# SafeCase Security Module

# FIPS 140-3 Non-Proprietary Security Policy

Document Version D18

July 01, 2025

**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

This document defines the non-proprietary Security Policy for the SafeCase Security Module by 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;
- does not implement mitigations of attacks outside the [FIPS140-3] specification.

The Module is validated to FIPS 140-3 overall Level 2 requirements with security levels as follows:

*Table 1: Security Levels*

| ISO/IEC 24759 Section 6. [Number Below] | FIPS 140-3 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 Parameters Management | 2 |
| 10 | Self-tests | 2 |
| 11 | Life-cycle Assurance | 2 |
| 12 | Mitigation of Other Attacks | N/A |

# 2 Cryptographic Module Specification

The Module is a firmware-hybrid Module as defined by [I19790], operated on a single-chip. The Module provides cryptographic services for use in SafeCase mobile device management systems. The tested Operational Environments are specified in Table 2 below.

*Table 2: Tested Operational Environments*

| # | Operating System | Hardware Platform | Processor | PAA/Acceleration |
|---|---|---|---|---|
| 1 | Free-RTOS | NXP MK81FN256VDC15 | NXP Kinetis K81FN256xxx15 | PAA is enabled for the K81[1] |

The module comprises of the disjoint firmware component libpcrypt.a and the disjoint hardware component, the Memory-Mapped Cryptographic Acceleration Unit (MMCAU) within the NXP MK81FN256VDC15 single-chip which is the physical perimeter (TOEPP) of the module. The UID information returned by pcrypt_status contains name (“SCSM”), hardware version (0x81001BD9) for the MMCAU disjoint hardware component and firmware version (“1.0.1”), consistent with the Module’s CMVP listing information and tabularized below:

[1] K81= NXP MK81FN256VDC15

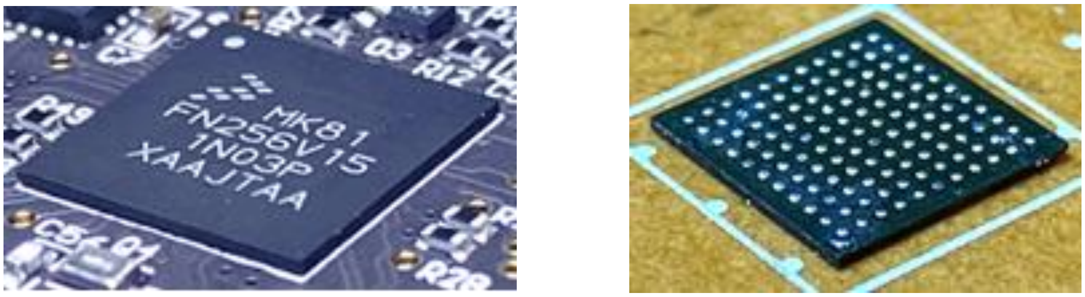
*Table 3: Tested Module Versioning Information/Identification*

| # | Firmware Component | Firmware Version | Hardware Component | Hardware Version | Module Identifier |
|---|---|---|---|---|---|
| 1 | libpcrypt.a | 1.0.1 | Hardware accelerator (MMCAU) within the NXP MK81FN256VDC15 i.e. K81 chip | 0x81001BD9 | SCSM |

Please Note: The disjoint firmware component of the module is the libpcrypt.a statically linked library (version 1.0.1) and the disjoint hardware component of the module is the hardware accelerator (MMCAU) version 0x81001BD9. The module is identified by the identifier returned by the module, i.e. the 'SCSM' string as detailed in the table above.

## 2.1 Cryptographic Boundary

The physical form of the Module is depicted below. The physical perimeter (i.e. the Tested Operational Environment's Physical Perimeter (TOEPP)) consists of the surfaces, edges and solder connections of the NXP MK81FN256VDC15 integrated circuit shown in Figure 1.



Top

Bottom

*Figure 1: Module Physical Perimeter*

The module is a hybrid module with the disjoint firmware component, libpcrypt.a (executing on the ARM CPU core) and the disjoint hardware component (MMCAU) forming the cryptographic boundary as depicted in the Figure 2 (outlined in red). The libpcrypt.a uses AES hardware cryptographic accleration from the MMCAU. Both the disjoint components reside within the NXP K81 SoC i.e. single chip.

