Trellix

Trellix Core Cryptographic AES Module

Document Version: 1.0

FIPS 140-3 Non-Proprietary Security Policy

Document prepared by:

http://www.lightshipsec.com

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This non-proprietary FIPS 140-3 Security Policy for the Trellix Core Cryptographic AES Module, version 6.0.0 describes how the module meets the security requirements specified in FIPS 140-3 for an overall security level 1 module and outlines the security rules and operating procedures required to maintain compliance.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Trellix Core Cryptographic AES Module is a Software multi-chip standalone cryptographic module. The module is a software library implementing general purpose cryptographic algorithms. The module provides AES encryption services to Trellix products and is packaged as a Microsoft Windows kernel mode device driver, a Microsoft Windows Dynamic Library (DLL), and a Unified Extensible Firmware Interface (UEFI) driver.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

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

Tested Operational Environment's Physical Perimeter (TOEPP):

The module's logical boundary is a software library. The physical boundary of the module is a General-Purpose Computer (GPC). Figure 1 shows the logical relationship of the Cryptographic Module to the TOEPP. The tested operational environments are listed in Table Tested Operational Environments - Software, Firmware, Hybrid, below.

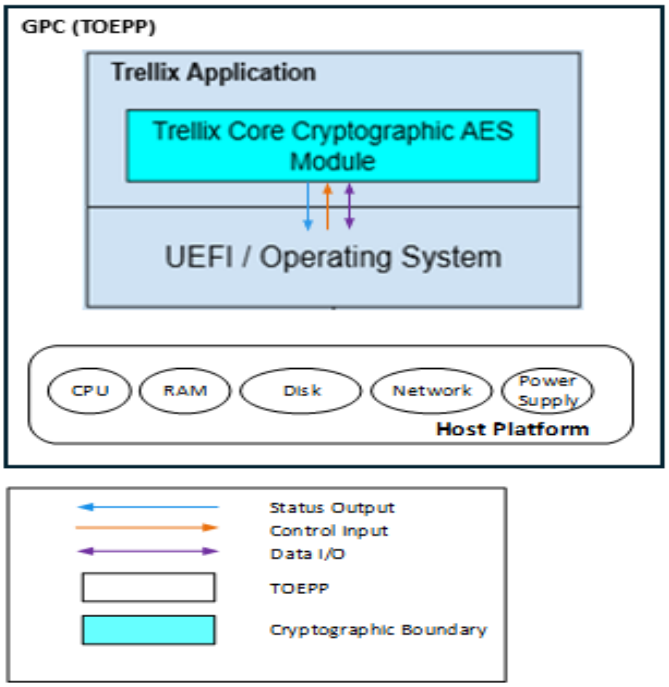

Figure 1: Module cryptographic boundary and TOEPP

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

Package or File Name	Software/ Firmware Version	Features	Integrity Test
MFECCFaa.sys	6.0.0	Microsoft Windows kernel driver (32-Bit and 64-Bit)	HMAC-SHA2-256

Package or File Name	Software/ Firmware Version	Features	Integrity Test
MFECCF32aa.dll	6.0.0	Microsoft Windows dynamic library (32-Bit)	HMAC-SHA2-256
MFECCF64aa.dll	6.0.0	Microsoft Windows dynamic library (64-Bit)	HMAC-SHA2-256
MFECCFaa.efi	6.0.0	Unified Extensible Firmware Interface (UEFI) driver (32-Bit and 64-Bit)	HMAC-SHA2-256

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Note: "*aa*" are alphanumeric product identifiers, which are denoted "DE" for Trellix Drive Encryption, "FF" for Trellix File & Removable Media Protection, and "MN" for Trellix Native Drive Encryption, the three main Trellix products that consume this cryptographic module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Windows 11 64-bit kernel	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	Yes		6.0.0
Windows 11 64-bit user mode	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	Yes		6.0.0
Windows 11 32-bit user mode	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	Yes		6.0.0
UEFI 64-bit Preboot	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	Yes		6.0.0
Windows 11 64-bit kernel	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	No		6.0.0
Windows 11 64-bit user mode	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	No		6.0.0
Windows 11 32-bit user mode	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	No		6.0.0
UEFI 64-bit Preboot	HP ZBook 17 G6	Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz	No		6.0.0

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

The module has been tested on the operational environments and platforms detailed in the table below:

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

The cryptographic module also operates in the following environment
• Windows Server 2016 onwards
• Windows 10 32-bit and 64-bit onwards
• With and without PAA processors

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.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Automatically entered whenever an approved service is requested	Approved	Return code ("1")

Table 4: Modes List and Description

Once these self-tests have completed successfully, the module transitions into the approved mode of operation. There are no other modes of operation implemented by the module.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7732	Direction - Decrypt, Encrypt Key Length - 256	SP 800-38A
AES-CFB8	A7732	Direction - Decrypt, Encrypt Key Length - 256	SP 800-38A
HMAC-SHA2-256	A7732	Key Length - Key Length: 256-448 Increment 8	FIPS 198-1
SHA2-256	A7732	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4

Table 5: Approved Algorithms

The Approved Algorithms tables above list the approved algorithms implemented by Trellix Core Cryptographic AES Module which are utilized by the module's callable services or internal functions.

