gemalto
a Thales company

# FIPS 140-2 Non-Proprietary Security Policy

---

# Gemalto FIPS Object module

**Software Version 2.1.0**

**Date: April 22nd, 2019**

## Modification History

| Version | Date | Description |
|---|---|---|
| Version 1.0 | 9/4/2018 | Initial Release |
| Version 1.1 | 9/18/2018 | Updates based on testing and Gemalto review |
| Version 1.2 | 10/09/2018 | Updates based on review and Gemalto feedback |
| Version 1.3 | 11/18/2018 | Updates based on additional CAVP testing and review |
| Version 1.4 | 12/20/2018 | Updates based on quality and technical review |
| Version 1.5 | 12/22/2018 | Editorial updates |
| Version 1.6 | 12/26/2018 | Editorial updates |
| Version 1.7 | 12/27/2018 | Editorial updates |
| Version 1.8 | 03/06/2019 | Editorial updates |
| Version 1.9 | 03/08/2019 | Editorial updates |
| Version 1.10 | 03/25/2019 | Editorial updates |
| Version 1.11 | 04/22/2019 | Logo updates |

# Table of contents



# List of Tables



# List of Figures

# 1 Introduction

This document is the non-proprietary security policy for the Gemalto FIPS Object Module, hereafter referred to as the Module.

The Module is a software library providing a C language application program interface (API) for use by other processes that require cryptographic functionality. The Module is classified by FIPS 140-2 as a software module, multichip standalone module embodiment. The physical cryptographic boundary is the general-purpose computing device on which the module is installed. The logical cryptographic boundary of the Module is the fipscanister object module, a single object module file named fipscanister.o. The Module performs no communications other than with the calling application (the process that invokes the Module services).

The FIPS 140-2 security levels for the Module are as follows:

| Security Requirements Section | Level |
|---|---|
| Cryptographic Module Specification | 1 |
| Cryptographic Module Ports and Interfaces | 1 |
| Roles and Services and Authentication | 1 |
| Finite State Machine Model | 1 |
| Physical Security | N/A |
| Operational Environment | 1 |
| Cryptographic Key Management | 1 |
| EMI/EMC | 1 |
| Self-Tests | 1 |
| Design Assurance | 3 |
| Mitigation of Other Attacks | N/A |

**Table 1: Security Level of Security Requirements**

The Module's software version for this validation is 2.1.0.

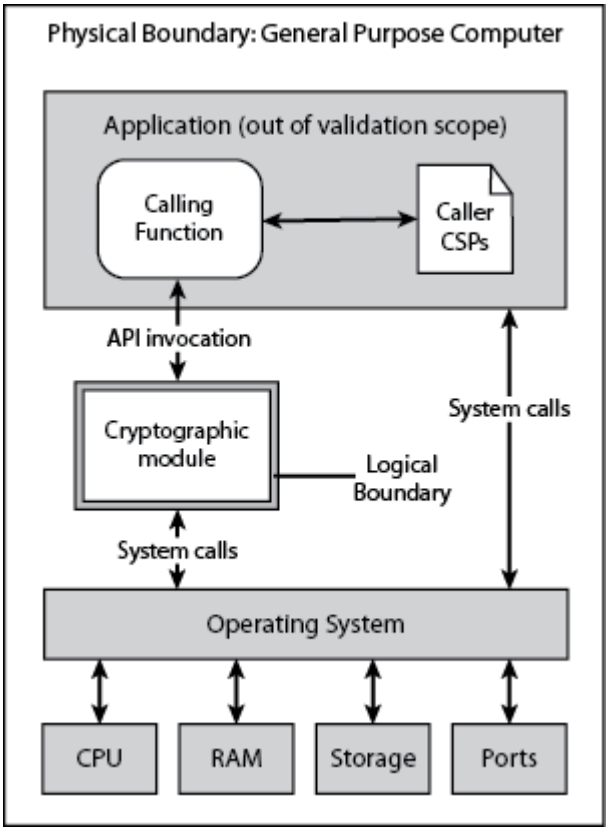


**Figure 1: Module Block Diagram**

## 2 Ports and Interfaces

The physical ports of the Module are the same as the general purpose computing device on which it is executing. The logical interface is a C language API.