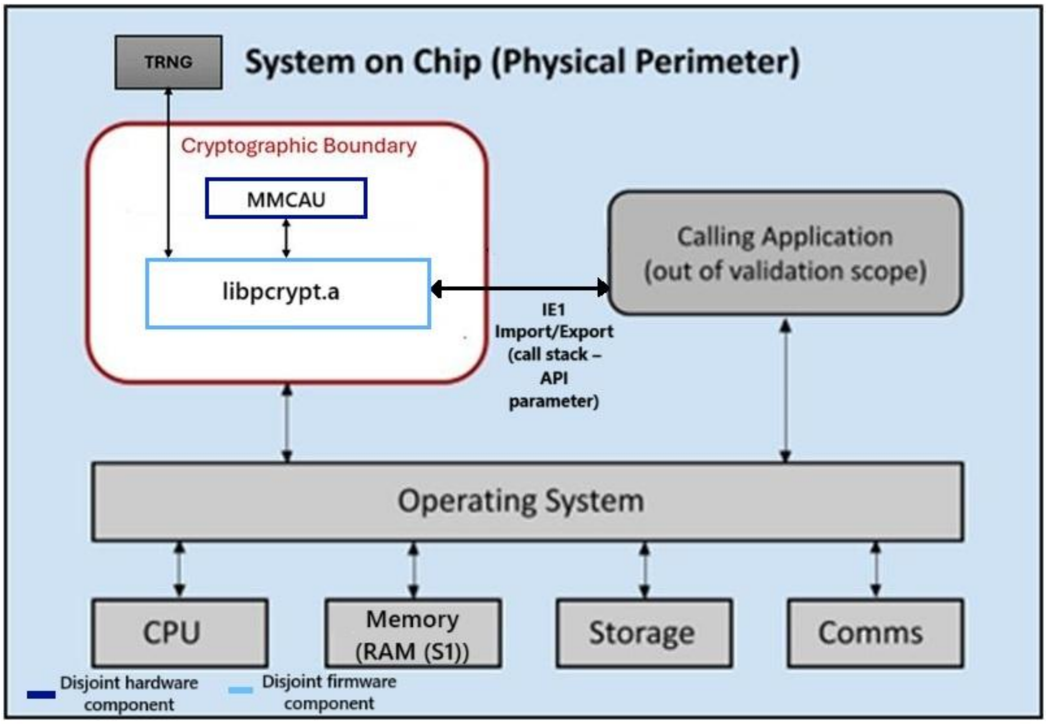

TRNG
System on Chip (Physical Perimeter)
Cryptographic Boundary
MMCAU
libpcrypt.a
Calling Application (out of validation scope)
IE1 Import/Export (call stack – API parameter)
Operating System
CPU
Memory (RAM (S1))
Storage
Comms
Disjoint hardware component
Disjoint firmware component

*Figure 2: Module Block Diagram*

## 2.2 Modes of Operation, Security Rules and Guidance

The Module supports only an Approved mode of operation by default, and 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. The Module does not output plaintext CSPs or intermediate key values.
6. 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. No specific installation requirements apply and the initialization process of the Module is as detailed in Section 11 of this document.

## 2.3 Degraded Operation

The Module does not support a degraded mode of operation.

## 2.4 Approved and Allowed Cryptographic Functionality

The Module implements the Approved cryptographic functions listed in Table 4. Equivalent strength in bits is given for each key or algorithm type (as some algorithms do not use or produce keys). The term *s* is used throughout to indicate security strength, following the notation used in the majority of the sources (refer to the notes below Table 4). This table is referenced by Table 13 (SSPs). All references to the algorithm standards cited throughout this document can be found in the References section.

*Table 4: Approved Algorithms*

| CAVP Cert | Algorithm and Standard | Mode / Method | Description / Key Size(s) / Key Strength(s) | Use / Function |
|---|---|---|---|---|
| A2494 | AES-ECB [FIPS197], [SP800-38A] | AES-ECB | AES-256 (s = 256) | Encryption (used only as a pre-requisite for AES-CCM) |
| A2494 | AES-CCM [SP800-38C] | AES-CCM | AES-256 (s = 256) | Authenticated encryption |
| Vendor Affirmed | CKG [SP800-133r2] | Direct<br>The module supports the following sections per NIST [SP800-133r2]: 4, 5.1, 5.2, 5.3, 5.4, 6.1, 6.2.1, 6.4 and 6.5. | 256-bit | Citation of [SP800-133r2] compliance required per [FIPS140-3_IG] D.H Scenario 2 |
| A2494 | ECDSA KeyGen [FIPS186-4] | Secret Generation Mode: Extra Bits | P-384 (s ~= 192)<br>See Note 2 | Key generation |
| A2494 | ECDSA SigGen [FIPS186-4] | SigGen (tested with SHA2-384) | P-384 (s ~= 192)<br>See Note 2 | Signature generation |
| A2494 | ECDSA SigVer [FIPS186-4] | SigVer (tested with SHA2-384) | P-384 (s ~= 192)<br>See Note 2 | Signature verification |
| A2494 | ENT [SP800-90B] | ENT (P) | 1024-bit seed;<br>1024-bit nonce | DRBG seeding |
| A2494 | Hash DRBG [SP800-90Ar1] | No prediction resistance | SHA2-256 (s = 256) | Random bit generation |
| A2494 | HMAC-SHA2-384 [FIPS198-1] | Generate HMAC-SHA2-384 MAC | SHA2-384 (s = 384) | HMAC generation and verification |
| A2494 | KAS [SP800-56Ar3] | Schemes: Ephemeral Unified<br>Roles: Initiator, Responder<br>KAS-ECC-SSC curves: P-384<br>KDA One-Step KDF | KAS-ECC per IG D.F Scenario 2 path (2) option 2<br><br>P-384 curve providing 192 bits of encryption strength | Key agreement with key derivation using [SP800-56Cr1] KDA |
| A2494 | KAS-ECC-SSC [SP800-56Ar3] | Scheme: ephemeralUnified<br>KAS Role: initiator, responder<br>IG D.F Scenario 2 path 2 with an approved KDF per [SP800-56Ar3] | P-384 (s ~= 192)<br>See Note 2 and Note 3 | Shared secret generation |
| A2494 | KDA OneStep [SP800-56Cr1] | One-Step KDF<br>Auxiliary function method: SHA2-384 | SHA2-384 (112 ≤ s ≤ 384)<br>See Note 6 | Derivation of keying material from a KAS shared secret |
| A2494 | RSA KeyGen [FIPS186-4] | Key generation mode: B.3.3<br>Primality tests per Table C.3 | k=2048 (s ~= 112)<br>See Note 5 | Key generation |
| A2494 | RSA SigGen [FIPS186-4] | Signature type: PKCS 1.5 tested with k=2048 and SHA2-256 | k=2048 (s ~= 112)<br>See Note 4 and Note 5 | Signature generation |
| A2494 | RSA Signature Primitive [FIPS186-4] | Private Key format: standard<br>Public Exponent Mode: fixed | k=2048 (s ~= 112)<br>See Note 4 and Note 5 | Signature primitive operation |
| A2494 | RSA SigVer [FIPS186-4] | Signature type: PKCS 1.5 tested with k=2048 and SHA2-256 | k=2048 (s ~= 112)<br>See Note 4 and Note 5 | Signature verification |
| A2494 | SHA2-256 [FIPS180-4] | SHA2-256 | SHA2-256 (s = 256)<br>See Note 1 | Secure hash generation |