Note: The AES-256 algorithm can run on processors with or without PAA capability. However, it will only use PAA instructions if run on AES-NI enabled processors.

Vendor-Affirmed Algorithms:

The module does not implement any vendor-affirmed algorithms.

Non-Approved, Allowed Algorithms:

The module does not implement any non-approved, allowed algorithms.

Non-Approved, Allowed Algorithms with No Security Claimed:

The module does not implement any non-approved, allowed algorithms with no security claimed.

Non-Approved, Not Allowed Algorithms:

The module does not implement any non-approved, not allowed algorithms.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Software Integrity Test	MAC	Integrity test for module software	Publication:FIPS 198-1	SHA2-256: (A7732) HMAC-SHA2-256: (A7732)
Encrypt Data	BC-UnAuth	Symmetric Encryption of data using AES	Publication:NIST SP 800-38A	AES-CBC: (A7732) AES-CFB8: (A7732)
Decrypt Data	BC-UnAuth	Symmetric Decryption of data using AES	Publication:NIST SP 800-38A	AES-CBC: (A7732) AES-CFB8: (A7732)

Table 6: Security Function Implementations

2.7 Algorithm Specific Information

The conditions for using the Module in the Approved mode of operation are:

1. The Trellix Core Cryptographic AES Module is a cryptographic library, and it is intended to be used with a calling application. The calling application is responsible for the usage of the primitives in the correct sequence including the IVs and sessions.
2. The keys used by the Trellix Core Cryptographic AES Module for cryptographic purposes are determined by the calling application. The calling application is required to provide keys in accordance with [140Drev2].
3. Data output is inhibited during self-tests, zeroization and error states.

2.8 RBG and Entropy

The module does not have an RBG or Entropy source within it

2.9 Key Generation

The module does not implement any key generation operation.

2.10 Key Establishment

The module does not implement any key establishment schemes.

2.11 Industry Protocols

The module does not implement any industry protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	Parameters passed to the module via API calls
N/A	Data Output	Data returned from the module via API calls
N/A	Control Input	API Calls and/or parameters passed to API calls
N/A	Status Output	Information received in response to API calls

Table 7: Ports and Interfaces

The Ports and Interfaces table above specifies the cryptographic module interfaces. The module provides all logical interfaces via Application Programming Interface (API) calls. These logical interfaces expose services that the User (i.e. application) may utilize directly. The module does not implement a control output interface.

The logical interfaces provided by the module are mapped onto FIPS 140-3 logical interfaces: data input, data output, control input, and status output.

The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not implement any authentication mechanisms. The operators implicitly assume an authorized role (or set of roles) based on the service selected.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None
User	Role	User	None

Table 8: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show status	Return module status	Global (completion of service)	Command ("get-fips-status")	Return code ("1")	None	Crypto Officer User
Show version	Return module name and version	Global (completion of service)	Command ("get-version")	Status Output ("Trellix Core Cryptographic AES Module-6.0.0")	None	Crypto Officer User
Perform self-tests on demand	Perform CASTs and Integrity tests	Global (completion of service)	Procedure/Reboot	Status Output (pass/fail)	Software Integrity Test	Crypto Officer
Zeroisation	Zeroise the SSPs stored temporarily in RAM	Global (completion of service)	Procedure/Reboot or using command (reset_key_context)	Return code ("0")	None	Crypto Officer - AES Symmetric Key: Z User - AES Symmetric Key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
AES Encryption	Perform Symmetric Key Encryption	Global (completion of service)	Plain Text	Cipher Text	Encrypt Data	User - AES Symmetric Key: W,E
AES Decryption	Perform Symmetric Key Decryption	Global (completion of service)	Cipher Text	Plain Text	Decrypt Data	User - AES Symmetric Key: W,E

Table 9: Approved Services

The abbreviations of the access rights to SSPs have the following interpretation:

G = Generate: The module generates or derives the SSP.

R = Read: The SSP is read from the module (e.g., the SSP is output).

W = Write: The SSP is updated, imported, or written to the module.

E = Execute: The module uses the SSP in performing a cryptographic operation.

Z = Zeroise: The module zeroises the SSP.

The module provides approved services to the operator who assumes the role as defined in this document.

