



FIPS 140-2 Non-Proprietary Security Policy

McAfee ePO Cryptographic Module (Version 1.0)

Document Version 1.4

July 19, 2011

Prepared For:



McAfee, Inc.

2821 Mission College Blvd

Santa Clara, CA 95054

www.mcafee.com

Prepared By:



Apex Assurance Group, LLC

530 Lytton Avenue, Ste. 200

Palo Alto, CA 94301

www.apexassurance.com

Abstract

This document provides a non-proprietary FIPS 140-2 Security Policy for the ePO Cryptographic Module (Version 1.0).

Table of Contents

1	Introduction	5
1.1	About FIPS 140.....	5
1.2	About this Document.....	5
1.3	External Resources.....	5
1.4	Notices.....	5
1.5	Acronyms.....	6
2	McAfee ePO Cryptographic Module (Version 1.0).....	7
2.1	Product Overview	7
2.2	Cryptographic Module Specification.....	7
2.3	Validation Level Detail.....	8
2.4	Cryptographic Algorithms.....	8
2.4.1	Algorithm Implementation Certificates	8
2.4.2	Non-Approved Algorithms	9
2.5	Module Interfaces.....	10
2.6	Roles, Services, and Authentication.....	11
2.6.1	Operator Services and Descriptions	11
2.6.2	Operator Authentication.....	15
2.7	Physical Security	16
2.8	Operational Environment	16
2.9	Cryptographic Key Management.....	17
2.9.1	Key Generation	20
2.9.2	Key Entry, Output, and Protection	20
2.10	Self-Tests.....	21
2.10.1	Power-On Self-Tests.....	21
2.10.2	Conditional Self-Tests.....	22
2.11	Mitigation of Other Attacks.....	22
3	Guidance and Secure Operation.....	23
3.1	Crypto Officer and User Guidance	23
3.1.1	Software Packaging and OS Requirements	23
3.1.2	Enabling FIPS Mode.....	23
3.1.3	Additional Rules of Operation.....	24

List of Tables

Table 1 – Acronyms and Terms.....6

Table 2 – Validation Level by DTR Section8

Table 3 – FIPS-Approved Algorithm Certificates Crypto J.....9

Table 4 – FIPS-Approved Algorithm Certificates Crypto C ME.....9

Table 5 – Logical Interface / Physical Interface Mapping11

Table 6 – Authenticated Module Services and Descriptions for Crypto J Implementation.....13

Table 7 – Unauthenticated Module Services and Descriptions for Crypto J Implementation.....13

Table 8 – Authenticated Module Services and Descriptions for Crypto C ME Implementation.....15

Table 9 – Unauthenticated Module Services and Descriptions for Crypto C ME Implementation15

Table 10 – Module Keys/CSPs.....20

Table 11 – Module Power-On Self-Tests21

Table 12 – Module Conditional Self-Tests22

List of Figures

Figure 1 – Module Interfaces Diagram10

1 Introduction

1.1 About FIPS 140

Federal Information Processing Standards Publication 140-2 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Communications Security Establishment of Canada (CSEC) Cryptographic Module Validation Program (CMVP) runs the FIPS 140 program. The CMVP accredits independent testing labs to perform FIPS 140 testing; the CMVP also validates test reports for products meeting FIPS 140 validation. *Validated* is the term given to a product that is documented and tested against the FIPS 140 criteria.

More information is available on the CMVP website at <http://csrc.nist.gov/groups/STM/cmvp/index.html>.

1.2 About this Document

This non-proprietary Cryptographic Module Security Policy for the ePO Cryptographic Module (Version 1.0) from McAfee provides an overview of the product and a high-level description of how it meets the security requirements of FIPS 140-2. This document contains details on the module's cryptographic keys and critical security parameters. This Security Policy concludes with instructions and guidance on running the module in a FIPS 140-2 mode of operation.

The McAfee ePO Cryptographic Module (Version 1.0) may also be referred to as the “module” in this document.

1.3 External Resources

The McAfee website (<http://www.mcafee.com>) contains information on the full line of products from McAfee, including a detailed overview of the ePO Cryptographic Module (Version 1.0) solution. The Cryptographic Module Validation Program website (<http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2011.htm>) contains links to the FIPS 140-2 certificate and McAfee contact information.

1.4 Notices

This document may be freely reproduced and distributed in its entirety without modification.