| CAVP Cert | Algorithm and Standard | Mode / Method | Description / Key Size(s) / Key Strength(s) | Use / Function |
|---|---|---|---|---|
| A2494 | SHA2-384<br>[FIPS180-4] | SHA2-384 | SHA2-384 (s = 384)<br>See Note 1 | Secure hash generation |

**Note 1**: Preimage resistance strength applies to hash algorithms used in DRBG, KDFs. Described also in [SP800-57P1r5] Table 3.

**Note 2**: Elliptic curve strengths are annotated as approximate (i.e., s ~=) since [SP800-186] Table 1 provides approximate security strengths.

**Note 3**: Approved elliptic curves for ECC key agreement are given in [SP800-56Ar3] Table 24.

**Note 4**: 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.

**Note 5**: Estimated security strengths of common RSA moduli are given in [SP800-56Br2] Table 4. IFC key types approved for Digital Signature Generation and Verification are given also in [SP800-57P1r5] Table 2. Equivalent strengths are annotated as approximate (i.e., s ~=) since [SP800-56Br2] Table 4 provides approximate security strengths.

**Note 6**: Security strengths for KDA One Step are given in [SP800-56Cr1] Table 1 (hash).

Reference sources for the strengths provided in Table 4 are as follows:

- AES (AES-256): [SP800-57P1r5] Table 2.
- ECC (P-384): [SP800-186] Table 1.
- IFC (k=2048): [SP800-56Br2] Table 4.
- SHA2 (SHA2-256, SHA2-384): [SP800-107r1] Table 1.

*Table 5: Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed*

| Algorithm | Caveat | Use/Function |
|---|---|---|
| RSADP | No security claimed | Decryption of derived keying material, [FIPS 140-3_IG] 2.4.A Scenario 1 |

Please see the note below Table 9 pertaining to usage of RSADP.

The module does not implement the following:

- Non-Approved Algorithms Allowed in the Approved Mode of Operation
- Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

# 3 Cryptographic Module Interfaces

The Module's logical interfaces are described in Table 6; the Module's physical ports are outside the cryptographic boundary.

*Table 6: Ports and Interfaces*

| Physical port | Logical Interface | Data that passes over port/interface |
|---|---|---|
| N/A: Internal (call stack) | Control Input | API entry point: stack frame including non-sensitive parameters |
| | Data Input | API call parameters passed by reference or value for cryptographic service input |
| | Status Output | API return value: enumerated status resulting from call execution |
| | Data Output | API call parameters passed by reference for cryptographic service output |

The Control Output interface is not applicable, as the module does not control other components.

# 4 Roles, Services and Authentication

The Module supports only the Cryptographic Officer (CO) role. It does not support multiple concurrent operators, a maintenance role or bypass capability. Operator authentication is described in Table 8.

