# Symantec Corporation

## Symantec PGP Cryptographic Engine FIPS 140-2 Non-proprietary

## Security Policy

Document Version 1.0.4

Revision Date 05/01/2015

# Table of Contents

# 1 Introduction

The PGP Cryptographic Engine (SW Version 4.3) (hereafter referred to as the "cryptographic module" or the "module") is a software only cryptographic module validated to the standards set forth by the *FIPS PUB 140-2 Security Requirements for Cryptographic Modules* document published by the National Institute of Standards and Technology (NIST). The module is intended to meet the security requirements of FIPS 140-2 Level 1 overall.

This document, the *Symantec PGP Cryptographic Engine FIPS 140-2 Non-proprietary Security Policy*, also referred to as the *Security Policy*, specifies the security rules under which the module must operate.

# 2 Module Specifications

The PGP Cryptographic Engine (SW Version 4.3) is a software-only cryptographic module embodied as a shared library binary that executes on general-purpose computer systems and is available on a number of operating systems. The specific operating system and version to be validated is specified in the "Operational Environment" section of this document.

The PGP Cryptographic Engine cryptographic module is accessible to client applications through an application-programming interface (API).

The module provides a FIPS mode of operation, which is described in the "Approved Mode of Operation" section of this document.

For the purposes of FIPS 140-2, the PGP Cryptographic Engine is classified as a multi-chip standalone module.

## 2.1 Supported Algorithms

The PGP Cryptographic Engine implements the following Approved algorithms in the FIPS Approved mode of operation.

| Type | Algorithm | Certificate Number |
|---|---|---|
| Symmetric Key | Triple-DES (3-Key) TECB, TCBC, TCFB | FIPS 46-3 (cert # 1675, 1676, 1683, 1684, 1711, 1712, 1713, 1714, 1715, 1716) |
| | AES (128,192,256)<br>ECB, CBC and CFB128 | FIPS 197 (cert # 2766, 2786, 2799, 2805, 2866, 2867, 2868, 2869, 2870, 2871) |
| Message Digest | SHA-1, 256, 384, 512 | FIPS 180-3 (cert # 2342, 2343, 2351, 2353, 2408, 2409, 2410, 2411, 2412, 2413) |
| Message Authentication | HMAC SHA-1, 256, 384, 512 | FIPS 198 (cert # 1746, 1747, 1755, 1756, 1805, 1806, 1807, 1808, 1809, 1810) |
| Digital Signature | RSA (2048, 3072) | FIPS 186-4 (cert # 1459, 1465, 1466, 1468, 1503, 1504, 1505, 1508, 1509, 1510) |
| | DSA (L = 2048, N = 224; L = 2048, N = 256; L = 3072, N = 256 | FIPS 186-4 (cert # 846, 847, 848, 849, 859, 860, 861, 862, 863, 864) |
| | ECDSA (P-256, P-384) | FIPS 186-4 (cert # 487, 488, 489, 490, 509, 510, 511, 512, 513, 514) |
| Key Establishment | CVL: ECC CDH Primitive (P-256, P-384) | SP 800-56A (cert # 240, 241, 248, 249, 302, 303, 304, 305, 306, 307) |
| DRBG | AES256_CTR with derivation function | SP 800-90A (cert # 473, 474, 478, 479, 510, 511, 512, 513, 514, 515) |

**Table 1 - Algorithms supported by the PGP Cryptographic Engine**

The PGP Cryptographic Engine also implements the following non-Approved but allowed in the FIPS Approved mode of operation Algorithms:

- NDRNG
- RSA (key wrapping; key establishment methodology provides 112 bits of encryption strength)

## 2.2 Non-Approved Algorithms

NOTICE: The PGP Cryptographic Engine module provides the following non-FIPS approved algorithms only in non-FIPS mode of operation. The services listed in Table 2 are available to the calling application. However the use of any such service is an explicit violation of this Security Policy and is explicitly disallowed by this Security Policy.

| **Non-Approved Service** | **Non-Approved Algorithms** |
|---|---|
| Non-Approved Encrypt/Decrypt | AES EME2 (non-compliant), AES PlumbCFB (non-compliant), AESMixCBC (non-compliant), RC2, ARC4, IDEA, CAST5, TwoFish, BlowFish, ElGamal |
| Non-Approved Signature generation and verification | RSA and DSA with modulus size 1024(non-compliant), RSA SHA-1(non-compliant), DSA SHA-1(non-compliant), ECDSA SHA-1(non-compliant) |
| Non-Approved Hashing | MD-5, RIPEMD160, MD-2, KECCEK |
| Non-Approved Key Derivation | PBKDF2(non-compliant), KBKDF(non-compliant) OpenPGP S2K Iterated salted |

**Table 2 – Non-Approved Algorithms supported by the PGP Cryptographic Engine**

## 2.3 Cryptographic Boundary

The physical cryptographic boundary is defined as the computer's case that the PGP Cryptographic Engine is installed in and includes all the accompanying hardware. The module’s logical cryptographic boundary is defined to be a subset of the PGP Cryptographic Engine binary software library as defined by the "Roles and Services" section of this document.