1.5 Acronyms

The following table defines acronyms found in this document:

Acronym	Term
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CSEC	Communications Security Establishment of Canada
CSP	Critical Security Parameter
DTR	Derived Testing Requirement
ePO	ePolicy Orchestrator
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
GPOS	General Purpose Operating System
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
KAT	Known Answer Test
NIST	National Institute of Standards and Technology
RSA	Rivest Shamir Adelman
RSD	Remote Sensor Detection
SHA	Secure Hashing Algorithm

Table 1 – Acronyms and Terms

2 McAfee ePO Cryptographic Module (Version 1.0)

2.1 Product Overview

McAfee ePolicy Orchestrator is a scalable management framework for centralized policy management and enforcement of McAfee's security products and the systems on which they reside. ePO provides the user interface for McAfee products and performs the following tasks:

- Agent deployment
- Security Management
- Reporting

More information on ePO can be found at <http://www.mcafee.com/us/products/epolicy-orchestrator.aspx>.

2.2 Cryptographic Module Specification

The module is the McAfee ePO Cryptographic Module (Version 1.0), provides the ePO application with cryptographic functionality. The module is a software-only module installed on a multi-chip standalone device, such as a General Purpose Computer running a General Purpose Operating System and provides cryptographic services to the McAfee ePO application.

The module is a uniquely identifiable set of libraries built into the ePO application. All operations of the module occur via calls from the ePO application and its internal daemons, and all calls are authenticated via digital signature. As such there are no untrusted services or daemons calling the services of the module. No security functions outside the cryptographic module provide FIPS-relevant functionality to the module.

Once configured for FIPS mode of operation (see the Guidance and Secure Operation section), the module cannot be placed into a non-FIPS mode.

The boundary is composed of the following files:

- ccme_base.dll
- cryptocme2.dll
- nmcomn32.dll
- jsafeJCEFIPS.jar
- orion-core-common.jar

This module contains two embedded, validated modules (see certificates 1047 and 1092). The module operates with both embedded modules simultaneously. Nmcomn32.dll accepts calls for cryptographic operations and performs the module integrity check over all files in the cryptographic boundary.

2.3 Validation Level Detail

The following table lists the level of validation for each area in FIPS 140-2:

FIPS 140-2 Section Title	Validation Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	2
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
Electromagnetic Interference / Electromagnetic Compatibility	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A

Table 2 – Validation Level by DTR Section

The “Mitigation of Other Attacks” section is not relevant as the module does not implement any countermeasures towards special attacks.

2.4 Cryptographic Algorithms

2.4.1 Algorithm Implementation Certificates

The module’s cryptographic algorithm implementations¹ have received the following certificate numbers from the Cryptographic Algorithm Validation Program:

Algorithm Type	Algorithm	CAVP Certificate	Use
Asymmetric Key	RSA 3072-bit	312	Sign / verify operations Key establishment Module integrity
Hashing	SHA-1, SHA-256	703	User password hashing Digital signature generation and validation (SHA-256)
Symmetric Key	AES 128 in ECB mode	670	Data encryption/ decryption

¹ Please note that the standards for each algorithm are listed with the respective CAVP certificate.

Algorithm Type	Algorithm	CAVP Certificate	Use
Random Number Generation	FIPS 186-2 PRNG (Change Notice 1-with and without the mod q step)	390	Random Number Generation

Table 3 – FIPS-Approved Algorithm Certificates Crypto J²

Algorithm Type	Algorithm	CAVP Certificate	Use
Asymmetric Key	RSA 2048-bit RSA 3072-bit	412	Sign / verify operations Module integrity
	DSA 1024-bit	311	Verification of legacy data
Hashing	SHA-1, SHA-256	855	Digital signature generation and verification (SHA-256) Verification of legacy data (SHA-1) User password hashing
Symmetric Key	AES 128-bit and 256-bit in CBC and ECB mode	860	Data encryption/ decryption
	3DES mode CBC mode	707	Decryption of legacy data
Random Number Generation	FIPS 186-2 PRNG (Change Notice 1-with and without the mod q step)	492	Random Number Generation

Table 4 – FIPS-Approved Algorithm Certificates Crypto C ME³

2.4.2 Non-Approved Algorithms

The module implements the following non-FIPS approved algorithms:

- Software-based random number generator
 - This RNG is used only as a seeding mechanism to the FIPS-approved PRNG.