*Table 7: Roles, Service Commands, Input and Output*

| Role | Service | Input | Output |
|---|---|---|---|
| CO | pcrypt_aesccm_decrypt | SC_EDK; ciphertext message | Plaintext message; status |
| CO | pcrypt_aesccm_encrypt | SC_EDK; plaintext message | Ciphertext message; status |
| N/A | pcrypt_aes_free (Perform zeroisation) | AES struct (includes SC_EDK) | Status (AES struct zeroised, freed) |
| CO | pcrypt_aesccm_init | SC_EDK | Initialized AES struct; status |
| CO | pcrypt_ecc_export_private | ECC struct inclusive of ECC_Private | ECC_Private; status |
| CO | pcrypt_ecc_export_public | ECC struct inclusive of ECC_Public | ECC_Public; status |
| CO | pcrypt_ecc_import_private | ECC_Private | Initialized ECC struct; status |
| CO | pcrypt_ecc_import_public | ECC_Public | Initialized ECC struct; status |
| CO | pcrypt_ecc_init_key | ECC_Private; ECC_Public | Initialized ECC struct; status |
| N/A | pcrypt_ecc_free (Perform zeroisation) | ECC struct (includes ECC_Private, ECC Public, ECC_SGK, ECC SVK) | Status |
| CO | pcrypt_ecc_gen_key | ECC key parameters | Initialized ECC struct; status |
| CO | pcrypt_ecc_sign | ECC_SGK; plaintext message | Signature; status |
| CO | pcrypt_ecc_verify_hash | ECC_SVK; plaintext message; signature | Status |
| CO | pcrypt_hmac_init | HMAC_MHK, uninitialized HMAC struct | Initialized HMAC struct, status |
| CO | pcrypt_hmac_update | HMAC struct, message | Status |
| CO | pcrypt_hmac_finalize | HMAC struct, output buffer pointer | Output buffer containing HMAC tag |
| N/A | pcrypt_hmac_free (Perform zeroisation) | HMAC struct (includes HMAC_HMK) | Status |
| CO | pcrypt_init | Password; memory management parameters | Status |
| CO | pcrypt_key_exchange | KAS_U_Private; KAS_V_Public; KAS_SS | KAS_DKM; status |
| CO | pcrypt_rng_generate_block | RBG_State | Random bit string; RBG_State; status |
| CO | pcrypt_rng_init | RBG_EI; RBG_State; parameters; flags | DRBG struct; status |
| CO | pcrypt_rsa_export_private | RSA struct inclusive of RSA_Private | RSA_Private; status |
| CO | pcrypt_rsa_export_public | RSA struct inclusive RSA_Public | RSA_Public; status |
| CO | pcrypt_rsa_import_private | RSA_Private | Initialized RSA struct; status |
| N/A | pcrypt_rsa_free (Perform zeroisation) | RSA struct (includes RSA_Private, RSA Public, RSA_SGK, RSA_SVK) | Status |
| CO | pcrypt_rsa_gen_key | RSA parameters | Initialized RSA struct; status |
| CO | pcrypt_rsa_init | RSA_Private; RSA_Public | Initialized RSA struct; status |
| CO | pcrypt_rsa_sign | RSA_SGK; plaintext message | Signature; status |
| CO | pcrypt_rsa_verify | RSA_SVK; plaintext message; signature | Status |
| CO | pcrypt_sha256_hash | Plaintext message | Message digest; status |
| CO | pcrypt_sha384_hash | Plaintext message | Message digest; status |
| CO | pcrypt_selftest (Perform self-tests) | None | Status |
| N/A | pcrypt_status (Show status and Show module's versioning information) | Valid pointer to UID structure (optional) | Status: READY or ERROR<br>UID: name, version; hardware version |
| CO | pcrypt_tamper_detected | None | Status |

| Role | Service | Input | Output |
|---|---|---|---|
| CO | pcrypt_uninit (Perform zeroisation) | None | Status |
| N/A | pcrypt_update_time | Time value | None |

The pcrypt_status service does not require authentication and provides information to address both AS04.13 (Show module's versioning information) and AS04.14 (output current status). The Module is similar to [FIPS140-3_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 module supports role-based authentication.

*Table 8: Roles and Authentication*

| Role | Authentication Method | Authentication Strength |
|---|---|---|
| CO | 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<br>9.65E+71 in 1 minute |

Table 9 describes all Approved services and service access to SSPs. The following annotations indicate the type of access by the Module service:

**G = Generate**: The Module generates or derives the SSP.
**R = Read**: The SSP is read from the Module (e.g., the SSP is output).
**W = Write**: The SSP is updated, imported, or written to the Module.

**E = Execute**: The Module uses the SSP in performing a cryptographic operation.
**Z = Zeroise**: The Module zeroises the SSP.

*Table 9: Approved Services*

| Service | Description | Approved Security Functions | Keys and/or SSPs | Roles | Access rights to Keys and/or SSPs | Indicator |
|---|---|---|---|---|---|---|
| pcrypt_aesccm_decrypt | Authenticated decrypt | AES-CCM #A2494<br>AES-ECB #A2494 | SC_EDK | CO | WEZ | SW |
| pcrypt_aesccm_encrypt | Authenticated encrypt | AES-CCM #A2494<br>AES-ECB #A2494 | SC_EDK | CO | WEZ | SW |
| pcrypt_aesccm_init | Initialize AES CCM struct | N/A | SC_EDK | CO | WRZ | SW |
| pcrypt_ecc_export_private,<br>pcrypt_ecc_export_public,<br>pcrypt_ecc_import_private,<br>pcrypt_ecc_import_public | Extract ECC key (from struct)<br>Initialize ECC struct | N/A | ECC_Private<br>ECC_Public | CO | WRZ<br>WRZ | SW |
| pcrypt_ecc_init_key | Initialize an ECC key struct | N/A | ECC_Private<br>ECC_Public | CO | WRZ<br>WRZ | SW |
| pcrypt_ecc_gen_key | Generate ECC key pair | ECDSA KeyGen #A2494<br>CKG | ECC_Private<br>ECC_Public | CO | GRZ<br>GRZ | SW |
| pcrypt_ecc_sign | Generate ECDSA signature | ECDSA SigGen #A2494 | ECC_SGK | CO | EWZ | SW |
| pcrypt_ecc_verify_hash | Verify ECDSA signature | ECDSA SigVer #A2494 | ECC_SVK | CO | EWZ | SW |
| pcrypt_hmac_init | Initialize HMAC | HMAC-SHA2-384 #A2494 | HMAC_HMK | CO | WE | SW |
| pcrypt_hmac_update | Update HMAC | HMAC-SHA2-384 #A2494 | HMAC_HMK | CO | E | SW |
| pcrypt_hmac_finalize | Generate HMAC tag | HMAC-SHA2-384 #A2494 | HMAC_HMK | CO | EZ | SW |
| pcrypt_init | Initialize the Module; executes FW integrity test and all CASTs | SHA2-256 #A2494,<br>SHA2-384 #A2494 | PW_Entry<br>PW_Ref | CO | E | SW |
| pcrypt_key_exchange | Key agreement and subsequent derivation of keying material from a shared secret | KAS-ECC-SSC #A2494,<br>KDA OneStep #A2494 | KAS_U_Private<br>KAS_V_Public<br>KAS_SS<br>KAS_DKM | CO | EWZ<br>EWZ<br>GEZ<br>GRZ | SW |
| pcrypt_rng_generate_block | Generate random bits | Hash DRBG #A2494 | RBG_State | CO | WER | SW |

