verification (Cert. #A7216) over all firmware in the Module.

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
STM32U585	STM32U585	N/A	ARM Cortex-M33	2 MB Flash memory, 786 KB SRAM
STM32U5A5	STM32U5A5	N/A	ARM Cortex-M33	4 MB Flash memory, 2.5 MB SRAM

Table 3: Tested Module Identification – Hybrid Disjoint Hardware

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Zephyr	STM32U5	ARM Cortex-M33	No	N/A	1.2.0

Table 4: Tested Operational Environments - Software, Firmware, Hybrid

In the Hardware Platform column, "STM32U5" refers to both STM32U585 and STM32U5A5 variants of the STM32U5 hardware platform; this usage is consistent with the organization of STM32U5 design and reference manual documentation.

2.3 Excluded Components

N/A for this Module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Nominal	Approved mode of operation	Approved	Status word

Table 5: Modes List and Description

The Module supports only an Approved mode of operation by default. See Section 11.1 for installation and initialization instructions.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A7216	Key Length - 256	SP 800-38C
AES-ECB	A7216	Direction - Encrypt Key Length - 256	SP 800-38A
ECDSA KeyGen (FIPS186-5)	A7216	Curve - P-384 Secret Generation Mode - extra bits	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7216	Curve - P-384 Hash Algorithm - SHA2-384	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A7216	Curve - P-384 Hash Algorithm - SHA2-384	FIPS 186-5
Hash DRBG	A7216	Prediction Resistance - No Mode - SHA2-256	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-384	A7216	Key Length - Key Length: 192	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A7216	Domain Parameter Generation Methods - P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA OneStep SP800-56Cr2	A7216	Derived Key Length - 384 Shared Secret Length - Shared Secret Length: 384	SP 800-56C Rev. 2
RSA KeyGen (FIPS186-5)	A7216	Key Generation Mode - probable Modulo - 2048 Primality Tests - 2pow100 Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A7216	Modulo - 2048 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A7216	Modulo - 2048 Signature Type - pkcs1v1.5	FIPS 186-5
SHA2-256	A7216	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A7216	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 6: Approved Algorithms

All algorithms are implemented using a combination of libcrypt.a and STM32U5_CHW – see Section 4.3 for additional detail.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type: Asymmetric	N/A	NIST, SP 800-133 Rev. 2 Section 4, 5.1 and 5.2

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AEAD Cipher	BC-Auth	AES-CCM Authenticated cipher		AES-CCM: (A7216) AES-ECB: (A7216)
ECC key generation	AsymKeyPair-KeyGen CKG	ECC key generation		ECDSA KeyGen (FIPS186-5): (A7216) CKG: () Key Type: Asymmetric
ECDSA	DigSig-SigGen DigSig-SigVer	ECDSA Sign, Verify		ECDSA SigGen (FIPS186-5): (A7216) ECDSA SigVer (FIPS186-5): (A7216)
HMAC	MAC	Keyed Message Authentication		HMAC-SHA2-384: (A7216)
Key agreement	KAS-SSC	ECC Shared secret calculation		KAS-ECC-SSC Sp800-56Ar3: (A7216)
Key derivation	CKG KAS-56CKDF	SP800-56Cr2 OneStep KDF		KDA OneStep SP800-56Cr2: (A7216) CKG: () Key Type: Asymmetric
Message Digest	SHA	SHA hash algorithms		SHA2-256: (A7216) SHA2-384: (A7216)
Random	DRBG	Random bit generation		Hash DRBG: (A7216)
RSA key generation	AsymKeyPair-KeyGen CKG	RSA key generation		RSA KeyGen (FIPS186-5): (A7216) CKG: () Key Type: Asymmetric
RSA	DigSig-SigGen DigSig-SigVer	RSA Sign, Verify		RSA SigGen (FIPS186-5): (A7216) RSA SigVer (FIPS186-5): (A7216)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

Key Agreement:

The Module implements the following Approved key agreement method which has been CAVP tested and validated:

- KAS per SP 800-56A Rev. 3 using KAS-ECC-SSC with SP 800-56C Rev. 2 KDA One-Step KDF (FIPS 140-3 IG D.F Scenario 2, path 1).