² Note this implementation has received FIPS 140-2 Level 1 validation certificate #1047:
<http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm#1047>

³ Note this implementation has received FIPS 140-2 Level 1 validation certificate #1092:
<http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2009.htm#1092>

2.5 Module Interfaces

The figure below shows the module's physical and logical block diagram:

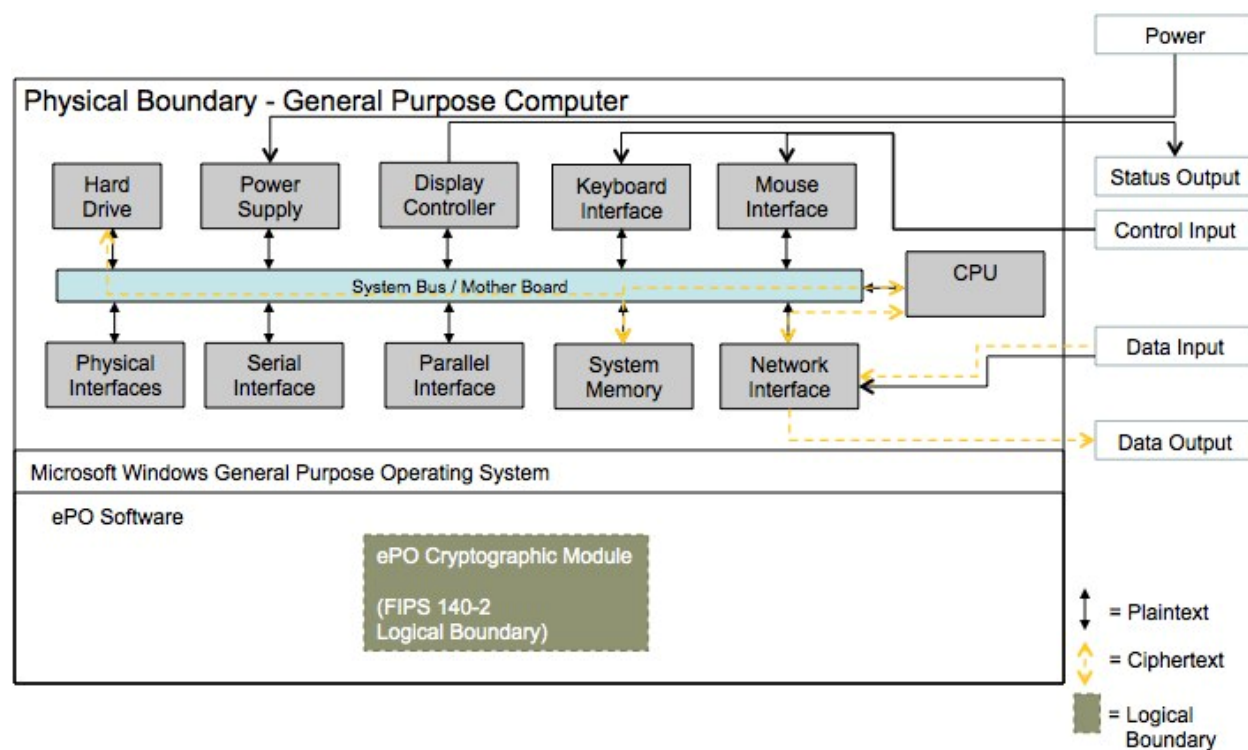


Figure 1 – Module Interfaces Diagram

The interfaces (ports) for the physical boundary include the computer keyboard port, CDROM drive, floppy disk, mouse, network port, parallel port, USB ports, monitor port and power plug. When operational, the module does not transmit any information across these physical ports because it is a software cryptographic module. Therefore, the module's interfaces are purely logical and are provided through the Application Programming Interface (API) that a calling daemon can operate. The logical interfaces expose services that applications directly call, and the API provides functions that may be called by a referencing application (see Section 2.6 – Roles, Services, and Authentication for the list of available functions).