| Service | Description | Approved Security Functions | Keys and/or SSPs | Roles | Access rights to Keys and/or SSPs | Indicator |
|---|---|---|---|---|---|---|
| | | | RBG_Seed | | | |
| pcrypt_rng_init | Instantiate DRBG | Hash DRBG #A2494 | RBG_EI<br>RBG_State | CO | EZ<br>EG | SW |
| pcrypt_rsa_export_private,<br>pcrypt_rsa_export_public,<br>pcrypt_rsa_import_private | Extract RSA key (from struct)<br>Initialize RSA struct | N/A | RSA_Private<br>RSA_Public | CO | WRZ | SW |
| pcrypt_rsa_gen_key | Generate RSA key pair | RSA KeyGen #A2494<br>CKG | RSA_Private<br>RSA_Public | CO | GRZ | SW |
| pcrypt_rsa_init | Initialize RSA struct | N/A | RSA_Private<br>RSA_Public | CO | WRZ | SW |
| pcrypt_rsa_sign | RSA sign a hashed message | RSA SigGen #A2494<br>RSA Signature Primitive #A2494 | RSA_SGK | CO | WEZ | SW |
| pcrypt_rsa_verify | Verify RSA signature | RSA SigVer #A2494 | RSA_SVK | CO | WEZ | SW |
| pcrypt_sha256_hash | Generate a message digest | SHA2-256 #A2494 | N/A | CO | N/A | SW |
| pcrypt_sha384_hash | Generate a message digest | SHA2-384 #A2494 | N/A | CO | N/A | SW |
| pcrypt_selftest (Perform self-tests) | On-demand invocation of self-tests | N/A | N/A | CO | N/A | SW |
| pcrypt_status (Show status and Show module's versioning information) | Provide Module status | N/A | N/A | N/A | N/A | SW |
| pcrypt_tamper_detected | Tamper detection | N/A | N/A | CO | N/A | SW |
| pcrypt_uninit<br>pcrypt_aes_free<br>pcrypt_ecc_free<br>pcrypt_hmac_free<br>pcrypt_rsa_free<br>(Perform zeroisation) | Zeroise the DRBG and release struct memory, zeroising SSPs | N/A | RBG_EI<br>RBG_State<br>RBG_Seed<br>SC_EDK<br>ECC_Private<br>ECC_Public<br>ECC_SGK<br>ECC_SVK<br>HMAC_HMK<br>RSA_Private<br>RSA_Public<br>RSA_SGK<br>RSA_SVK<br>KAS_DKM<br>KAS_U_Private<br>KAS_V_Public<br>KAS_SS | N/A | Z | SW |
| pcrypt_update_time | Update module time | N/A | N/A | N/A | N/A | SW |

SW refers to the 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 startup 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 most significant byte indicates the use of a non-Approved algorithm; at the time of validation, this only applies to RSA Decrypt (RSADP). PCRYPT_STATUS_FIPS_NOALLOW: set to 1 if the service does not use an approved algorithm. Usage of RSADP is deemed non-Approved but allowed and conforms to [FIPS 140_3_IG] 2.4.A example scenario 1.

All functions zeroise 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 zeroisation (overwriting with zeros) prior to deallocation. Zeroisation of *PW_Ref* requires destruction of the firmware image via the SafeCase product Crypto Reset function (Table 13 "Z2"). As allowed by [SP800-140DTR] VE09.38.03, this mechanism is provided as a service of the SafeCase product. The indicator of zeroisation is a status word (SW) returned by the module as specified above.

# 5 Software/Firmware Security

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. The operator can initiate the integrity test on demand by either power cycling the Module or by invoking the pcrypt_init service. The module does not support loading of firmware from an external source.

# 6 Operational Environment

The Module executes in a limited operational environment as defined by [I19790]. The module does not support loading firmware from an external source.

# 7 Physical Security

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(s) to ensure that the physical security is maintained.