No SSPs are established into the Module via the KAS functionality; rather, the Module provides the KAS method cited above via the *pcrypt_key_exchange* service, to be used by the calling application. Current KAS Guidance requires the following statement for this scenario:

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer some or all of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS.

The Module obtains the FIPS 140-3 IG D.F required key agreement assurances:

- SP 800-56A Rev. 3 in accordance with Section 5.6.2.

RSA:

The Module complies with FIPS 140-3 IG C.F as follows:

- RSA Key Generation, Signature Generation and Signature Verification have been tested and validated with all implemented modulus lengths for which CAVP testing is available: k = 2048.
- The Module does not support any untestable modulus lengths for RSA Key Generation, Signature Generation or Signature Verification.

2.8 RBG and Entropy

Cert Number	Vendor Name
E11	STMicroelectronics
E135	STMicroelectronics

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
STM32U5x TRNG	Physical	STMU585	128	Full	AES-CMAC-A1729
STM32U59x/5Ax TRNG	Physical	STMU5A5	128	Full	AES-CMAC-A5412

Table 10: Entropy Sources

The Module generates entropy within the Module's physical perimeter per FIPS 140-3 IG 9.3.A option 1(b) using the specified SP 800-90B compliant entropy source present on the SoC component.

Per SP 800-90A Rev. 1 Table 2, the SHA2-256 Hash DRBG requires 256 bits of entropy (equivalent to security strength) within the RBG_Seed value.

As input to the SP 800-90A Rev. 1 Hash_df, the Module collects 800 bits of data from the on-chip entropy source to use as entropy and nonce input. The on-chip entropy source is ESV listed, with entropy per bit assessed as Full Entropy. Hence, the 800 bits is in excess of the 384 bits of entropy required for the entropy and nonce input.

2.9 Key Generation

The Module:

- Produces random values in accordance with SP 800-133 Rev. 2 Section 4, in that the DRBG output is provided directly as the random output.
- Does not provide any service beyond random value generation for symmetric key generation. SSPs used with symmetric key algorithms are provided by the calling process.
- Produces asymmetric keys in accordance with SP 800-133 Rev. 2 Section 5, in that all asymmetric keys generated by the Module (the *pcrypt_ecc_gen_key* and *pcrypt_rsa_gen_key* services) provide the output of the approved key generation algorithm with no post-processing or manipulation of the generated key pairs. As noted in the first item, random values used in the asymmetric key generation algorithms are direct outputs of the DRBG. Keys produced by the Module use an internal Hash DRBG (with SHA2-256) for which the security strength is 256 bits.
- Supports symmetric key derivation in accordance with SP 800-133 Rev. 2 Section 6.2, using the approved and CAVP listed KDF algorithms.

2.10 Key Establishment

The Module implements a key agreement method compliant with FIPS 140-3 IG D.F, with strengths as shown in the SSP Table. As required by FIPS 140-3 IG D.F Additional Note 12, the Module's KAS primitives support Diffie-Hellman methods.

2.11 Industry Protocols

N/A for this Module.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Control Input Data Input	API input: stack frame including non-sensitive parameters
N/A	Data Output Status Output	API output: output parameters and return value resulting from call execution
AES Hardware accelerator register set	None	See below table for detail: AES-HW-CI, AES-HW-DI, AES-HW-DO, AES-HW-SO
Hash Hardware accelerator register set	None	See below table for detail: HASH-HW-CI, HASH-HW-DI, HASH-HW-DO, HASH-HW-SO
PKA Hardware accelerator register set	None	See below table for detail: PKA-HW-CI, PKA-HW-DI, PKA-HW-DO, PKA-HW-SO
TRNG Hardware accelerator register set	None	See detail below table, labeled TRNG-CI, TRNG-DI, TRGN-DO, TRNG-SO

Table 11: Ports and Interfaces

The Module's physical ports are outside the cryptographic boundary. The Control Output interface is not applicable, as the Module does not control other components. The interfaces between libpcrypt.a and the STM32U5_CHW set of devices are memory mapped registers with the following content:

AES-HW-CI: Parameters that govern AES accelerator usage, including controls that govern mode.