| Logical Interface Type | Description |
|---|---|
| Control Input | API entry point and corresponding stack or register parameters |
| Data Input | API entry point data input stack or register parameters |
| Status Output | API entry point return values and status stack or register parameters |
| Data Output | API Entry point data output stack or register parameters |

**Table 2: Logical Interfaces**

As a software module, control of the physical ports is outside module scope. However, when the module is performing self-tests, or is in an error state, all output on the logical data output interface is inhibited. The module is single-threaded and in error scenarios returns only an error value (no data output is returned).

# 3 Modes of Operation

The Module supports FIPS 140-2 Approved, Allowed and Non-Approved algorithms in a single mixed mode of operation.

## 3.1 Approved Mode

The Module supports the following services and algorithms in FIPS Approved Mode:

| Function | Algorithm | Options | Cert# |
|---|---|---|---|
| Random Number Generation; symmetric key generation | [SP 800-90A] DRBG[1] Prediction resistance supported for all variations | **Hash_Based DRBG**: [ Prediction Resistance Tested: Enabled and Not Enabled (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)] **HMAC_Based DRBG**: [ Prediction Resistance Tested: Enabled and Not Enabled (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)] **CTR_DRBG**: [ Prediction Resistance Tested: Enabled and Not Enabled; BlockCipher_Use_df: (AES-128, AES-192, AES-256)] BlockCipher_No_df: (AES-128, AES-192, AES-256)] | 2407 |
| Cryptographic Key Generation (CKG) | [SP 800-133] CKG | Vendor Affirmed | N/A |
| Encryption, Decryption, and CMAC | [SP 800-67] [SP 800-38A] | 3-Key TDES TECB, TCBC, TCFB (CFB1/CFB8/CFB64), TOFB; CMAC generate and verify | 2861 |
| | [FIPS 197] AES | 128/ 192/256 ECB, CBC, OFB, CFB 1, CFB 8, CFB 128, CTR, XTS[2]; CCM; GCM; CMAC generate and verify | 5829 |
| | [SP 800-38B] CMAC [SP 800-38C] CCM [SP 800-38D] GCM [SP 800-38E] XTS | | |
| Message Digests | [FIPS 180-4] | SHA-1, SHA-2 (224, 256, 384, 512) | 4615 |
| Keyed Hash | [FIPS 198] HMAC | SHA-1, SHA-2 (224, 256, 384, 512) | 3840 |
| Digital Signature and asymmetric Key Generation | [FIPS 186-2] RSA | SigGen9.31, SigGenPKCS1.5, SigGenPSS (4096 with all SHA-2 sizes) SigVer9.31, SigVerPKCS1.5, SigVerPSS (1024/1536/2048/3072/4096 with all SHA sizes) | 3078 |
| | [FIPS 186-4] RSA | KeyGen, SigGen9.31, SigGenPKCS1.5, SigGenPSS, SigVer9.31, SigVerPKCS1.5, SigVerPSS(2048/3072 with all SHA-2 sizes) | 3078 |
| | [FIPS 186-4] DSA | Key Pair Gen (2048/3072) PQG Gen, Sig Gen (2048/3072 with all SHA-2 sizes) PQG Ver, Sig Ver (1024/2048/3072 | 1473 |

*[1]For all DRBGs the "supported security strengths" is just the highest supported security strength per [SP800-90A] and [SP800-57].*