An operator is accessing (or using) the module whenever one of the library calls is executed through the API and thus the module logical interfaces are provided by the API calls.

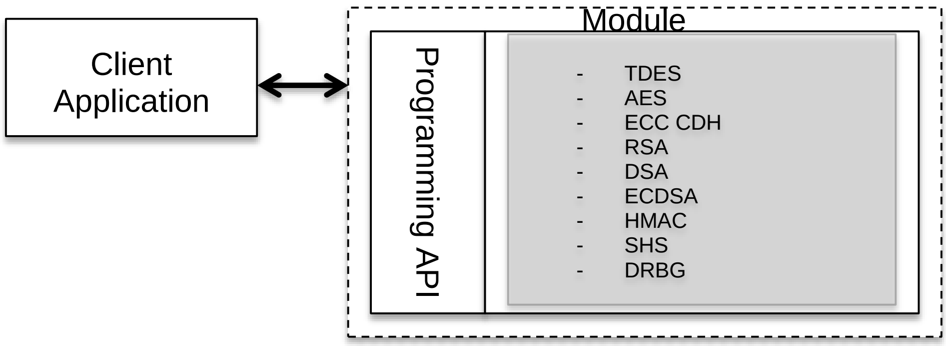


**Figure 1 - Module Cryptographic Boundary**

Note that the dashed line represents the PGP Cryptographic Engine crypto boundary.

## 2.4 Ports and Interfaces

The module restricts all access to its Critical Security Parameters (CSPs) through the API calls as enumerated in the "Roles and Services" section of this document. This API acts as the logical interface to the module.

Although the computer's physical ports such as keyboards, mouse, displays, hard disks, smart card interfaces, etc. provide a means to access the cryptographic module, the actual interface is via the API itself.

For the purpose of FIPS 140-2, the logical interfaces can be modeled as described in the following table.

| | |
|---|---|
| Data Input | Parameters passed to the module via API calls. |
| Data Output | Data returned by the module via API calls. |
| Control Input | Control Input – API function calls. |
| Status Output | Error and status codes returned by API calls. |

**Table 3 - PGP Cryptographic Engine Logical Ports**

Input and output data can consist of plain-text, cipher-text, and cryptographic keys as well as other parameters. The module does not support a cryptographic bypass mode.

All data output is inhibited during an error state. Data output is also inhibited during the self-test process.

## 2.5 Security Level

The PGP Cryptographic Engine Module meets the overall security requirements of FIPS 140-2 Level 1.

| **Security Requirements Area** | **Level** |
|---|---|
| Cryptographic Module Specification | 1 |
| Cryptographic Module Ports and Interfaces | 1 |
| Roles, Services and Authentication | 1 |
| Finite State 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 4 - Module Security Level Specification**

## 2.6 Operational Environment

The following Operating Systems were used to operationally test and validate the PGP Cryptographic Engine to the requirements of FIPS-140-2.

- Apple Mac OS X 10.7 with AES-NI
- Apple Mac OS X 10.7 without AES-NI
- Microsoft Windows 7 32-bit with AES-NI
- Microsoft Windows 7 32-bit without AES-NI
- Microsoft Windows 7 64-bit with AES-NI
- Microsoft Windows 7 64-bit without AES-NI Red Hat Enterprise Linux (RHEL) 6.2 32-bit with AES-NI
- Red Hat Enterprise Linux (RHEL) 6.2 32-bit without AES-NI
- Red Hat Enterprise Linux (RHEL) 6.2 64-bit with AES-NI
- Red Hat Enterprise Linux (RHEL) 6.2 64-bit without AES-NI

As per FIPS Implementation Guidance the PGP Cryptographic Engine module will remain compliant with the requirements of FIPS 140-2 when operating on the following compatible Operating Systems:

- Microsoft Windows 8 32-bit
- Microsoft Windows 8 64-bit
- Apple Mac OS X 10.8
- Apple Mac OS X 10.9
- Virtualized vSphere 5.1 / ESXi 5.1 hypervisor w/ Windows 8.1 update 1 x64 with AES-NI
- Virtualized vSphere 5.1 / ESXi 5.1 hypervisor w/ Windows Server 2012 R2 x64 with AES-NI

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.