*Table 10: Physical Security Inspection Guidelines*

| Physical Security Mechanism | Recommended Frequency of Inspection/Test | Inspection/Test Guidance Details |
|---|---|---|
| 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 11: EFP/EFT*

| | Temperature or voltage measurement | Specify EFP or EFT | Specify if this condition results in a shutdown or zeroisation |
|---|---|---|---|
| Low Temperature | -40 C | EFP | Shutdown (inoperable state) |
| High Temperature | +105 C | EFP | Shutdown (inoperable state) |
| Low Voltage | 1.54 V | EFP | Shutdown (inoperable state) |
| High Voltage | 3.72 V | EFP | Shutdown (inoperable state) |

*Table 12: Hardness testing temperature ranges*

| | Hardness tested temperature measurement |
|---|---|
| Low Temperature | -40 C |

| | |
|---|---|
| High Temperature | +105 C |

# 8 Non-Invasive Security

[SP800-140F] currently does not define applicable metrics. The Module does not implement non-invasive security measures.

# 9 Sensitive Security Parameters Management

Table 13 summarizes the SSPs implemented by the Module.

*Table 13: SSPs*

| Key/SSP Name/Type | Strength[2] | Security Function and Cert. Number | Generation | Import/Export | Establishment | Storage | Zeroisation | Use & related keys |
|---|---|---|---|---|---|---|---|---|
| ECC_Private<br>CSP | 192 | ECDSA KeyGen #A2494<br>CKG | G2 | IE1<br>MD<br>/EE | -- | S1 | Z1 | Private component of ECC key pair generated on caller request (key pair purpose is unspecified); related to ECC_Public |
| ECC_Public<br>PSP | 192 | ECDSA KeyGen #A2494<br>CKG | G2 | IE1<br>MD<br>/EE | -- | S1 | Z1 | Public component of ECC key pair generated on caller request (key pair purpose is unspecified); related to ECC_Private |
| ECC_SGK<br>CSP | 192 | ECDSA SigGen #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Private key for ECC signature generation; related to ECC_SVK |
| ECC_SVK<br>PSP | 192 | ECDSA SigVer #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Public key for ECC signature verification; related to ECC_SGK |
| HMAC_HMK<br>CSP | 384 | HMAC-SHA2-384 #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Key to generate and verify and HMAC |
| KAS_DKM<br>CSP | 112 ≤ s ≤ 384 | KDA OneStep #A2494 | -- | IE1<br>MD<br>/EE | E2 | S1 | Z1 | Key Derivation derived keying material[3] |
| KAS_U_Private<br>CSP | 192 | KAS-ECC-SSC #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Private key pair component provided by the local participant, used for Diffie-Hellman shared secret generation; related to KAS_V_Public |
| KAS_V_Public<br>PSP | 192 | KAS-ECC-SSC #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Public key pair component provided by the local participant, used for Diffie-Hellman shared secret generation; related to KAS_U_Private |
| KAS_SS<br>CSP | 192 | KAS-ECC-SSC #A2494 | -- | -- | E1<br>E2 | S1 | Z1 | Shared secret calculation; z output value is expected to be used by a KDF |
| PW_Entry<br>CSP | 256 | N/A | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Authentication input |
| PW_Ref<br>CSP | 256 | SHA2-256,<br>SHA2-384 | -- | IE2 | -- | S2 | Z2 | Authentication reference (hashed) |
| RBG_EI<br>CSP | See Table 14 | ENT (P) | G3 | -- | -- | S1 | Z1 | Entropy input and nonce |

[2] Strength is provided in bits. Please refer to Table 4 and the notes below it for the strength provenance (traceability to applicable standards and special publications).

[3] The separation into specific keys is done outside the scope of the module but must be conformant to [SP800-56Cr1].

| Key/SSP Name/Type | Strength[2] | Security Function and Cert. Number | Generation | Import/Export | Establishment | Storage | Zeroisation | Use & related keys |
|---|---|---|---|---|---|---|---|---|
| RBG_Seed<br>CSP | 440 bits | ENT (P), Counter DRBG | G3 | -- | -- | S1 | Z1 | DRBG seed derived from the entropy input |
| RBG_State<br>CSP | 256 | Hash DRBG #A2494 | -- | -- | E3 | S1 | Z1 | Hash DRBG (SHA2-256) state: V (440 bits) and C (440 bits) |
| RSA_Private<br>CSP | 112 | RSA KeyGen #A2494<br>CKG | G1 | IE1<br>MD<br>/EE | -- | S1 | Z1 | Private component of RSA key pair generated on caller request (key pair purpose is unspecified); related to RSA_Public |
| RSA_Public<br>PSP | 112 | RSA KeyGen #A2494<br>CKG | G1 | IE1<br>MD<br>/EE | -- | S1 | Z1 | Public component of RSA key pair generated on caller request (key pair purpose is unspecified); related to RSA_Private |
| RSA_SGK<br>CSP | 112 | RSA SigGen #A2494<br>RSA Signature Primitive #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Private key for RSA signature generation; related to RSA_SVK |
| RSA_SVK<br>PSP | 112 | RSA SigVer #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | Public key for RSA signature verification; related to RSA_SGK |
| SC_EDK<br>CSP | 256 | AES-CCM #A2494<br>AES-ECB #A2494 | -- | IE1<br>MD<br>/EE | -- | S1 | Z1 | AES key used for symmetric encryption (including AES authenticated encryption) |

