Trellix Core Cryptographic Module (user)

FIPS 140-2 Non-Proprietary Security Policy

Version: 2.2.0.17.0

Date: March 8, 2024

Trellix
6220 America Center Drive
San Jose, CA 95002
888.847.8766
https://www.trellix.com/

Prepared for Trellix by

2400 Research Bvld, Suite 395
Rockville, MD 20850
Phone: +1 (703) 375 9820
Info@acumensecurity.net
https://www.acumensecurity.net

Table of Contents

1. Introduction 4
2. Cryptographic Module Specification 6
2.1 Cryptographic Boundary 6
2.2 Ports and Interfaces 8
2.3 Mode of Operation 8
2.4 Transitioning Between Modes of Operation 8
3. Cryptographic Functionality 8
3.1 Approved Cryptographic Algorithms 8
3.2 Allowed Cryptographic Algorithms and Protocols 9
3.3 Non-Approved Algorithms 9
3.4 Cryptographic Key Management 10
3.4.1 Key Generation 11
3.4.2 Key Zeroization 11
4. Roles, Services and Authentication 11
4.1 Roles 11
4.2 Services 12
4.3 Authentication 14
5. Self-tests 14
5.1 Power-On Self-Tests 14
5.2 Conditional Self Tests 15
6. Physical Security 15
7. Operational Environment 15
8. Guidance and Secure Operation 15
9. Mitigation of other Attacks 16
10. Design Assurance 16
11. References and Standards 17
12. Acronyms and Definitions 17

List of Tables

Table 1 - Tested Operational Environments 4
Table 2 - Security Level Detail 5
Table 3 – Module binary image 6
Table 4 - Ports and Interfaces 8
Table 5 - Approved Algorithms and CAVP Certificates 9
Table 6 - Cryptographic Algorithms Allowed in FIPS Mode 9
Table 7 - Non-FIPS Approved Algorithms 10
Table 8 – Secret Keys, Private Keys and CSPs 11
Table 9 – Roles 12
Table 10 - Approved Services, Roles and Access Rights 14
Table 11 - Power-up self-tests 15
Table 12 - Conditional self-tests 15
Table 13 - References 17
Table 14 - Acronyms 18

List of Figures

Figure 1 - Logical and Physical Boundary – Preboot 7
Figure 2 - Logical and Physical Boundary – Windows environments 7

1. Introduction

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 Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140 program. The NVLAP accredits independent testing labs to perform FIPS 140 testing; the CMVP validates modules meeting FIPS 140 validation. Validated is the term given to a module 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.

About this Document

This non-proprietary Cryptographic Module Security Policy for Trellix Core Cryptographic Module (user) from Trellix provides an overview of the product and a high-level description of how it meets the overall Level 1 security requirements of FIPS 140-2.

The Trellix Core Cryptographic Module (user) module may also be referred to as the "module" in this document.

The Cryptographic Module version 2.2.0.17.0 is defined as multiple-chip standalone for the purposes of FIPS 140-2, and the module was tested on the operational environments and platforms detailed in table 1. The module's source code is the same for all operational environments.

#	Operational Environment	Hardware Platform	Processor	PAA[1]/Acceleration
1	Windows 10 64-bit	Microsoft Surface 3	Intel i5-520M	With and without AES-NI
2	Windows 7 32-bit	Dell Latitude E7270	Intel i5-6300U	with and without AES-NI
3	UEFI 64-bit Preboot	Microsoft Surface 3	Intel i5-520M	with and without AES-NI
4	BIOS Preboot	Dell Inspiron 5558	Intel i5-5250U	with and without AES-NI

Table 1 - Tested Operational Environments

The Cryptographic Module is also supported on the following operating environments for which operational testing and algorithm testing was not performed:

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012

[1] AES-NI (the Intel Advanced Encryption Standard (AES) New Instructions (AES-NI)) is an extension to the x86 instruction set architecture for microprocessors from Intel and AMD. The purpose of the instruction set is to improve the speed of applications performing encryption and decryption using AES.
The MFE algorithm implementation was tested with and without PAA for all operating environments however the BSAFE algorithm implementation was tested with PAA for all environments except Windows 7 32-bit; IG 1.21 is met in the algorithm testing of this module.