The API provided by the module is mapped onto the FIPS 140-2 logical interfaces: data input, data output, control input, and status output. Each of the FIPS 140-2 logical interfaces relates to the module's callable interface, as follows:

FIPS 140-2 Interface	Logical Interface	Module Physical Interface
Data Input	Input parameters of API function calls	Ethernet/Network port
Data Output	Output parameters of API function calls	Ethernet/Network port

FIPS 140-2 Interface	Logical Interface	Module Physical Interface
Control Input	API function calls	Keyboard and mouse
Status Output	For FIPS mode, function calls returning status information and return codes provided by API function calls.	Monitor
Power	None	Power supply/connector

Table 5 – Logical Interface / Physical Interface Mapping

The module's logical interfaces are provided only through the Application Programming Interface (API) that a calling daemon can operate. The module distinguishes between logical interfaces by logically separating the information according to the defined API.

As shown in Figure 1 – Module Interfaces Diagram and Table 6 – Authenticated Module Services and Descriptions, the output data path is provided by the data interfaces and is logically disconnected from processes performing key generation or zeroization. No key information will be output through the data output interface when the module zeroizes keys.

2.6 Roles, Services, and Authentication

The module supports a Crypto Officer and a User role as specified in the following section. The module does not support a Maintenance role.

2.6.1 Operator Services and Descriptions

The services available to the User and Crypto Officer roles in the module are as follows:

Service	Description	Service Input/Output (API)	Key/CSP Access	Roles
Configure	Initializes the module for FIPS mode of operation	<code>com.mcafee.orion.core.security.FIPS140.init()</code>	RSA Integrity and Authentication Public Key	User
Sign / Verify	Sign data or Verify a signature	<p>CryptoJ classes which are subtypes of Java's <code>java.security.SignatureSpi</code> abstract class.</p> <p>Example Usage: <code>com.mcafee.orion.core.CertUtil.generateSignedCert()</code> <code>com.mcafee.orion.core.ext.ExtensionSignatureUtil</code></p>	RSA Public Key RSA Private Key DSA Public Key for verification of legacy data RSA Integrity and Authentication Public Key	User
Encrypt / Decrypt	Encrypts or Decrypts a block of data	<p>CryptoJ classes which are subtypes of Java's <code>javax.crypto.CipherSpi</code> abstract class.</p> <p>Example Usage: <code>com.mcafee.orion.core.cert.DefaultObfuscator</code></p>	Session Key TDES Legacy Key RSA Integrity and Authentication Public Key	User
Random Number Generation	Generates random numbers for crypto operations	<p>CryptoJ classes which are subtypes of Java's <code>java.security.SecureRandomSpi</code> class.</p> <p>Example Usage: <code>com.mcafee.orion.core.cert.CertUtil</code> <code>com.mcafee.orion.core.cert.DefaultObfuscator</code> <code>org.apache.tomcat.util.net.jsse.JSSE14SocketFactory</code></p>	FIPS 186-2 PRNG Seed FIPS 186-2 PRNG Seed Key RSA Integrity and Authentication Public Key	User
Self Test	Performs self tests on critical functions of module	<code>com.mcafee.orion.core.RSASecurityProviders</code> <code>runFIPS140SelfChecks()</code>	RSA Integrity and Authentication Public Key	User
Show Status	Shows status of the module and view audit information	<code>com.rsa.jsafe.CryptoJ.getMode()</code> <code>com.rsa.jsafe.CryptoJ.getState()</code>	RSA Integrity and Authentication Public Key	User

Service	Description	Service Input/Output (API)	Key/CSP Access	Roles
Zeroization	Zeroizes keys	<code>com.rsa.jsafe.provider.SensitiveData.clear()</code>	RSA Integrity and Authentication Public Key	User

Table 6 – Authenticated Module Services and Descriptions for Crypto J Implementation

Service	Description	Service Input/Output (API)	Key/CSP Access	Roles
Reboot	Restart module or application	Restart module or application	None	Crypto Officer
Procedural Zeroization	Zeroize keys stored on disk in keystore	RAM keys are zeroized when the Operating System clears the process memory, static Keys stored on disk are zeroized according to IG 7.9 by uninstalling the module and formatting the disk	All Keys	Crypto Officer

Table 7 – Unauthenticated Module Services and Descriptions for Crypto J Implementation