**Legend**

| Generation |
|---|
| G1: [FIPS186-4] RSA keypair generation |
| G2: [FIPS186-4] ECDSA keypair generation |
| G3: [SP800-90B] DRBG seed material |

| Import/Export |
|---|
| IE1: Call stack (API) parameters |
| IE2: Entered in manufacturing |

| Establishment |
|---|
| E1: [SP800-56Ar3] §5.7.1.2 ECC CDH |
| E2: [SP800-56Cr1] KDA OneStep |
| E3: [SP800-90Ar1] Hash_df; Instantiate; Generate; Reseed |

| Storage |
|---|
| S1: RAM |

| Zeroisation |
|---|
| Z1: Cleared after use, module initiated |
| Z2: Cleared by Crypto Reset, operator initiated |

*Table 14: Non-Deterministic Random Number Generation Specification*

| Entropy sources | Minimum number of bits of entropy | Details |
|---|---|---|
| K81[4]<br>ENT (P) | [SP800-90Ar1] *min_length*: 256 bits<br>[SP800-90Ar1] *seedlen*: 440 bits | [FIPS140-3_IG] 9.3.A: The Module generates entropy within the Module's physical perimeter: option 1(b) using a [SP800-90B] compliant ENT present on the SoC component<br>Per [SP800-90Ar1] Table 2, the SHA2-256 Hash DRBG requires 256 bits of entropy (equivalent to security strength) within the 440-bit *DRBG_Seed* value<br>As input to the [SP800-90Ar1] Hash_df, the Module collects 1024 bits of data from the ENT to use as entropy and nonce input. The [SP800-90B] compliant assessment supports at least |

[4] K81= NXP MK81FN256VDC15

| Entropy sources | Minimum number of bits of entropy | Details |
|---|---|---|
| | | 0.99 bits of entropy per bit of ENT output; as such the DRBG seeding material contains at least 1013 bits of entropy, well in excess of the requirement |

## 10 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. The disjoint firmware component of the module, i.e., the libpcrypt.a, performs an ECDSA P-384 with SHA2-384 signature verification over all files in the Module boundary during module initialization i.e. on every boot.

On power-on or reset, the Module performs the self-tests described below. 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. All CASTs are implemented as known answer tests.

If one of the CASTs or the firmware integrity test fails, the Module enters the STFAIL error state. The error state is persistent, and no services are available. All attempts to use the Module’s services result in the return of a non-zero error code, PCRYPT_STATUS_STFAIL. A power-cycle or reset of the Module is required to recover from an error state, causing it to retry all self-tests.

**Pre-Operational Self-Tests**

- FW integrity test: ECDSA #A2494 (P-384/SHA2-384) signature verification over all firmware in the Module boundary.

**Conditional Cryptographic Algorithm Self-Tests (CASTs)**

- AES-CCM #A2494: Encrypt CAST performed using reference inputs (256-bit AES key size). Covers self-test requirement for AES-ECB encrypt.
- AES-CCM #A2494: Decrypt CAST performed using reference inputs (256-bit AES key size).
- ECDSA SigGen #A2494: Signature generation CAST using reference inputs (P-384, SHA2-384).
- ECDSA SigVer #A2494: Signature verification CAST using reference inputs (P-384, SHA2-384).
- ENT: [SP800-90B] Startup and conditional tests; executes a suite of self-tests prior to raising a “valid” flag.
- ENT(P) developer defined health tests.
- Hash DRBG #A2494: Instantiate, Generate and Reseed CASTs using reference inputs on the SHA2-256 Hash DRBG.
- HMAC-SHA2-384 #A2494: HMAC CAST using SHA2-384 and reference inputs (192-bit HMAC key).
- KAS-ECC-SSC #A2494: KAS-ECC-SSC CASTs using reference inputs (P-384 curve).
- KDA OneStep #A2494: One-Step KDF CAST using a known shared secret reference input (256-bit z).
- RSA SigGen #A2494: Signature generation CAST using reference inputs (k = 2048).
- RSA SigVer #A2494: Signature verification CAST using reference inputs (k = 2048).
- SHA2-256 #A2494: Message digest generation CAST using SHA2-256.
- SHA2-384 #A2494: Message digest generation CAST using SHA2-384.

**Conditional Pairwise Consistency Tests (PCTs)**

- ECDSA KeyGen #A2494: Pairwise consistency test of generated key pair.
- RSA KeyGen #A2494: Pairwise consistency test of generated key pair.

## 11 Life-cycle Assurance

The Privoro SafeCase Security Module FIPS 140-3 Guidance [GD] 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.

## 12 Mitigation of Other Attacks

The Module does not implement mitigations of other attacks outside the scope of [FIPS140-3].