## 2.7 Approved Mode of Operation

The PGP Cryptographic Engine provides a FIPS 140-2 compliant mode of operation. It is possible to use various non-approved algorithms (see section 2.2) in the non-FIPS mode of operation; in this case the FIPS 140-2 self-tests are still required to be run and pass validation prior to using the non-approved algorithms.

The client application can, at any time, verify the status by performing the `PGPceGetSDKErrorState()` API call.

An application can also check the module error state and run all or any specific self-test through making the proper API calls.

# 3 Security Rules

Following is a list of security requirements that specify the Approved mode of operation and must be adhered to when complying with FIPS 140-2.

1. PGP Cryptographic Engine must be used as described in this document.
2. Installation of the module is the responsibility of the Crypto Officer.
3. The cryptographic module provides a FIPS 140-2 compliant mode of operation. Before the module can be used, it must be initialized as described in the "Approved Mode of Operation" section of this document.
4. The cryptographic module conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B ( i.e., for Home use) which vacuously satisfies Class A.
5. Only FIPS approved or allowed cryptographic algorithms as enumerated in the "Supported Algorithms" section of this document are to be used.
6. The cryptographic module inhibits data output during self-tests and error states. The data output interface is logically disconnected from the processes performing zeroization.
7. The zeroization process can be achieved using the appropriate API function: `PGPceFreeSymmetricCipherContext`, `PGPceFreeCBCContext`, `PGPceFreeCFBContext`, `PGPceFreeHashContext`, `PGPceFreeHMACContext` or `PGPceWipeSymmetricCipher`, `PGPceFreePKContext`.
8. PGP Cryptographic Engine is designed to meet FIPS 140-2, security Level 1, therefore the module does not provide authentication mechanisms.

# 4 Roles and Services

The module operator is defined as any client application that is linked to the PGP Cryptographic Engine shared library (PGPce.dll on the Windows platforms, libPGPce.dylib on OS X platforms, and libPGPCE.so.4.3.0 on Linux platforms)

The cryptographic module supports two roles (described below). An operator accesses both roles while using the module and the means of access is the same for both roles. A role is implicitly assumed based on the services that are accessed.

The Crypto Officer (CO) is any entity that can install the module library onto the computer system, configure the operating system, and validate the compliance of the module. The Crypto Officer's role is implicitly selected when installing the module, or configuring the operating system.

Installation is accomplished by running an installation program. The Crypto Officer must have permission to write the library constituting the PGP Cryptographic Engine into an operating system directory; typically, this requires administrator access to the operating system.

The roles are defined as the following:

- User: Shall be allowed to perform all services provided by the module.
- Crypto Officer: Shall be allowed to perform all services provided by the module and additionally is responsible for the installation of the module.

# Access Control Policy

In the PGP Cryptographic Engine, access to critical security parameters is controlled. A module User or Crypto Officer can only read, modify, or otherwise access the security relevant data through the cryptographic module services provided by the module API interface. This section details the Critical Security Parameters (CSPs) in the cryptographic module that a User or Crypto Officer can access, how the CSPs can be accessed in the cryptographic module, and which services are used for access to the data item.

## 4.1 Critical Security Parameters

The Critical Security Parameters (CSPs) used by the PGP Cryptographic Engine module are protected from unauthorized disclosure, modification, and substitution.

Definition of CSPs:

- TDES Key - used to TDES encrypt/decrypt data.
- AES Key - used to AES encrypt/decrypt data.
- RSA Key Pairs - used for signing and verification
- RSA Key Pairs – used for encrypt/decrypt (key wrapping only)
- DSA Key Pairs - used for signing and verification
- ECDSA Key Pairs - used for signing and verification
- ECC CDH Key Pairs – used for key pair establishment
- DRBG entropy and seed – used for random bit generation
- HMAC Key - used for message authentication of data.

## 4.2 Accesses

The types of access to CSPs in the PGP Cryptographic Engine module are listed in the following table.

| Access | Description |
|---|---|
| create | The item is created. |
| destroy | The item is destroyed, in other words the data is cleared (actively overwritten) from any memory in the cryptographic module and then that memory is released. |
| read | The item is accessed for reading and use. |
| write | The item is modified or changed. |

**Table 5 - CSP Access Types**

## 4.3 Service to CSP Access Relationship

The following table shows which CSPs are accessed by each service, the role(s) the operator must be in for access, and how the CSP is accessed on behalf of the operator when the service is performed.