*[2] Only key lengths 128 and 256 options can be CAVP tested for this mode.*

| | | | |
|---|---|---|---|
| | | with all SHA sizes) | |
| | [FIPS 186-4] ECDSA | Key Pair Gen: CURVES<br>P-224<br>P-256<br>P-384<br>P-521<br>(ExtraRandomBits<br>TestingCandidates )<br>PKV: CURVES (ALL-P)<br>SigGen: CURVES<br>P-224: (SHA-224, 256, 384, 512)<br>P-256: (SHA-224, 256, 384, 512)<br>P-384: (SHA-224, 256, 384, 512)<br>P-521: (SHA-224,256, 384, 512)<br>SigVer: CURVES<br>P-192: (SHA-1, 224, 256, 384, 512)<br>P-224: (SHA-1, 224, 256, 384, 512)<br>P-256: (SHA-1, 224, 256, 384, 512)<br>P-384: (SHA-1, 224, 256, 384, 512)<br>P-521: (SHA-1, 224, 256, 384, 512) | 1558 |
| ECC CDH CVL (KAS) | [SP 800-56A] (§5.7.1.2) | All NIST defined P curves except size 192 | 2098 |

**Table 3: FIPS Approved Cryptographic Functions**

## 3.2 Non-Approved But Allowed Services

The Module supports the following non-approved but allowed services.

| Category | Algorithm | Description |
|---|---|---|
| Key Agreement | DH | Key agreement is a service provided by the module to establish session keys for secure communication with another module using the Diffie-Hellman algorithm. |
| Key Agreement | EC DH | Key agreement is a service provided by the module to establish session keys for secure communication with another module using the EC Diffie-Hellman algorithm. |
| Key Encryption/Decryption | RSA | RSA may be used to perform key establishment with another module by securely exchanging symmetric encryption keys with another module. |
| Entropy Source | NDRNG | The module obtains the entropy data from NDRNG to seed the DRBG |

**Table 4: Non-FIPS Approved but Allowed Cryptographic Functions**

The module supports the following non-FIPS 140-2 approved but allowed algorithms:

- RSA (key wrapping; key establishment methodology provides between 112 and 256 bits of encryption strength)
- Diffie-Hellman (key agreement; key establishment methodology provides between 112 and 219 bits of encryption strength; non-compliant less than 112 bits of encryption strength)
- EC Diffie-Hellman (CVL Cert. #2098, key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)

## 3.3 Non-Approved Services

The Module implements the following services which are Non-Approved per the FIPS 140-2 IG and SP 800-131Ar1 transition:

| Function | Algorithm | Options |
|---|---|---|
| Digital Signature and Asymmetric Key Generation | [FIPS 186-2] RSA | GenKey9.31, SigGen9.31, SigGenPKCS1.5, SigGenPSS (2048/3072/4096 with SHA1) |
| | [FIPS 186-2] DSA | PQG Gen, Key Pair Gen, Sig Gen (2048/3072 with SHA1) |
| | [FIPS 186-4] DSA | PQG Gen, Key Pair Gen, Sig Gen (2048/3072 with SHA-1) |
| | [FIPS186-2] ECDSA | PKG: CURVES (P-192) SIG (gen): CURVES (P-192 P-224 P-256 P--384 P-521) with SHA-1 |
| | [FIPS 186-4] ECDSA | PKG: CURVES (P-192)<br>SigGen: CURVES (P-192: (SHA-1, 224, 256, 384, 512) P224:(SHA-1) P-256:(SHA-1) P-384: (SHA-1) P-521:(SHA-1) |
| ECC CDH (KAS) | [SP800-56A] (§5.7.1.2) | P curves sizes 163 and 192 |

**Table 5: Non-FIPS Approved Cryptographic Functions**

These algorithms shall not be used when operating in the FIPS Approved mode of operation. The use of these non-conformant algorithms with any of the services in Table 9 will result in the module being placed in the non-approved mode.