- Windows Server 2008
- Windows 11
- Windows 8.1 32-bit and 64-bit
- Windows 8 32-bit and 64-bit
- Windows 7-64-bit

As per FIPS 140-2 Implementation Guidance G.5, compliance is maintained by vendor or user affirmation for other versions of the respective operational environments where the module binary is unchanged. The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys so when ported if the specific operational environment is not listed on the validation certificate.

The platform(s) used during testing met Federal Communications Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part 15, Subpart B.

The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys so when ported if the specific operational environment is not listed on the validation certificate.

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

FIPS 140-2 Section Title	Level
1. Cryptographic Module Specification	1
2. Cryptographic Module Ports and Interfaces	1
3. Roles, Services, and Authentication	1
4. Finite State Model	1
5. Physical Security	N/A
6. Operational Environment	1
7. Cryptographic Key Management	1
8. EMI/EMC	1
9. Self‑Tests	1
10. Design Assurance	3
11. Mitigation of Other Attacks	N/A
Overall Level	1

Table 2 - Security Level Detail

2. Cryptographic Module Specification

This section provides the details of how the module meets the FIPS 140-2 requirements.

2.1 Overview

The module provides cryptographic services to Trellix products and is packaged differently depending on its operational environment.

There are no specific hardware or firmware requirements for the module. The module is a software only module, which resides on a General-Purpose Computer (see Figure 1 - Logical and Physical Boundary). The module's physical boundary is that of the device on which it is installed. The device shall be running a supported operating system (OS) and supporting all standard interfaces, including keys, buttons and switches, and data ports.

The module is packaged as distinct binary images, one for each of the following operating environments:

FILE NAME	OPERATING ENVIRONMENT	PACKAGE
MFECCF[32|64]*aa*.dll	Microsoft Windows	32-bit/64-bit
MFECCF[32|64]*aa*.dlm	BIOS Pre-boot environment	32-bit/64-bit
MFECCF[32|64]*aa*.efi	UEFI (PC Pre-boot environment)	32-bit/64-bit

Table 3 – Module binary image

Note: "*aa*" are alphanumeric product identifiers, with prefixes such as MFECCF32DE or MFECCF64DE for the Trellix product Drive Encryption and MFECCF32FF or MFECCF64FF for the Trellix product Files and Removable Media Protection. The differences in the module due to the *"aa"* product identifiers for a particular operating environment are in file name only.

Although there is a common core of functionality between these images, within this group there are distinct variants. The differences in implementation relate to:

1. An AES implementation that is not utilized in the pre-boot builds. The source code is the same, but the functionality is not used in some environments. The low-level disk handler AES implementation is only applicable to the non-preboot environment.

Except for in the distinct areas clearly described above, the various modules are identical and, in those cases, the term "the module" applies equally to each variant.

2.1 Cryptographic Boundary

Figure 1 and Figure 2 show the logical relationship of the module to the other software and hardware components of the computer.

The module is a software module running in a pre-boot or Windows operating environment on a general-purpose computer. The processor of this platform executes all software. All software components of the module are persistently stored within the device and, while executing, are stored in the device local RAM.