Several services provided by the PGP Cryptographic Engine module do not access any CSPs and are included here for completeness.

| Service | CO | User | CSP | create | destroy | read | write |
|---|---|---|---|---|---|---|---|
| Encrypt/decrypt data with symmetric key | X | X | TDES Encrypt Key<br>AES Encrypt Key | ● | ● | ● | ● |
| Signature generation and verification | X | X | RSA/DSA/ECDSA key pairs | ● | ● | ● | ● |
| Hash data | X | X | N/A | | | | |
| Compute HMAC on data | X | X | HMAC Key | ● | ● | ● | ● |
| ECC CDH key pair establishment | X | X | ECC CDH key pairs | ● | ● | ● | ● |
| Data storage management | X | X | N/A | | | | |
| Show status | X | X | N/A | | | | |
| Run self-tests | X | X | N/A | | | | |
| Zeroize | X | X | TDES Key<br>AES Key<br>RSA Key Pairs<br>DSA Key Pairs<br>ECDSA Key Pairs<br>ECC CDH Key Pairs<br>DRBG entropy and seed<br>HMAC Key | | ● | | |

**Table 6 - Module Services vs Role Access**

# 5 Physical Security Policy

The PGP Cryptographic Engine is implemented as a software module, and as such the physical security section of FIPS 140-2 is not applicable.

| Physical Security Mechanisms | Recommended Frequency of Inspection/Test | Inspection/Test Guidance Details |
|---|---|---|
| N/A | N/A | N/A |

**Table 7: Inspection/Testing of Physical Security Mechanisms**

# 6 Self-Tests

The PGP Cryptographic Engine provides for two forms of self-tests: power-on, and on-demand.

Software integrity test is performed using ECDSA P-384 with SHA-256 signature verification.

The FIPS integrity check and self-test are a mandatory operation and run automatically without operator intervention. The results of the integrity check and self-tests are reported by `PGPceGetSDKErrorState()`.

All data output is prohibited during the self-test process.

If any of these test fail, the module will enter an error state, which can only be cleared by powering down the module. Once in an error state, all further cryptographic operations and data output is disabled.

A client application can also ascertain module at anytime by using the `PGPceGetSDKErrorState()` function. Possible error codes returned by the self-test routines include:

- `kPGPError_NoErr` – self-test was successful.
- `kPGPError_SelfTestFailed` – self-test Failed.

## 6.1 Power-Up Tests

The following self-tests will run and in the following order until all the tests have been completed successfully or until one of the tests fail.

| Algorithm | Test Attributes |
|---|---|
| Software Integrity Test | ECDSA signature verification |
| TDES | Encrypt, ECB mode, 3 key KAT[1] |
| TDES | Decrypt, ECB mode, 3 key KAT |
| DSA | Sign, Verify using 2048-bit with SHA-256 KAT |
| AES | Encrypt, CBC mode, 128-bit, 192-bit, 256-bit KAT |
| AES | Decrypt, CBC mode, 128-bit, 192-bit, 256-bit KAT |
| RSA | Sign, Verify using 2048-bit with SHA-256 KAT |
| RSA | 2048-bit Encrypt KAT |
| RSA | 2048-bit Decrypt KAT |
| SHA | SHA-1, 256, 384, 512 from FIPS 180-4 KAT |
| HMAC | HMAC SHA-1, 256, 384, 512 KAT |
| ECDSA | Sign, Verify P-256 with SHA-256 KAT |
| DRBG | CTR_DRBG: AES, 256-bit KAT |
| ECC CDH | ECC CDH P-256 Primitive “Z” computation KAT |

**Table 8 - Power On Self Tests**

[1] Known Answer Test

## 6.2 Conditional self-tests

| Algorithm | Test Attributes |
|---|---|
| ECDSA | Sign, Verify (using SHA-256)<br>Pair-wise consistency |
| RSA | Sign, Verify (using SHA-256)<br>Pair-wise consistency |
| RSA | Encrypt, Decrypt<br>Pair-wise consistency |
| DSA | Sign, Verify (using SHA-256)<br>Pair-wise consistency |
| NDRNG | Continuous Random Number Generation test |
| DRBG | Continuous Random Number Generation test |

**Table 9 - Conditional self-tests**

## 6.3 On-Demand Tests

Power-up tests can be initiated on-demand by power cycling the module. The client application can optionally initiate a specific test or all tests on demand by using the `PGPceRunSelfTest()` or `PGPceRunAllSelfTests()` functions respectively. Note that if the on-demand tests fail, the module will enter an error state in a manner identical to the power-on self-tests.

# 7 Mitigation of Other Attacks

The Mitigation of Other attacks security section of FIPS 140-2 is not applicable to the PGP Cryptographic Engine module. The module is not designed to mitigate against attacks outside the scope of FIPS 140-2.

| Other Attacks | Mitigation Mechanism | Specific Limitations |
|---|---|---|
| N/A | N/A | N/A |

**Table 10 – Mitigation of Other Attacks**