## 3.4 Critical Security Parameters and Public Keys

All CSPs used by the Module are described in this section. All access to these CSPs by Module services are described in Section 4. The CSP names are generic, corresponding to API parameter data structures.

| CSP Name | Description |
|---|---|
| RSA SGK | RSA (2048 to 16384 bits) signature generation key |
| RSA KDK | RSA (2048 to 16384 bits) key decryption (private key transport) key |
| DSA SGK | [FIPS 186-4] DSA (2048/3072) signature generation key |
| DH Private | Diffie-Hellman ≥ 2048 private key agreement key |
| ECDSA SGK | ECDSA (All NIST defined P curves except size 192) signature generation key |
| EC DH Private | EC DH (All NIST defined P curves except size 192) private key agreement key. |
| AES EDK | AES (128/192/256) encrypt / decrypt key |
| AES CMAC | AES (128/192/256) CMAC generate / verify key |
| AES GCM | AES (128/192/256) encrypt / decrypt / generate / verify key |
| AES XTS | AES (256/512[3]) XTS encrypt / decrypt key |
| Triple-DES EDK | Triple-DES (3-Key) encrypt / decrypt key |
| Triple-DES CMAC | Triple-DES (3-Key) CMAC generate / verify key |
| HMAC Key | Keyed hash key (160/224/256/384/512) |
| Hash_DRBG CSPs | V (440/888 bits) and C (440/888 bits), seed (128/192/256 bits) and entropy input (128/192/256 bits) |
| HMAC_DRBG CSPs | V (160/224/256/384/512 bits) and Key (160/224/256/384/512 bits), seed (128/192/256 bits) and entropy input (128/192/256 bits) |
| CTR_DRBG CSPs | V (128 bits) and Key (AES 128/192/256 bits), seed (128/192/256 bits) and entropy input (128/192/256 bits) |

**Table 6: Critical Security Parameters**

[3] *Only key lengths 128 and 256 options can be CAVP tested for this mode.*

The module does not output intermediate key generation values.

| CSP Name | Description |
|---|---|
| RSA SVK | RSA (1024 to 16384 bits) signature verification public key |
| RSA KEK | RSA (2048 to 16384 bits) key encryption (public key transport) key |
| DSA SVK | [FIPS 186-4] DSA (2048/3072) signature verification key |
| ECDSA SVK | ECDSA (All NIST defined P curves) signature verification key |
| DH Public | Diffie-Hellman public key agreement key |
| EC DH Public | EC DH (All NIST defined P curves) public key agreement key |

**Table 7: Public Keys**

**For all CSPs and Public Keys:**

**Storage:** RAM, associated to entities by memory location. The Module stores DRBG state values for the lifetime of the DRBG instance. The module uses CSPs passed in by the calling application on the stack or registers. The Module does not store any CSP persistently (beyond the lifetime of an API call), with the exception of DRBG state values used for the Modules' default key generation service.

**Generation:** The Module implements SP 800-90A compliant DRBG services for creation of symmetric keys, and for generation of DSA, elliptic curve, and RSA keys as shown in Table 3. The calling application is responsible for storage of generated keys returned by the module.

FIPS 140-2 IG 7.14 1 (b) is applicable to this module. The module, a software library based in user space will request entropy from the Operational Environment as appropriate to the security strength and seeding configuration for the DRBG that is using it.

For operation in the Approved mode, Module users (the calling applications) shall use entropy sources that contain at least 112 bits of entropy. To ensure full DRBG strength, the entropy sources must meet or exceed the security strengths shown in the table below.

| DRBG Type | Underlying Algorithm | Minimum Seed Entropy |
|---|---|---|
| Hash_DRBG or HMAC_DRBG | SHA-1 | 128 |
| | SHA-224 | 192 |
| | SHA-256 | 256 |
| | SHA-384 | 256 |
| | SHA-512 | 256 |
| CTR_DRBG | AES-128 | 128 |
| | AES-192 | 192 |
| | AES-256 | 256 |

**Table 8 - DRBG Entropy Requirements**

**Entry:** All CSPs enter the Module's logical boundary in plaintext as API parameters, associated by memory location. However, none cross the physical boundary.

**Output:** The Module does not output CSPs, other than as explicit results of key generation services. However, none cross the physical boundary.