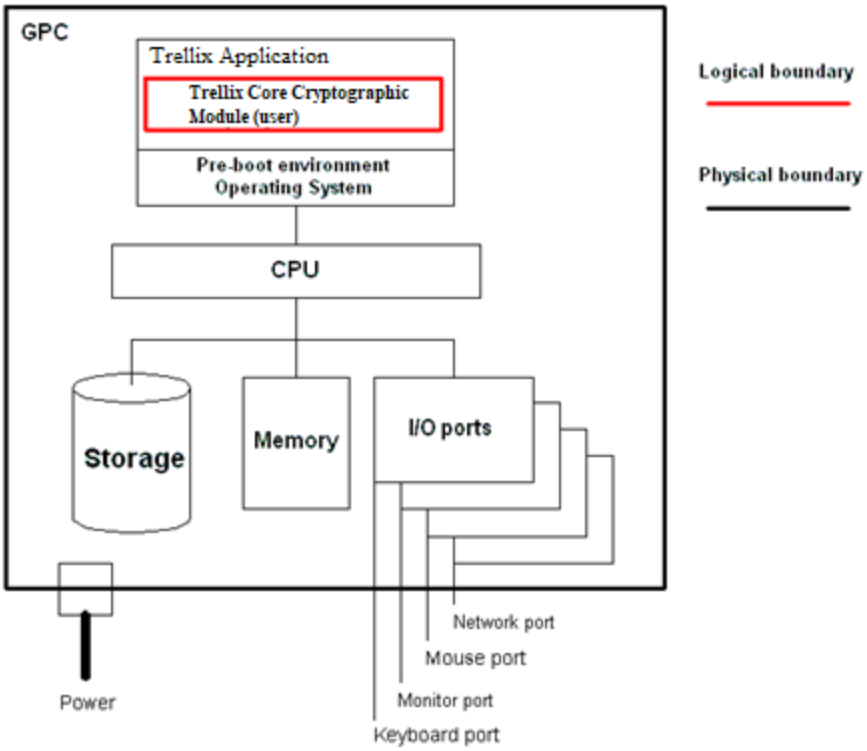

Figure 1 - Logical and Physical Boundary – Preboot

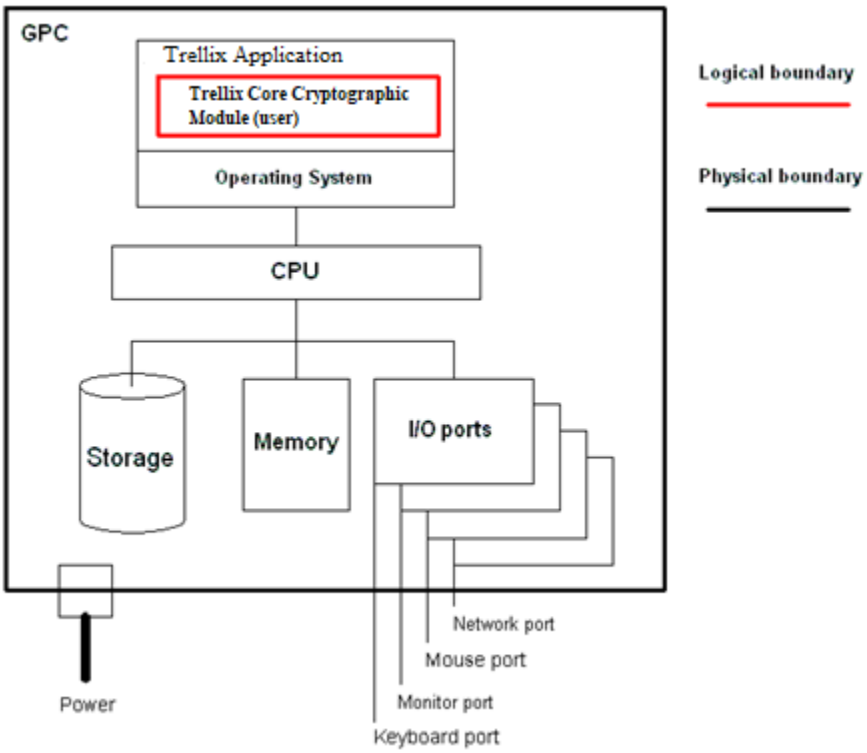

Figure 2 - Logical and Physical Boundary – Windows environments.

2.2 Ports and Interfaces

The module provides all logical interfaces via Application Programming Interface (API) calls. These logical interfaces expose services (described in section 4.2) that the User and operating system may utilize directly.

The logical interfaces provided by the module are mapped onto the FIPS 140-2 logical interfaces: data input, data output, control input, and status output as follows:

Description	Logical Interface Type
Parameters passed to the module via API calls	Data Input
Data returned from the module via API calls	Data Output
API Calls and/or parameters passed to API calls	Control Input
Information received in response to API calls	Status Output
There is no separate power or maintenance access interface beyond the power interface provided by the GPC itself	Power Interface

Table 4 - Ports and Interfaces

2.3 Mode of Operation

The module supports an Approved mode of operation and a non-approved mode of operation.