AES-HW-DI: Plaintext data to be encrypted; ciphertext data to be decrypted.

AES-DO: Decrypted (plaintext) data; encrypted (ciphertext) data.

AES-SO: AES accelerator status; includes busy and computation completion status, and confirmation of expected key size (per AS03.15).

HASH-HW-CI: Parameters that govern Hash accelerator usage.

HASH-HW-DI: Data to be hashed.

HASH-DO: Hash output.

HASH-SO: Hash accelerator status information; includes busy and computation completion status and counts of expected words and received words (per AS03.15).

PKA-HW-CI: Parameters that govern PKA accelerator usage. The PKA accelerator provides ECC and IFC (RSA) acceleration functions.

PKA-HW-DI: Message to be signed or verified.

PKA-DO: Signed message or message verification output.

PKA-SO: PKA accelerator health and status information; includes busy and computation completion status, and confirmation of expected initialization and operation (per AS03.15).

TRNG-HW-CI: Parameters that govern TRNG accelerator usage.

TRNG-HW-DI: None.

TRNG-DO: Entropy input use by the *pcrypt_rng_init* service (RBG_EI).

TRNG-SO: Health (health test errors indicated as a composite *Seed Error Current Status* field) and status information, e.g. Data ready.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
CO_auth	CO role authentication	Authentication of a 256-bit memorized secret, in accordance with [SP800-140E] and [SP800-63B];Authentication attempts require at least 50 µs	2^256 = 1.16E+77	2^256/(60*1000*20) = 9.65E+70 (per 1 minute)