**Destruction:** Zeroization of sensitive data is performed automatically by API function calls for temporarily stored CSPs. In addition, the module provides functions to explicitly destroy CSPs related to random number generation services. The calling application is responsible for parameters passed in and out of the module.

Private and secret keys as well as seeds are provided to the Module by the calling application and are destroyed when released by the appropriate API function calls. Keys residing in internally allocated data structures (during the lifetime of an API call) can only be accessed using the Module defined API. The operating system protects memory and process space from unauthorized access. Only the calling application that creates or imports keys can use or export such keys. All API functions are executed by the invoking calling application in a non-overlapping sequence such that no two API functions will execute concurrently. An authorized application as user (Crypto Officer and User) has access to all key data generated during the operation of the Module.

**Use:** In the event Module power is lost and restored the calling application must ensure that any AES-GCM keys used for encryption or decryption are redistributed.

In the case of AES-GCM, the IV generation method is user selectable and the value can be computed in more than one manner as follows:

- TLS 1.2: The module's AES-GCM implementation conforms to IG A.5, scenario #1, following RFC 5288 for TLS. The counter portion of the IV is set by the module within its cryptographic boundary. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, to encounter this condition will trigger a handshake to establish a new encryption key in accordance with RFC 5246.

- Non-TLS 1.2: The module's AES-GCM implementation conforms to IG A.5, scenario #3, when operating in a FIPS approved mode of operation, AES GCM, IVs are constructed both internally and deterministically and are a minimum of 96-bits in length as specified in SP 800-38D, Section 8.2.1.

The selection of the IV construction method is the responsibility of the user of this cryptographic module

The calling application shall ensure that the same Triple-DES key is not used to encrypt more than 216 64-bit blocks of data. In approved mode, users of the module must not utilize GCM with an externally generated IV.

Per the requirements of SP 800-38E, AES-XTS mode shall be used for storage purposes only.

# 4 Roles, Authentication and Services

The Module implements the required User and Crypto Officer roles. Only one role may be active at a time and the Module does not support user authentication. The User and CO roles are implicitly assumed by the entity accessing services implemented by the module. A user is considered the owner of the thread that instantiates the module and, therefore, only one concurrent user is allowed.

Both roles have access to all of the services provided by the Module.

- User Role (User): Loading the Module and calling any of the API functions.
- Crypto Officer Role (CO): Installation of the Module on the host general purpose computing device and calling of any API functions.

All services implemented by the Module are listed below, along with a description of service CSP access. The access types are determined as follows:

- Generate (G): Generates the Critical Security Parameter (CSP) using an approved Random Bit Generator
- Read (R): Export the CSP (to an assigned location in memory)
- Write (W): Enter/establish and store a CSP (to an assigned location in memory)
- Destroy (D): Overwrite the CSP
- Execute (E): Employ the CSP
- None: No access to CSP's

| Service | Role | Description | Access Type |
|---|---|---|---|
| Initialize | User, CO | Module initialization. Does not access CSPs. | None |
| Self-test | User, CO | Perform self-tests (FIPS_selftest). | None |
| Show status | User, CO | Functions that provide module status information:<br>• Version (as unsigned long or const char *)<br>• FIPS Mode (Boolean value "1") | None |
| Zeroize | User, CO | Functions that destroy CSPs:<br>fips_drbg_uninstantiate<br>DRBG CSPs (Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs)<br><br>All other services automatically overwrite CSPs stored in allocated memory. Stack cleanup is the responsibility of the calling application. | R/D |
| Random number generation | User, CO | Used for random number and symmetric key generation.<br>• Seed or reseed a DRBG instance<br>• Determine security strength of a DRBG instance<br>• Obtain random data<br><br>Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs. | R/W |
| Asymmetric key generation | User, CO | Used to generate DSA, ECDSA and RSA keys:<br>RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK | R/E/G/W |
| Symmetric encrypt/decrypt | User, CO | Used to encrypt or decrypt data.<br><br>AES EDK, TRIPLE-DES EDK, AES GCM, AES XTS (passed in by the calling process). | R/E |
| Symmetric digest | User, CO | Used to generate or verify data integrity with CMAC.<br><br>AES CMAC, TRIPLE-DES CMAC (passed in by the calling process) | R/E/G/W |
| Message digest | User, CO | Used to generate a SHA-1 or SHA-2 message digest. | None |