Service	Description	Service Input/Output (API)	Key/CSP Access	Roles
Configure	Initializes the module for FIPS mode of operation	<code>R_FIPS140_library_init()</code> <code>PRODUCT_FIPS_140_MODE_RESOURCE_LIST()</code> <code>R_FIPS140_get_default()</code> <code>R_FIPS140_get_mode()</code> <code>R_FIPS140_get_info()</code>	RSA Integrity and Authentication Public Key	User
Sign	Sign data	<code>R_CR_sign()</code>	RSA Private Key RSA Integrity and Authentication Public Key	User
Verify	Verify a signature	<code>R_CR_verify()</code>	RSA Public Key DSA Public Key for verification of legacy data RSA Integrity and Authentication Public Key	User

Service	Description	Service Input/Output (API)	Key/CSP Access	Roles
Decrypt	Decrypts a block of data Using AES	R_CR_decrypt_final() R_CR_decrypt_init() R_CR_decrypt_update()	Session Key TDES Legacy Key RSA Integrity and Authentication Public Key	User
Encrypt	Encrypts a block of data Using AES	R_CR_encrypt_final() R_CR_encrypt_init() R_CR_encrypt_update()	Session Key RSA Integrity and Authentication Public Key	User
Random Number Generation	Generates random numbers for crypto operations	R_CR_random_bytes() R_CR_random_seed()	FIPS 186-2 PRNG Seed FIPS 186-2 PRNG Seed Key RSA Integrity and Authentication Public Key	User
Generate Key	Generates an asymmetric key	R_PKEY_from_public_key_binary() R_PKEY_to_public_key_binary() R_PKEY_from_binary() R_PKEY_to_binary() R_CR_generate_key()	RSA Public Key RSA Private Key RSA Integrity and Authentication Public Key	User
Self Test	Performs self tests on critical functions of module	comnutil_CheckTrustedModules()	RSA Integrity and Authentication Public Key	User
Show Status	Shows status of the module and view audit information	R_FIPS140_get_mode()	RSA Integrity and Authentication Public Key	User

Service	Description	Service Input/Output (API)	Key/CSP Access	Roles
Zeroization	Zeroizes keys	R_PKEY_free() R_SKEY_free()	RSA Integrity and Authentication Public Key	User

Table 8 – Authenticated Module Services and Descriptions for Crypto C ME Implementation

Service	Description	Service Input/Output (API)	Key/CSP Access	Roles
Reboot	Restart module or application	Restart module or application	None	Crypto Officer
Procedural Zeroization	Zeroize keys stored on disk in keystore	RAM keys are zeroized when the Operating System clears the process memory, static Keys stored on disk are zeroized according to IG 7.9 by uninstalling the module and formatting the disk	All Keys	Crypto Officer

Table 9 – Unauthenticated Module Services and Descriptions for Crypto C ME Implementation

2.6.2 Operator Authentication

The module supports Level 2 requirements for authentication, which defines role-based authentication. The module verifies the digital signatures of calling daemons prior to the allowing access to any module services. The signature is RSA 2048-bit key with SHA-256 hash signature. Since this key has 112-bits of security strength the probability of a successful random attempt is $1/2^{112}$, which is less than $1/1,000,000$. Assuming a scripted attack of 60 attempts in one minute, the probability of a success with multiple consecutive attempts in a one-minute period is $60/2^{112}$ which is less than $1/100,000$.

The module contains User authentication data in the form of the public key but does not contain CO authentication data. The User Services require authentication, which is performed by the module as described above. The Crypto Officer services do not require authentication as they are not security relevant functions. The Reboot and Procedural Zeroization services do not affect the security of the module; these services do not create, disclose, or substitute cryptographic keys or CSPs, nor do they utilize any Approved security functions.

The module does not permit an operator to change roles. The services described in Table 6 – Authenticated Module Services and Descriptions for Crypto J Implementation and Table 8 – Authenticated Module Services and Descriptions for Crypto C ME Implementation are only available to the roles specified, and access to these services is protected via authentication mechanisms discussed above.

2.7 Physical Security

This section of requirements does not apply to this module. The module is a software-only module and does not implement any physical security mechanisms.

2.8 Operational Environment

The module operates on a general-purpose computer (GPC) running a general-purpose operating system (GPOS). The module was tested on the following:

- Microsoft Windows XP Professional on Intel Core2 Duo

For FIPS purposes, the module is running on a platform in single user mode and does not require any additional configuration to meet the FIPS requirements.

The GPC(s) used during testing met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B. FIPS 140-2 validation compliance is maintained when the module is operated on other versions of the Microsoft Windows GPOS running in single user mode, assuming that the requirements outlined in NIST IG G.5 are met.