2.4 Transitioning Between Modes of Operation

During operation, the module can switch service by service between an Approved mode of operation and a non-Approved mode of operation. The module will transition to the non-Approved mode of operation when a non-Approved security function in Table 6 is utilized in lieu of an Approved one. The module can transition back to the Approved mode of operation by utilizing an Approved security function.

CSPs shall not be shared between the Approved and non-Approved modes of operation. The module shall zeroize all CSPs through a power-cycle whenever switching from mode of operation to another mode of operation.

3. Cryptographic Functionality

The module implements the FIPS Approved (table 5) and Non-Approved but Allowed (table 6) cryptographic functions in a FIPS Approved mode of operation. Non-approved algorithms are listed in table 7.

3.1 Approved Cryptographic Algorithms

The approved and vendor affirmed security functions included in the Cryptographic Module are utilized by the module's callable services or internal functions.

CAVP /ACVP Cert	Algorithm [Standard]	Mode/Method	Description / Key Size(s)	Use / Function
A1555	SHS [180-4]	SHA-256	256-bit	Used in the module integrity test.
A1556	SHS [180-4]	SHA-256	256-bit	Hashing service

CAVP /ACVP Cert	Algorithm [Standard]	Mode/Method	Description / Key Size(s)	Use / Function
A1555 **A1556**	HMAC [198]	HMAC-SHA-256	256-bit	Module integrity testing and constituent of the random number generator.
A1556	HMAC SHA-256 DRBG [90A]		256-bit	Symmetric and Asymmetric key generation.
A1555	AES [197]	CBC [38A] CFB8 [38A] ECB [38A]	256-bit	Service provided to encrypt and decrypt blocks of data.
A1556	AES [197]	CBC [38A] CFB128 [38A] ECB [38A]	256-bit	Service provided to encrypt and decrypt blocks of data.
A1556	RSA [186]	Key Generation Mode B.3.6	2048-bit	Key generation
Vendor Affirmed	CKG [133]		Symmetric key generation using unmodified DRBG output	Key generation

Table 5 - Approved Algorithms and CAVP Certificates

Notes:

There are two implementations of AES-256 in the module, one to support each of the following:

- MCCi_crypto_sym_encryptor service (BSAFE)
- MCCi_crypto_disk_algs service interface (MFE) – this can run on processors with or without AES-NI capability. However, it will use AES-NI instructions if run on AES-NI enabled processors. This also supports the low-level disk handler (INT 13 disk hook, Windows only).

3.2 Allowed Cryptographic Algorithms and Protocols

The following cryptographic algorithms are not approved but are allowed to be used in a FIPS approved mode of operation.

Algorithm	Caveat	Use / Function
RSA	Key wrapping, key establishment provides 112 bits encryption strength, PKCS 1.5 padding is used by default	Encrypt/decrypt

Table 6 - Cryptographic Algorithms Allowed in FIPS Mode

3.3 Non-Approved Algorithms

The module supports the following non-FIPS approved and not allowed algorithms. The use of the non-conformant algorithms listed in Table 7 will place the module in a non-approved mode of operation.

Algorithm/Function	Use/Function
PKCS#5	A password hashing algorithm using SHA-1 (non-compliant).
SHA-1	Used by PKCS#5

Table 7 - Non-FIPS Approved Algorithms

PKCS#5 is not allowed for use in the FIPS Approved mode of operation. When this algorithm is used, the module is not operating in the FIPS Approved mode of operation.