Table 12: Authentication Methods

4.2 Roles

Name	Type	Operator Type	Authentication Methods
CO	Role	CO	CO_auth

Table 13: Roles

The Module supports only the Cryptographic Officer (CO) role. It does not support multiple concurrent operators, a maintenance role or bypass capability.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
pcrypt_aesccm_decrypt	Authenticated decrypt	Status word	API-Input	API-Output	AEAD Cipher	CO - SC_EDK: W,E,Z
pcrypt_aesccm_encrypt	Authenticated encrypt	Status word	API-Input	API-Output	AEAD Cipher	CO - SC_EDK: W,E,Z
pcrypt_aesccm_init	Initialize AES CCM struct	Status word	API-Input	API-Output	AEAD Cipher	CO - SC_EDK: W,E,Z
pcrypt_ecc_export_import	Extract ECC key (from struct); Initialize ECC struct	Status word	API-Input	API-Output	None	CO - ECC_SGK: W - ECC_SVK: W
pcrypt_ecc_init_key	Initialize an ECC key struct	Status word	API-Input	API-Output	None	CO - ECC_SGK: W - ECC_SVK: W
pcrypt_ecc_gen_key	Generate ECC key pair	Status word	API-Input	API-Output	ECC key generation	CO - ECC_Private: G,R - ECC_Public: G,R
pcrypt_ecc_sign	Generate ECDSA signature	Status word	API-Input	API-Output	ECDSA	CO - ECC_SGK: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
pcrypt_ecc_verify_hash	Verify ECDSA signature	Status word	API-Input	API-Output	ECDSA	CO - ECC_SVK: E
pcrypt_hmac_init	Initialize HMAC	Status word	API-Input	API-Output	HMAC	CO - HMAC_HMK: W
pcrypt_hmac_update	Update HMAC	Status word	API-Input	API-Output	HMAC	CO - HMAC_HMK: E
pcrypt_hmac_finalize	Generate HMAC tag	Status word	API-Input	API-Output	HMAC	CO - HMAC_HMK: E
pcrypt_init	Initialize the Module; executes FW integrity test and all CASTs	Status word	API-Input	API-Output		CO - PW_Entry: W - PW_Ref: E
pcrypt_key_exchange	Key agreement and subsequent derivation of keying material from a shared secret	Status word	API-Input	API-Output	Key agreement Key derivation	CO - KAS_DKM: R - KAS_SS: E - KAS_U_Private: E - KAS_V_Public: E
pcrypt_rng_generate_block	Generate random bits	Status word	API-Input	API-Output	Random	CO - RBG_State_C: E - RBG_State_V: E
pcrypt_rng_init	Instantiate DRBG	Status word	API-Input	API-Output	Random	CO - RBG_EI: G,E,Z - RBG_Seed: G,E,Z - RBG_State_C: G,R - RBG_State_V: G,R
pcrypt_rsa_gen_key	Generate RSA key pair	Status word	API-Input	API-Output	RSA key generation	CO - RSA_Private: G,R - RSA_Public: G,R
pcrypt_rsa_init	Initialize RSA struct	Status word	API-Input	API-Output		CO
pcrypt_rsa_sign	RSA sign a hashed message	Status word	API-Input	API-Output	RSA	CO - RSA_SGK: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
pcrypt_rsa_verify	Verify RSA signature	Status word	API-Input	API-Output	RSA	CO - RSA_SVK: E
pcrypt_sha256_hash	Generate a message digest	Status word	API-Input	API-Output	Message Digest	CO
pcrypt_sha384_hash	Generate a message digest	Status word	API-Input	API-Output	Message Digest	CO
pcrypt_selftest	On-demand invocation of self-tests	Status word	API-Input	API-Output	None	CO
Show status	pcrypt_status - Provide Module status	Status word	API-Input	API-Output	None	CO
pcrypt_tamper_detected	Tamper detection	Status word	API-Input	API-Output	None	CO
Show version	Provided in pcrypt_status call.CMVP automation rule checking requires the service name 'Show Version'	Status word	API-Input	API-Output	None	CO
Zeroize	Various pcrypt_*_free methods and pcrypt_uninit.CMVP automation rule checking requires the service name 'Zeroize'	Status word	API-Input	API-Output	None	CO - ECC_Private: Z - ECC_Public: Z - ECC_SGK: Z - ECC_SVK: Z - HMAC_HMK: Z - KAS_DKM: Z - KAS_SS: Z - KAS_U_Private: Z - KAS_V_Public: Z - PW_Entry: Z - RBG_EI: Z - RBG_Seed: Z - RBG_State_C: Z - RBG_State_V: Z - RSA_Private: Z - RSA_Public: Z - RSA_SGK: Z - RSA_SVK: Z - SC_EDK: Z

Table 14: Approved Services

The *pcrypt_status* service does not require authentication and provides information to address AS04.14 (output current status). The Module is similar to FIPS 140-3 IG 2.4.C Scenario 2, where *pcrypt_status* provides a global dynamic indication of Module status and operation in the approved mode, augmented by a status value returned on each API call. The *Show version* service addresses AS04.13 (show module's versioning information). The UID information returned by *Show version* contains the module identifier ("SCSM"), hardware identifier (0x481 [indicating STM32U5A5] or 0x482 [indicating STM32U585]) and firmware version ("1.2.0"), consistent with the Module's CMVP listing information.

The Module provides a status word to the external calling process, inclusive of an enumerated status return value, encoded in two bytes. The least significant byte gives status as one of the following:

PCRYPT_STATUS_READY: Module operation successful/normal
PCRYPT_STATUS_NOINIT: Module not initialized (Default start-up state)
PCRYPT_STATUS_STFAIL: Module self-test failure
PCRYPT_STATUS_BADARG: Module passed invalid argument
PCRYPT_STATUS_DISABLED: Module is disabled (tamper detected)
PCRYPT_STATUS_ERROR: Module internal error

The Show Status service reflects overall module status, including the status of all disjoint components of the module.