| | | | |
|---|---|---|---|
| Keyed Hash | User, CO | Used to generate or verify data integrity with HMAC.<br><br>HMAC Key (passed in by the calling process). | R/E/G/W |
| Key transport[4] | User, CO | Used to encrypt or decrypt a key value on behalf of the calling process (does not establish keys into the module).<br><br>RSA KDK, RSA KEK (passed in by the calling process). | R/E |
| Key agreement | User, CO | Used to perform key agreement primitives on behalf of the calling process (does not establish keys into the module).<br><br>Diffie-Hellman/EC Diffie-Hellman Private, Diffie-Hellman/EC Diffie-Hellman Public (passed in by the calling process) | R/E/G/W |
| Digital signature | User, CO | Used to generate or verify RSA, DSA or ECDSA digital signatures.<br><br>RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK (passed in by the calling process). | R/E/G/W |
| Utility | User, CO | Miscellaneous helper functions. | None |

**Table 9: Services and CSP Access**

[4] *"Key transport" can refer to a) moving keys in and out of the module, or b) the use of keys by an external application. The latter definition is the one that applies to the OpenSSL FIPS Object Module*

# 5 Self-Tests

The Module performs the self-tests listed below on invocation of Initialize or Self-test.

| Algorithm | Type | Test Attributes |
|---|---|---|
| Software integrity | KAT | HMAC-SHA-1 |
| HMAC | KAT | One KAT per SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 Per IG 9.3, this testing covers SHA POST requirements. |
| AES | KAT | Separate encrypt and decrypt, ECB mode, 128-bit key length |
| AES CCM | KAT | Separate encrypt and decrypt, 192 key length |
| AES GCM | KAT | Separate encrypt and decrypt, 256 key length |
| XTS-AES | KAT | 128, 256-bit key sizes to support either the 256-bit key size (for XTS-AES-128) or the 512-bit key size (for XTS-AES-256) |
| AES CMAC | KAT | Sign and verify CBC mode, 128, 192, 256 key lengths |
| Triple-DES | KAT | Separate encrypt and decrypt, ECB mode, 3-Key |
| Triple-DES CMAC | KAT | CMAC generate and verify, CBC mode, 3-Key |
| RSA | KAT | Sign and verify using 2048 bit key, SHA-256, PKCS#1 |
| DSA | PCT | Sign and verify using 2048 bit key, SHA-384 |
| DRBG | KAT | CTR_DRBG: AES, 256-bit with and without derivation function<br>HASH_DRBG: SHA256<br>HMAC_DRBG: SHA256 |
| ECDSA | PCT | Keygen, sign, verify using P-224 and SHA-512 |
| ECC CDH | KAT | Shared secret calculation per SP 800-56A §5.7.1.2, IG 9.6 |

Table 10: Power-On Self-Tests

Per IG 9.10, the Module implements a default entry point and automatically runs the FIPS self-tests upon startup.

The module has a function called FIPS_module_mode_set() within the init code that is automatically set to enable “FIPS Mode” by default. When the Gemalto FIPS Object Module is initialized, it will always run its power-on self-tests meeting the IG 9.10 requirement.

The Module also implements the following conditional tests:

| Algorithm | Test |
|---|---|
| DRBG | [SP 800-90A] Section 11.3 Instantiate, Generate, Reseed health tests for Hash_DRBG/HMAC_DRBG and CTR_DRBG |
| DRBG | FIPS 140-2 continuous test for stuck fault |
| NDRNG | FIPS 140-2 Continuous test for NDRNG |
| DSA | Pairwise consistency test on each generation of a key pair |
| ECDSA | Pairwise consistency test on each generation of a key pair |
| RSA | Pairwise consistency test on each generation of a key pair |

**Table 11: Conditional Self-Tests**

In the event of a DRBG self-test failure the calling application must un-instantiate and re-instantiate the DRBG per the requirements of [SP 800-90A]; this is not something the Module can do itself.