3.4 Cryptographic Key Management

The table below provides a complete list of Private Keys and CSPs used by the module:

Key/CSP Name	Key/CSP Type	Generation/Input	Output	Storage	Zeroization	Use
Data Key	AES 256 bit	Generated using key generation service (MCCi_rand_source) and/or entered into the module via user service API	N/A	Plaintext in RAM	Zeroized using the key zeroization service.	To encrypt and decrypt data using the symmetric encryption services.
Public Key	RSA 2048 bit	Generated using key generation service and/or entered into the module via user service API	Output from the module via user service API.	Plaintext in RAM	Zeroized using the key zeroization service.	To encrypt data or keys using one of the asymmetric encryption services.[2]
Private Key	RSA 2048 bit	Generated using key generation service and/or entered into the module via user service API	Output from the module via user service API.	Plaintext in RAM	Zeroized using the key zeroization service.	To decrypt data or keys previously encrypted by the Public Key.
HMAC DRBG CSPs: Key, V, seed	Key CSP: 512 bits V CSP: 512 bits Seed CSPs: 2048 bits	Initial value of 64 bytes all set to "0x00". The initial value of 64 bytes all set to "0x01". Generated/derived internally by the SP 800-90A DRBG using the entropy input from the	Key is output from the module via user service API.	Plaintext in RAM	Zeroized using the key zeroization service.	These are variables used internally by the HMAC DRBG

[2] MCCi_crypto_asym_encryptor and MCCi_crypto_asym_encryptor_var_pad

Key/CSP Name	Key/CSP Type	Generation/Input	Output	Storage	Zeroization	Use
		entropy source. The seed is not generated, but input.				
Entropy Input	2048 bits	Entropy is passively loaded from an external source	N/A	Plaintext in RAM	Zeroized using the key zeroization service.	Used to instantiate or reseed the DRBG

Table 8 – Secret Keys, Private Keys and CSPs

3.4.1 Key Generation

The module contains an approved HMAC SHA256 SP800-90A approved DRBG.

Keys generated internally are generated by the HMAC SHA256 DRBG seeded by system entropy.

The module directly uses an output from the approved DRBG as a symmetric key or as a seed to be used in the asymmetric key generation. In accordance with IG D.12, the Cryptographic Module performs Cryptographic Key Generation (CKG) for symmetric and asymmetric keys per [133] (vendor affirmed).

2048 bits of entropy is used to seed the approved DRBG. The HMAC DRBG instantiation function requires 256 bits of full entropy to give a security strength of 256 bits. The entropy provided to the module falls into scenario 2(b) of IG 7.14. The module has no control over the quality of entropy that it receives but expects good quality entropy to be provided and that the 2048 bits of entropy used provides sufficient entropy for a security strength of 256 bits. The calling application shall use entropy sources that contains at least 256 bits of entropy.

3.4.2 Key Zeroization

All key material managed by the module can be zeroized by power-cycling the module's host platform. The module does not persistently store keys. As such, the calling application is responsible for parameters passed in and out of the module. The Operating Environment and the calling application are responsible for cleaning up temporary or ephemeral keys.

There are no user-accessible plaintext keys or CSPs in the module.

4. Roles, Services and Authentication

4.1 Roles

The Cryptographic Module implements both Crypto Officer role and User role. Roles are assumed implicitly upon accessing the associated services. Section 4.2 summarizes the services available to each role.

The module has a Single Operator Mode.

ROLE	DESCRIPTION
Crypto Officer	The administrator of the module having full configuration and key management privileges.
User	General User of the module

Table 9 – Roles

The service table below is related to the information needed for each role.

4.2 Services

The Approved services supported by the module and access rights within services accessible over the module's public interface are listed in the table below.

Service	Approved Security Functions	Keys and/or CSPs	Roles	Access rights to Keys and/or CSPs
MCCi_crypto_asym_encryptor	Provides RSA encryption and decryption.	Public Key Private Key	User	GWE GWE
MCCi_crypto_asym_encryptor_var_pad	Provides RSA encryption and decryption but allows the padding to be specified.	Public Key Private Key	User	GWE GWE
MCCi_crypto_asym_key_gen	Provides RSA key generation. The RSA keys are generated by the FIPS approved DRBG seeded by a non-approved NDRNG (outside of the module).	Public Key Private Key HMAC DRBG Key HMAC DRBG V	User	GR GR E E
MCCi_crypto_asym_key_gen2	Provides RSA key generation but allows the keys to be encoded in various formats. The RSA keys are generated by the FIPS approved DRBG seeded by a non-approved NDRNG (outside of the module).	Public Key Private Key HMAC DRBG Key HMAC DRBG V	User	GR GR E E
MCCi_rand_source	Generates symmetric keys and accepts seed	Data Key HMAC DRBG Key HMAC DRBG V	User	G E E