2.9 Cryptographic Key Management

The table below provides a complete list of Critical Security Parameters used within the module:

Key/CSP Name	Description / Use	Generation	Storage	Establishment / Export	Interface	Privileges
RSA Integrity and Authentication Public Key	Public 2048-bit RSA certificate used to verify binaries at startup (integrity test and authentication)	Generated at build time	Storage: RAM, on-disk in plaintext Type: Ephemeral Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: Electronic Output: NA	Module Integrity check (verify)	CO R W D User R W D
RSA Public Key	RSA 2048-bit or 3072-bit public key for signature verification RSA 1024-bit public key for verification of legacy signatures	Generated via FIPS-approved PRNG	Storage: RAM plaintext Type: Ephemeral Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: NA Output: NA	Sign / verify	CO D User R W D

Key/CSP Name	Description / Use	Generation	Storage	Establishment / Export	Interface	Privileges
RSA Private Key	RSA 2048-bit or 3072-bit private key for signature generation	Generated via FIPS-approved PRNG	Storage: RAM plaintext Type: Ephemeral Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: NA Output: NA	Sign / verify	CO D User R W D
Session Key	General purpose AES 128-bit ECB key for data encryption / decryption	Generated via FIPS-approved PRNG	Storage: RAM plaintext Type: Ephemeral Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: NA Output: NA	Encrypt / Decrypt	CO D User R W D
TDES Legacy Key	General purpose TDES key for data decryption of legacy data	Passed by calling process	Storage: RAM plaintext Type: Ephemeral Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: Electronic by calling application Output: NA	Decrypt	CO D User R W D

Key/CSP Name	Description / Use	Generation	Storage	Establishment / Export	Interface	Privileges
DSA Public Key	McAfee public repository DSA 1024-bit key for verifying legacy signatures	Generated at build time	Storage: RAM, on-disk in plaintext Type: Static Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: Electronic by calling application Output: NA	Verify	CO D User R W D
FIPS 186-2 PRNG Seed	Seed value for approved PRNG	Internally generated	Storage: RAM plaintext Type: Ephemeral Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: NA Output: NA	Random Number Generation	CO D User R W D
FIPS 186-2 PRNG Seed Key	Seed key for approved PRNG	Internally generated	Storage: RAM plaintext Type: Ephemeral Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: NA Output: NA	Random Number Generation	CO D User R W D

Key/CSP Name	Description / Use	Generation	Storage	Establishment / Export	Interface	Privileges
CO Password	Crypto Officer password	No	Storage: on disk plaintext Type: Static Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.	Agreement: NA Entry: Electronic Output: NA	Control Input Physical Interface	CO R W D

R = Read W = Write D = Delete

Table 10 – Module Keys/CSPs

2.9.1 Key Generation

The module supports the generation of the asymmetric and symmetric keys via Federal Information processing Standard 186-2, Digital Signature Standard (FIPS 186-2) Approved random number generator.

2.9.2 Key Entry, Output, and Protection

All keys and CSPs reside on memory internally allocated by the module and can only be output using the exposed APIs. The module does not support key entry or output from the physical boundary. The operating system and runtime environment protect the memory and process space from unauthorized access.

2.10 Self-Tests

The module includes an array of self-tests that are run during startup and periodically during operations to prevent any secure data from being released and to ensure all components are functioning correctly. In the event of any self-test failure, the module/ePO application will output an error to the audit log and will shutdown. In addition to self-test failures, successful loading of the module is also logged. To access status of self-tests, success or failure, the application provides access to the audit log. Status is viewable via operating environment's audit mechanism and by verifying proper loading and operation of the ePO application. While the module is running self-tests, the module will not output data. The ePO application makes calls to the ePO Cryptographic Module (Version 1.0), and data will not be returned until the self-tests complete.

No keys or CSPs will be output when the module is in an error state. The module will halt and the process will terminate; as such, no data will be output via the data output interface. Additionally, the module does not support a bypass function, and the module does not allow plaintext cryptographic key components or other unprotected CSPs to be output on physical ports. No external software or firmware is allowed to be loaded in a FIPS mode of operation.

The following sections discuss the module's self-tests in more detail.