Draft IG 1.B calls for explanation – in several places - of the breakdown of module functionality between the firmware and hardware components of the Module. This topic is addressed in Section 2.2 below the block diagram, in an effort to describe the topic in its entirely in one place in this document.

4.4 Non-Approved Services

N/A for this Module.

4.5 External Software/Firmware Loaded

N/A for this Module. The Module does not support loading of firmware from an external source.

5 Software/Firmware Security

5.1 Integrity Techniques

The executable form of the disjoint firmware component of the Module is a firmware library statically linked in the SafeCase firmware. During initialization (without operator intervention and prior to operation), it performs an ECDSA P-384 with SHA2-384 signature verification over all files in the Module boundary.

5.2 Initiate on Demand

The operator can initiate the integrity test on demand by either power cycling the Module or by invoking the *pcrypt_init* service.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Single-chip packaging	The Module is intended to be mounted in additional packaging; physical inspection of the chip is not practical after packaging	N/A

Table 15: Mechanisms and Actions Required

The Module is a single-chip embodiment as shown in Figure 1 with a tamper-evident hard coating applied to it. The single-chip packaging is opaque in the visible spectrum.

No actions are required by the operator to ensure that the physical security is maintained.

7.2 EFP/EFT Information

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature	-40°C	EFP	Shutdown
HighTemperature	+85°C	EFP	Shutdown
LowVoltage	1.71V	EFP	Shutdown
HighVoltage	3.6V	EFP	Shutdown

Table 16: EFP/EFT Information

7.3 Hardness Testing Temperature Ranges

Temperature Type	Temperature
LowTemperature	-40°C
HighTemperature	+85°C

Table 17: Hardness Testing Temperatures

8 Non-Invasive Security

N/A for this Module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	R: Random access memory	Dynamic

Table 18: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API-I	Calling process	Call stack (API) input parameters	Plaintext	Automated	Electronic	
API-O	Call stack (API) output parameters	Calling process	Plaintext	Automated	Electronic	

Table 19: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
pcrypt_uninit	Zeroize DRBG state	Overwrites with zeros	API call
pcrypt_aes_free	Zeroize SSPs and release struct memory	Overwrites with zeros	API call
pcrypt_ecc_free	Zeroize SSPs and release struct memory	Overwrites with zeros	API call
pcrypt_hmac_free	Zeroize SSPs and release struct memory	Overwrites with zeros	API call
pcrypt_rsa_free	Zeroize SSPs and release struct memory	Overwrites with zeros	API call

Table 20: SSP Zeroization Methods

All functions zeroize SSPs within the function scope after use. Call stack cleanup is the responsibility of the application. The Module-provided methods to deallocate memory perform active zeroization (overwriting with zeros) prior to deallocation. The indicator of zeroization is a status word (SW) returned by the module as specified in Section 4.3. Zeroization of PW_Ref requires destruction of the firmware image (see Section 9.5).