Service	Approved Security Functions	Keys and/or CSPs	Roles	Access rights to Keys and/or CSPs
	data to add to the random pool. Entropy data is used to seed the DRBG which is used to generate the symmetric keys.			
MCCi_crypto_enc_data	Utility class to hold encrypted data returned by the symmetric encryptor.	Data Key	User	RWE
MCCi_crypto_sym_encryptor	Provides symmetric algorithm encryption and decryption. Supports AES256 (CBC and CFB128 modes).	Data Key	User	RWE
MCCi_crypto_disk_algs	Provides access to a copy of the MFECCF*aa* symmetric disk encryption algorithms (statically linked into this, the MFECCF[32|64]*aa* module). Supports only FIPS AES 256 modes ECB, CBC, and CFB8.	Data Key	User	RWE
MCCi_crypto_digest_gen	Allows hashing of arbitrary data. Supports SHA1[3] and SHA256.		User	
MCCi_crypto_self_tests	Runs all the known answer tests		User	
Self-tests	The power-up software integrity test and Known answer tests in table 11 are run automatically when the module is loaded and started.		User	

[3] Using (non-compliant) SHA-1 results in a non-approved mode of operation.

Service	Approved Security Functions	Keys and/or CSPs	Roles	Access rights to Keys and/or CSPs
Show Status	Status is returned in response to individual service API calls and at the completion of the self-tests.		User	
Installation	The module is deployed as part of a Trellix product installation.		CO	
Uninstallation	The module is uninstalled during the uninstallation of the product that deployed the module.		CO	
Key Zeroization	Keys are zeroized by power-cycling the module's host platform.	All	CO	Z

Table 10 - Approved Services, Roles, and Access Rights

G or Generate: The module generates the CSP(s)
R or Read: The CSP is read from the module (e.g. the CSP is output)
W or Write: The CSP is updated or written to the module
E or Execute: Capability to execute or use the Critical Security Parameter
Z or Zeroize: The module zeroizes the CSP

4.3 Authentication

The module does not support operator authentication.

5. Self-tests

5.1 Power-On Self-Tests

On power up or reset, the module performs the self-tests described below. All KATs must be completed successfully prior to any other use of cryptography by the module. If one of the KATs fails, the module enters the Soft Failure error state.

The module performs the following power-up self-tests:

OBJECT	TEST
SHA-256	BSAFE Known answer test
HMAC-SHA-256	BSAFE Known answer test

OBJECT	TEST
AES-256	A separate encryption and decryption known answer test for each of the AES' implementations within the module: BSAFE: AES-CBC Encrypt and Decrypt, key length 256. AES-CFB128 Encrypt and Decrypt, key length 256. AES-ECB Encrypt and Decrypt, key length 256. MFE: AES-CBC Encrypt and Decrypt, key length 256. AES-CFB8 Encrypt and Decrypt, key length 256. AES-ECB Encrypt and Decrypt, key length 256.
HMAC DRBG	Known answer test – during DRBG instantiation. Section 11.3 health tests
Module software	HMAC SHA-256 Integrity Check[4]
RSA	Pairwise Consistency Test on key pair generation

Table 11 - Power-up self-tests

5.2 Conditional Self Tests

EVENT	TEST	CONSEQUENCE OF FAILURE
Module requests a random number from the FIPS Approved SP800-90 DRBG	A continuous random number generator test	Random number is not generated, and module enters an error state after the 10th consecutive error.
RSA key pair generation	Pairwise consistency test	Key is not generated, and module enters an error state.

Table 12 - Conditional self-tests

The following error message is returned in the event of a known answer test error: MCC_exception("Cryptographic module is in an error state and cannot be accessed", E_MCC_GEN_SELF_TEST_FAILED).