2.10.1 Power-On Self-Tests

Power-on self-tests are run upon every initialization of the module and if any of the tests fail, the module will not initialize. The module will enter an error state and no services can be accessed by the users. The module implements the following power-on self-tests:

Implementation	POST
Crypto J	<ul style="list-style-type: none"> • RSA pairwise consistency (signing and signature verification) • SHA-1 and SHA-256 KAT • AES KAT (encryption and decryption) • KAT for Approved PRNG • KAT for non-approved software RNG
Crypto C ME	<ul style="list-style-type: none"> • RSA pairwise consistency (signing and signature verification) • DSA pairwise consistency (signing and signature verification) • SHA-1 and SHA-256 KAT • AES KAT (encryption and decryption) • TDES KAT (encryption and decryption) • KAT for Approved PRNG • KAT for non-approved software RNG
ePO	<ul style="list-style-type: none"> • Module integrity check via Crypto J implementation of RSA

Table 11 – Module Power-On Self-Tests

The module performs all power-on self-tests automatically when the module is initialized. All power-on self-tests must be passed before a User/Crypto Officer can perform services. The Power-on self-tests can

be run on demand by reinitializing the module in FIPS approved Mode of Operation. Upon passing the power-on self-tests, the module will log the success and will continue to boot normally; successful loading of the ePO application will indicate that all self-tests have passed. If a self-test fails, the module will not load, the ePO application will halt, and an error will be logged in the server.log file.

2.10.2 Conditional Self-Tests

Conditional self-tests are on-demand tests and tests run continuously during operation of the module. If any of these tests fail, the module will enter an error state and no services can be accessed by the users. The module can be re-initialized to clear the error and resume FIPS mode of operation. The module performs the following conditional self-tests:

Implementation	Conditional Test
Crypto J	<ul style="list-style-type: none"> • RSA pairwise consistency • Continuous RNG test run on output of Approved PRNG • Continuous test on output of Approved PRNG seed mechanism • Test to ensure Approved PRNG output and seed do not match
Crypto C ME	<ul style="list-style-type: none"> • RSA pairwise consistency • DSA pairwise consistency • Continuous RNG test run on output of Approved PRNG • Continuous test on output of Approved PRNG seed mechanism • Test to ensure Approved PRNG output and seed do not match

Table 12 – Module Conditional Self-Tests

The module will inhibit data output via the output interface when conditional tests are performed. Once the tests have passed and the keys have been generated, the module will pass the key to the calling daemon. If a self-test fails, the module will halt, the ePO application will halt, and an error will be logged in the server.log file.

2.11 Mitigation of Other Attacks

The module does not mitigate other attacks.

3 Guidance and Secure Operation

This section describes how to configure the module for FIPS-approved mode of operation. Operating the module without maintaining the following settings will remove the module from the FIPS-approved mode of operation.

3.1 Crypto Officer and User Guidance

3.1.1 Software Packaging and OS Requirements

The ePO Cryptographic Module (Version 1.0) can only be used by the ePO application, which is not being validated under FIPS 140-2. The ePO application must be installed on a Microsoft Windows Operating System running in single user mode. To configure single-user mode, the following must be disabled:

- Remote registry and remote desktop services
- Remote assistance
- Guest accounts
- Server and terminal services

Contact Microsoft support for configuration details; specific configuration steps are beyond the scope of this document.

3.1.2 Enabling FIPS Mode

To meet the cryptographic security requirements, certain restrictions on the installation and use of ePO must be followed. The steps below will ensure that the module implements all required self-tests and uses only approved algorithms. Please note that once the module is in FIPS-approved mode, it cannot transition to a non-approved mode.

3.1.2.1 Installation

1. The installation must be a new install. Upgrading from a previous version of ePO is not valid.
2. During the installation process, enter the following command at the CLI:

```
C:\> <ePO 4.6 install package>\setup.exe ENABLEFIPSMODE=1
```

3. After installation, edit <ePO install dir>\server\logs\log-config.xml. Change the `priority` value parameter of the root logger level from `warn` to `info`.

3.1.3 Additional Rules of Operation

1. All host system components that can contain sensitive cryptographic data (main memory, system bus, disk storage) must be located in a secure environment.
2. The writable memory areas of the Module (data and stack segments) are accessible only by the ePO application so that the Module is in "single user" mode, i.e. only the ePO application has access to that instance of the Module.
3. Only 2048-bit asymmetric keys should be used where available.
4. The operating system is responsible for multitasking operations so that other processes cannot access the address space of the process containing the Module.