## References

- [FIPS140-3]: FIPS 140-3, Security Requirements for Cryptographic Modules, 3/22/2019
- [SP800-140DTR]: NIST SP 800-140, FIPS 140-3 Derived Test Requirements (DTR): CMVP Validation Authority Updates to ISO/IEC 24759, 3/20/2020
- [SP800-140A]: NIST SP 800-140A, CMVP Documentation Requirements: CMVP Validation Authority Updates to ISO/IEC 24759, 3/20/2020
- [SP800-140B]: NIST SP 800-140B, CMVP Security Policy Requirements: CMVP Validation Authority Updates to ISO/IEC 24759 and ISO/IEC 19790 Annex B, 3/20/2020
- [SP800-140Cr2]: CNIST SP 800-140C Rev. 2, Cryptographic Module Validation Program (CMVP)-Approved Security Functions: CMVP Validation Authority Updates to ISO/IEC 24759, 7/25/2023

  Supplemental Information: SP 800-140C: Approved Security Functions, 7/25/2023
- [SP800-140Dr2]: NIST SP 800-140D Rev. 2, Cryptographic Module Validation Program (CMVP)-Approved Sensitive Security Parameter Generation and Establishment Methods: CMVP Validation Authority Updates to ISO/IEC 24759, 7/25/2023

  Supplemental Information: SP 800-140D: Approved SSP Generation and Establishment Methods, 7/25/2023
- [SP800-140E]: NIST SP 800-140E, CMVP Approved Authentication Mechanisms: CMVP Validation Authority Requirements for ISO/IEC 19790 Annex E and ISO/IEC 24579 Section 6.17, 3/20/2020
- [SP800-140F]: NIST SP 800-140F, CMVP Approved Non-Invasive Attack Mitigation Test Metrics: CMVP Validation Authority Updates to ISO/IEC 24759, 3/20/2020
- [FIPS140-3_IG]: Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program, 1/29/2024
- [I19790]: ISO/IEC 19790:2012 Information technology -- Security techniques -- Security requirements for cryptographic modules, 11/1/2015
- [I24759]: ISO/IEC 24759:2017 Information technology -- Security techniques -- Test requirements for cryptographic modules, 3/1/2017
- [FIPS180-4]: FIPS 180-4, Secure Hash Standard (SHS), 8/4/2015
- [FIPS186-4]: FIPS 186-4, Digital Signature Standard (DSS), 7/19/2013
- [FIPS197]: FIPS 197, Advanced Encryption Standard (AES), 5/09/2023
- [FIPS198-1]: FIPS 198-1, The Keyed-Hash Message Authentication Code (HMAC), 7/16/2008
- [SP800-38A]: NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, 12/1/2001

- [SP800-38C]: NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, 7/20/2007
- [SP800-56Ar3]: NIST SP 800-56A Rev. 3, Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, 4/16/2018
- [SP800-56Br2]: NIST SP 800-56B Rev. 2, Recommendation for Pair-Wise Key-Establishment Using Integer Factorization Cryptography, 3/21/2019
- [SP800-56Cr1]: NIST SP 800-56C Rev. 1, Recommendation for Key-Derivation Methods in Key-Establishment Schemes, 4/16/2018
- [SP800-56Cr2]: NIST SP 800-56C Rev. 2, Recommendation for Key-Derivation Methods in Key-Establishment Schemes, 8/18/2020
- [SP800-57P1r5]: NIST SP 800-57 Part 1 Rev. 5, Recommendation for Key Management: Part 1 – General, 5/04/2020
- [SP800-63B]: NIST SP 800-63B, Digital Identity Guidelines: Authentication and Lifecycle Management, 3/02/2020
- [SP800-90Ar1]: NIST SP 800-90A Rev. 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, 6/24/2015
- [SP800-90B]: NIST SP 800-90B, Recommendation for the Entropy Sources Used for Random Bit Generation, 1/10/2018
- [SP800-107r1]: NIST SP 800-107 Rev. 1, Recommendation for Applications Using Approved Hash Algorithms, 8/24/2012
- [SP800-133r2]: NIST SP 800-133 Rev. 2, Recommendation for Cryptographic Key Generation, 6/04/2020
- [SP800-186]: NIST SP 800-186, Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters, 2/03/2023
- [GD]: SafeCase Security Module FIPS 140-3 Guidance, 12/17/2021

## Acronyms and Definitions

- CAVP: Cryptographic Algorithm Validation Program
- ACVTS: Automated Cryptographic Validation Testing System
- AES: Advanced Encryption Standard, see [FIPS197]
- API: Application Programming Interface
- CAST: Cryptographic Algorithm Self-Test
- CCM: Counter with CBC-MAC
- CKG: Cryptographic Key Generation
- CMVP: Cryptographic Module Validation Program
- CO: Cryptographic Officer
- CSP: Critical Security Parameter, see [FIPS140-3]
- CPU: Central Processing Unit
- CCCS: Canadian Centre for Cybersecurity
- DRAM: Dynamic Random-Access Memory
- DRBG: Deterministic Random Number Generator, see [SP800-90Ar1]
- DTR: Derived Test Requirements
- ECB: Electronic Code Book

- ECC: Elliptic Curve Cryptography
- ECDSA: Elliptic Curve Digital Signature Algorithm, see [FIPS186-4]
- FIPS: Federal Information Processing Standard
- HMAC: Keyed-Hash Message Authentication Code, see [FIPS198-1]
- IG: Implementation Guidance, see [FIPS140-3_IG]
- KAS: Key Agreement Scheme
- KDF: Key Derivation Function
- MAC: Message Authentication Code
- NIST: National Institute of Standards and Technology
- OE: Operating Environment
- PCT: Pairwise Consistency Test
- PSP: Public Security Parameter
- RSADP: RSA Decryption Primitive
- SHA/SHS: Secure Hash Algorithm/Standard, see [FIPS180-4]
- SP: NIST Special Publication
- SSC: Shared Secret Calculation
- SSP: Sensitive Security Parameter