6. Physical Security

The Cryptographic Module is comprised of software only and thus does not claim any physical security.

7. Operational Environment

The Cryptographic Module operates under the operating environment(s) specified in Table 1.

8. Guidance and Secure Operation

This Cryptographic Module is built into Trellix products and is not publicly available to be installed as a stand-alone module. Initialization and guidance instructions are not applicable.

[4] Both the MFE SHA-256 and HMAC-SHA-256 KATs are covered by this test

9. Mitigation of other Attacks

The module does not mitigate any other attacks.

10. Design Assurance

Trellix employs industry standard best practices in the design, development, production, and maintenance of all its products, including the FIPS 140-2 module.

This includes the use of an industry standard configuration management system that is operated in accordance with the requirements of FIPS 140-2, such that each configuration item that forms part of the module is stored with a label corresponding to the version of the module and that the module and all its associated documentation can be regenerated from the configuration management system with reference to the relevant version number.

Design documentation for the module is maintained to provide clear and consistent information within the document hierarchy to enable transparent traceability between corresponding areas throughout the document hierarchy, for instance, between elements of this Cryptographic Module Security Policy (CMSP) and the design documentation.

Delivery of the Cryptographic Module to customers from the vendor is via the internet. When a customer purchases a license to use the Trellix product containing the Cryptographic Module software, they are issued with a grant number as part of the sales process. This is then used as a password to allow them to download the software that they have purchased. Once the Cryptographic Officer has downloaded the Cryptographic Module, it is his responsibility to ensure its secure delivery to the users that he is responsible for.

11. References and Standards

For more information on Trellix products please visit: https://www.trellix.com. For more information on NIST and the Cryptographic Module Validation Program (CMVP), please visit http://csrc.nist.gov/groups/STM/cmvp/index.html.

The following Standards are referred to in this Security Policy.

Abbreviation	Full Specification Name
[FIPS140-2]	*Security Requirements for Cryptographic Modules*, May 25, 2001
[IG]	*Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*
[133]	*National Institute of Standards and Technology, Recommendation for Cryptographic Key Generation, Special Publication 800-133rev2, June 2020.*
[135]	*National Institute of Standards and Technology, Recommendation for Existing Application-Specific Key Derivation Functions, Special Publication 800-135rev1, December 2011.*
[186]	National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4, July 2013.
[197]	*National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 26, 2001*
[38A]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001*
[38D]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, Special Publication 800- 38D, November 2007*
[198]	*National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198- 1, July, 2008*
[180-4]	*National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, August, 2015*
[67]	*National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Special Publication 800-67, May 2004*
[90A]	National Institute of Standards and Technology, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90A Rev 1, June 2015.

Table 13 - References

12. Acronyms and Definitions

The following Acronyms are referred to in this Security Policy:

Acronym	Definition
AES	Advanced Encryption Standard

Acronym	Definition
AES-NI	Advanced Encryption Standard New Instructions. Seven instructions for accelerating different sub-steps of the AES algorithm included in some Intel and AMD microprocessors.
API	Application Programming Interface
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher-Block Chaining
CCCS	Canadian Centre for Cyber Security
CMSP	Cryptographic Module Security Policy
CMVP	Crypto Module Validation Program
CO	Cryptographic Officer
CPU	Central Processing Unit
CSP	Critical Security Parameter
DLL	Dynamic Link Library
DLM	Dynamic Link Module (a type of DLL used in the Pre-boot environment)
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECDH	Elliptic Curve Diffie-Hellman
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
HMAC	key-Hashed Message Authentication Code
IG	Implementation Guidance
IV	Initialization Vector
KAT	Known Answer Test
MAC	Message Authentication Code
N/A	Not Applicable
NIST	National Institute of Standards and Technology
NDRNG	Non-Deterministic Random Number Generator
NVLAP	National Voluntary Laboratory Accreditation Program
OS	Operating System
Pre-boot environment	The operating environment of a GPC before the operating system is loaded
RSA	Public-key encryption technology developed by RSA Data Security, Inc.
SHA	Secure Hash Algorithms
SP	Security Policy

Table 14 - Acronyms