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ECC_Private	General ECDSA (private) key.	384 - 192	P-384 - CSP	ECC key generation		ECC key generation
ECC_Public	General ECDSA (public) key; related to GKP_ Private.	384 - 192	P-384 - PSP	ECC key generation		ECC key generation
ECC_SGK	SigGen (private) key.	384 - 192	P-384 - CSP	ECC key generation		ECDSA
ECC_SVK	SigVer (public) key.	384 - 192	P-384 - PSP	ECC key generation		ECDSA
HMAC_HMK	Keyed Hash key.	384 - 384	HMAC - CSP			HMAC
KAS_DKM	Key derivation derived keying material.	112, 256, 384 - 112, 256, 384	Other - CSP	Key derivation		Key derivation
KAS_SS	Shared secret calculation z output value (for KDF).	192 - 192	Other - CSP	Key derivation		Key derivation
KAS_U_Private	Key pair component used for shared secret generation.	384 - 192	P-384 - CSP			Key agreement
KAS_V_Public	Peer key pair component used for shared secret generation.	384 - 192	P-384 - PSP			Key agreement
PW_Entry	Module password	256 - 256	Other - CSP			
PW_Ref	Module password - stored reference value.	256 - 256	Other - CSP			
RBG_EI	Entropy Input and nonce from on-chip TRNG.	1024 - 256	DRBG_EI - CSP	Random		Random
RBG_Seed	Seed used for Hash DRBG Instantiation and Reseed.	440 - 256	DRBG_EI plus optional personalization string - CSP	Random		Random
RBG_State_C	Element of Hash DRBG state.	440 - 256	Hash_DRBG_C - CSP	Random		Random
RBG_State_V	Element of Hash DRBG state.	440 - 256	Hash_DRBG_V - CSP	Random		Random
RSA_Private	General RSA (private) key.	2048 - 112	k=2048 - CSP	RSA key generation		RSA key generation
RSA_Public	General RSA (public) key.	2048 - 112	k=2048 - PSP	RSA key generation		RSA key generation
RSA_SGK	SigGen (private) key.	2048 - 112	k=2048 - CSP			RSA
RSA_SVK	SigVer (public) key.	2048 - 112	k=2048 - PSP			RSA
SC_EDK	Symmetric encryption and decryption.	256 - 256	AES-256 - CSP			AEAD Cipher

Table 21: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ECC_Private	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_ecc_free	ECC_Public:Paired with
ECC_Public	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_ecc_free	ECC_Private:Paired with
ECC_SGK	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_ecc_free	ECC_SVK:Paired with
ECC_SVK	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_ecc_free	ECC_SGK:Paired with
HMAC_HMK	API-I	RAM:Plaintext	Call lifetime	pcrypt_hmac_free	
KAS_DKM	API-O	RAM:Plaintext	Call lifetime	pcrypt_uninit	:
KAS_SS		RAM:Plaintext	Call lifetime	pcrypt_uninit	:
KAS_U_Private	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_ecc_free	KAS_V_Public:Paired with
KAS_V_Public	API-I	RAM:Plaintext	Call lifetime	pcrypt_ecc_free	KAS_U_Private:Paired with
PW_Entry	API-I	RAM:Plaintext	Call lifetime	pcrypt_uninit	PW_Ref:Compared to
PW_Ref		RAM:Plaintext	Call lifetime		PW_Entry:Compared to
RBG_EI		RAM:Plaintext	Call lifetime	pcrypt_uninit	RBG_Seed:Contains
RBG_Seed		RAM:Plaintext	Call lifetime	pcrypt_uninit	RBG_EI:Component of
RBG_State_C		RAM:Plaintext	Call lifetime	pcrypt_uninit	RBG_State_V:Used with RBG_Seed:Derived from
RBG_State_V		RAM:Plaintext	Call lifetime	pcrypt_uninit	RBG_State_C:Used with RBG_Seed:Derived from
RSA_Private	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_rsa_free	RSA_Public:Paired with
RSA_Public	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_rsa_free	RSA_Private:Paired with
RSA_SGK	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_rsa_free	RSA_SVK:Paired with
RSA_SVK	API-I API-O	RAM:Plaintext	Call lifetime	pcrypt_rsa_free	RSA_SGK:Paired with
SC_EDK	API-I	RAM:Plaintext	Call lifetime	pcrypt_aes_free	

Table 22: SSP Table 2

9.5 Additional Information

The *pcrypt_uninit* command zeroizes CSPs managed internal to the Module, returning the Module to the uninitialized state. The Module itself does not manage persistent SSPs or any user data; the only CSP that persists is the authentication password, PW_Ref. Removal of this SSP entails destruction of the Module (i.e. zeroization of the firmware image). This cannot be achieved by the Module itself but requires the calling application to instruct the device's bootloader to erase the firmware image.

There are no additional procedures required for secure destruction of the Module.