Pairwise consistency tests are performed for both possible modes of use, e.g. Sign/Verify and Encrypt/Decrypt

# 6 Operational Environment

The operational environments tested are considered modifiable operational environments under FIPS 140-2. The tested operating systems segregate user processes into separate process spaces. Each process space is logically separated from all other processes by the operating system software and hardware. The Module functions entirely within the process space of the calling application, and implicitly satisfies the FIPS 140-2 requirement for a single user mode of operation.

## 6.1 Tested Configurations

The module was tested in the following configurations.

| # | Operational Environment | Processor | Platform |
|---|---|---|---|
| 1 | Android 4.4[5] | Qualcomm APQ8064 (ARMv7) | LG Nexus 4 |
| 2 | Android 5.1 | Samsung Exynos (ARMv8) | Samsung Galaxy S6 |
| 3 | Android 6.0 | Qualcomm Snapdragon 810(ARMv8) | Huawei Nexus 6P |
| 4 | Android 7.1 | Qualcomm Snapdragon 821 (ARMv8) | Google Pixel XL |
| 5 | Android 8.1 | Qualcomm Snapdragon 835 (ARMv8) | Google Pixel 2 |
| 6 | iOS 9.3 | Apple A8 (ARMv8) | iPhone 6 |
| 7 | iOS 10.3 | Apple A6 (ARMv7) | iPhone 5 |
| 8 | iOS 11.4 | Apple A11 (ARMv8) | iPhone 8 |

**Table 12: Tested Configurations**

## 6.2 Vendor Affirmed Configurations

The following platforms have not been tested as part of the FIPS 140-2 level 1 certification however Gemalto "vendor affirms" that these platforms are equivalent to the tested and validated platforms. Additionally, Gemalto affirms that the module will function the same way and provide the same security services on any of the operating systems listed below.

- Android 5.0
- Android 7.0
- Android 8.0
- Android 9
- iOS 10.0 - 10.2
- iOS 11.0 – 11.3
- iOS 12

As per FIPS 140-2 Implementation Guidance G.5, the CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

[5] *Tested with and without PAA*

# 7 Mitigations of Other Attacks

The module is not designed to mitigate against attacks which are outside of the scope of FIPS 140-2.

# 8 References

The FIPS 140-2 standard, and information on the CMVP, can be found at https://csrc.nist.gov/projects/cryptographic-module-validation-program . More information describing the module can be found on the Gemalto web site at www.gemalto.com.This Security Policy contains non-proprietary information. All other documentation submitted for FIPS 140-2 conformance testing and validation is "Gemalto - Proprietary" and is releasable only under appropriate non-disclosure agreements.

| References | Full Specification Name |
|---|---|
| [FIPS 140-2] | Security Requirements for Cryptographic modules, May 25, 2001 |
| [FIPS 180-4] | Secure Hash Standard |
| [FIPS 186-4] | Digital Signature Standard |
| [FIPS 197] | Advanced Encryption Standard |
| [FIPS 198-1] | The Keyed Hash Message Authentication Code (HMAC) |
| [SP 800-38A] | Recommendation for Block Cipher Modes of Operation: Methods and Techniques |
| [SP 800-38B] | Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication |
| [SP 800-38C] | Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality |
| [SP 800-38D] | Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC |
| [SP 800-56A] | Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography |
| [SP 800-67R1] | Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher |
| [SP 800-89] | Recommendation for Obtaining Assurances for Digital Signature Applications |
| [SP 800-90A] | Recommendation for Random Number Generation Using Deterministic Random Bit Generators |
| [SP 800-131A] | Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths |

**Table 14: References**