Key/Algorithm Type Equivalent Strengths: Reference sources for the strengths provided in SSP Table 1 are specified below. Equivalent strength is given for each key or algorithm type (as some algorithms do not use or produce keys).

- AES (AES-256): SP 800-57 Part 1 Rev. 5 Table 2.
- ECC (P-384): SP 800-186 Table 1.
    - Approved elliptic curves for ECC key agreement are given in SP 800-56A Rev. 3 Table 24.
- KDA One-Step KDF (SHA2-384): SP 800-56C Rev. 2 Table 1.
- IFC (k=2048): SP 800-56B Rev. 2 Table 4.
    - IFC key types approved for Digital Signature Generation and Verification are given also in SP 800-57 Part 1 Rev. 5 Table 2.
    - In Digital Signature applications, security strength is primarily associated with the asymmetric key pair specification. The hash function used must have equivalent strength equal to or greater than the security strength of the associated key pair.
- SHA2 (SHA2-256, SHA2-384): SP 800-107 Rev. 1 Table 1.
    - Preimage resistance strength applies to hash algorithms used in DRBG, KDFs. Described also in SP 800-57 Part 1 Rev. 5 Table 3.

Note: The minimum approved security strength will transition from 112 bits to 128 bits after the year 2030 according to SP 800-57 Part 1 Rev 5 Table 4.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
FW Integrity	ECDSA SigVer (P-384, SHA2-384) #A7216	ECDSA signature verification over all firmware in the module	SW/FW Integrity	PCRYPT_STATUS_STFAIL	Verify

Table 23: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CCM	256-bit	KAT	CAST	PCRYPT_STATUS_STFAIL	Encrypt	Performed on module load.
AES CCM Decrypt	256-bit	KAT	CAST	PCRYPT_STATUS_STFAIL	Decrypt	Performed on module load.
ECDSA SigGen (FIPS186-5)	P-384 with SHA2-384	KAT	CAST	PCRYPT_STATUS_STFAIL	Sign	Performed on module load.
ECDSA SigVer (FIPS186-5)	P-384 with SHA2-384	KAT	CAST	PCRYPT_STATUS_STFAIL	Verify	Performed on module load.
ESV (P)		APT, RCT	CAST	PCRYPT_STATUS_STFAIL	Within TRNG hardware, with configurable APT, RCT	Continuous (prior to use) or on demand via pcrypt_selftest service.
Hash DRBG	SHA2-256	KAT	CAST	PCRYPT_STATUS_STFAIL	Instantiate, Generate, Reseed	Performed on module load.
HMAC-SHA2-384	SHA2-384 with a 192-bit key	KAT	CAST	PCRYPT_STATUS_STFAIL	Generate	Performed on module load.
KAS-ECC-SSC Sp800-56Ar3	P-384	KAT	CAST	PCRYPT_STATUS_STFAIL	Ephemeral Unified Shared Secret (Z) Computation	Performed on module load.
KAS-KDF OneStep SP800-56Cr2	SHA2-384	KAT	CAST	PCRYPT_STATUS_STFAIL	[SP800-56Cr2] Section 4 OneStep KDF (AKA OpenSSL single-step or SS-KDF)	Performed on module load.
RSA SigGen (FIPS186-5)	k=2048 with SHA2-256	KAT	CAST	PCRYPT_STATUS_STFAIL	Sign	Performed on module load.
RSA SigVer (FIPS186-5)	k=2048 with SHA2-256	KAT	CAST	PCRYPT_STATUS_STFAIL	Verify	Performed on module load.
ECDSA KeyGen (FIPS186-5)	PCT performed using the generated key pair	PCT	PCT	PCRYPT_STATUS_STFAIL	Sign, Verify	Performed on ECC (ECDSA, KAS-ECC-SSC) key pair generation, prior to returning the key pair on conclusion of the call.
RSA KeyGen (FIPS186-5)	PCT performed using the generated key pair	PCT	PCT	PCRYPT_STATUS_STFAIL	Sign, Verify	Performed on RSA key pair generation, prior to returning the key pair on conclusion of the call.

Table 24: Conditional Self-Tests

All cryptographic algorithm self-tests (CASTs) must complete successfully prior to any other use of cryptography by the Module. The ECDSA CASTs are performed prior to the firmware integrity test.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
FW Integrity	ECDSA signature verification over all firmware in the module	SW/FW Integrity	On demand	Module load

Table 25: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CCM	KAT	CAST	On demand	On power on or reset
AES CCM Decrypt	KAT	CAST	On demand	On power on or reset
ECDSA SigGen (FIPS186-5)	KAT	CAST	On demand	On power on or reset
ECDSA SigVer (FIPS186-5)	KAT	CAST	On demand	On power on or reset
ESV (P)	APT, RCT	CAST	Each use	Continuously running
Hash DRBG	KAT	CAST	On demand	On power on or reset
HMAC-SHA2-384	KAT	CAST	On demand	On power on or reset
KAS-ECC-SSC Sp800-56Ar3	KAT	CAST	On demand	On power on or reset
KAS-KDF OneStep SP800-56Cr2	KAT	CAST	On demand	On power on or reset
RSA SigGen (FIPS186-5)	KAT	CAST	On demand	On power on or reset
RSA SigVer (FIPS186-5)	KAT	CAST	On demand	On power on or reset
ECDSA KeyGen (FIPS186-5)	PCT	PCT	On demand	On power on or reset
RSA KeyGen (FIPS186-5)	PCT	PCT	On demand	On power on or reset

Table 26: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Self-test failure	The self-test failure error state	Self-test failure	Reload the Module into memory	PROV_R_FIPS_MODULE_IN_ERROR_STATE

Table 27: Error States

10.5 Operator Initiation of Self-Tests

Each time the Module is powered on, it tests that the cryptographic algorithms still operate correctly and that sensitive data has not been damaged. CASTs are available on demand and can be tested periodically using the *pcrypt_selftest* command. The integrity test can be run on demand by either power cycling the Module or by invoking the *pcrypt_init* service.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Module FIPS 140-3 Guidance documentation describes all procedures for secure installation, initialization, configuration, provisioning, decommissioning and sanitization of the Module. The Module is a component of the SafeCase product, integrated in the Privoro manufacturing setting and thus no further installation procedures are required of the Crypto Officer. The initialization process for the Module involves loading the Module and successfully authenticating to it as the Crypto Officer using the *pcrypt_init* service.

There are no maintenance requirements for the Module.

11.2 Administrator Guidance

The Guidance documentation is inclusive of all information required per ISO/IEC 19790:2012 Section 7.11.9.

11.3 Non-Administrator Guidance

N/A for this Module.

11.4 Design and Rules

The Module enforces the following security rules:

1. No additional interface or service is implemented by the Module which would provide access to SSPs.
2. Data output is inhibited during key generation, self-tests, zeroisation, and error states.
3. There are no restrictions on which keys or SSPs are zeroised by the zeroisation service.
4. The Module does not support manual key entry.
5. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the Module.

The Module design corresponds to the Module security rules.

12 Mitigation of Other Attacks

12.1 Attack List

The Module's modular exponentiation, ECC scalar multiplication and ECDSA signature generation implement differential power analysis (DPA) protections.

12.2 Mitigation Effectiveness

The Module's DPA protections were examined by a 3rd party laboratory in cooperation with the hardware vendor. The process involved:

- A security model analysis inclusive of specifying applicable threat models.
- An assessment of the module using power monitoring and statistical analysis, demonstrating protection with and without countermeasures.
- A test process based on attack potential, factoring in attack (elapsed) time, attacker expertise, knowledge of the target, and ready access to targets and the type equipment available for attack. In this case, the protections were assessed as passing relative to the Very High attack potential, as defined by the SESIP/PSA Level 3 methodology.

12.3 Guidance and Constraints

The Module does not have any special handling (installation or configuration) requirements, nor identified constraints on the DPA protections.