The approved services defined in this section implement the security function implementations defined in section 2.6 of this document.

4.4 Non-Approved Services

The module does not implement any non-approved services.

4.5 External Software/Firmware Loaded

The module does not allow the loading of external software or firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

The approved software integrity technique is implemented by the cryptographic module itself as part of the pre-operational self-test, which is executed when the module is initialized.

Software integrity test:

The entire module software is covered with an approved integrity technique (HMAC-SHA2-256) with a 256-bit key which is implemented in the module itself. If the calculated integrity value does not match the reference value embedded into the software, the module enters the Error state and terminates execution of module.

5.2 Initiate on Demand

The conditional algorithm self-tests are run as part of the pre-operational self-test in addition to the firmware integrity test. The operator can initiate the self-tests on demand by power-cycling the host platform (TOEPP).

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

The operator should confirm that the module is operating in the approved mode by checking for the approved mode indicator per the instructions in Section 11.2 of this document.

7 Physical Security

The module is software only, and therefore this section is not applicable.

8 Non-Invasive Security

The module does not implement any security mechanisms which protect against non-invasive attacks.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Temporary storage for SSPs used by the module as a part of service execution	Dynamic

Table 10: Storage Areas

The module stores keys and input/output data temporarily in volatile memory (RAM). The module does not store keys or data persistently.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API input parameters	Calling application	RAM	Plaintext	Manual	Electronic	Encrypt Data
API output parameters	RAM	Calling application	Plaintext	Manual	Electronic	Decrypt Data

Table 11: SSP Input-Output Methods

SSPs are only input from the calling applications within the module's TOEPP. This method is categorized as manual distribution, electronic entry ("CM Software from App via TOEPP Path").

9.3 SSP Zeroisation Methods

Zeroization Method	Description	Rationale	Operator Initiation
Power Cycle/ Re-instantiate module/ API call	Reboot the host platform or use the reset_key_context to zeroise the keys.	Keys are procedurally zeroized by rebooting the host platform, which is acceptable at Software level 1 or by using reset_key_context.	Crypto Officer reboots the host platform or restarts the application for User Mode.

Table 12: SSP Zeroization Methods

SSPs are zeroised procedurally by power-cycling the host platform.

9.4 SSPs

The following table summarizes the Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented:

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Symmetric Key	Encryption and Decryption	256 bits - 256 bits	Symmetric Key - CSP			Encrypt Data Decrypt Data

Table 13: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Symmetric Key	API input parameters API output parameters	RAM:Plaintext	Until zeroised	Power Cycle/ Re-instantiate module/ API call	

Table 14: SSP Table 2

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A7732)	HMAC-SHA2-256	KAT	SW/FW Integrity	Logged into windows registry ("Passed" OR "Failed").	Software component Integrity test using an approved keyed hash (HMAC-SHA2-256)

Table 15: Pre-Operational Self-Tests

The startup integrity test is performed upon the module.

As the modules do not implement bypass capability or any FIPS-defined critical functions, no additional pre-operational self-tests are required.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A7732)	256 bits	KAT	CAST	Logged into windows registry ("Passed" OR "Failed").	Encryption	Before first operational use of encryption service. / Immediately when the module enters the approved mode of operation.
AES-CFB8 (A7732)	256 bits	KAT	CAST	Logged into windows registry ("Passed" OR "Failed").	Encryption	Before first operational use of encryption service. / Immediately when the module enters the approved mode of operation.
HMAC-SHA2-256 (A7732)	HMAC-SHA2-256	KAT	CAST	Logged into windows registry ("Passed" OR "Failed").	Keyed hash compare KAT	Before first operational use of HMAC-SHA2-256 which includes the integrity check.
SHA2-256 (A7732)	SHA2-256	KAT	CAST	Logged into windows registry ("Passed"	Hash compare KAT	Before first operational use of SHA2-256

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				OR "Failed").		
AES-CBC Decrypt (A7732)	256 bits	KAT	CAST	Logged into windows registry ("Passed" OR "Failed").	Decryption	Before first operational use of decryption service. / Immediately when the module enters the approved mode of operation.
AES-CFB8 Decrypt (A7732)	256 bits	KAT	CAST	Logged into windows registry ("Passed" OR "Failed").	Decryption	Before first operational use of decryption service. / Immediately when the module enters the approved mode of operation.

Table 16: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A7732)	KAT	SW/FW Integrity	Upon module startup	Manual (reboot host platform)

Table 17: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A7732)	KAT	CAST	On Demand / On Startup	Manually, by reboot of host device
AES-CFB8 (A7732)	KAT	CAST	On Demand / On Startup	Manually, by reboot of host device
HMAC-SHA2-256 (A7732)	KAT	CAST	On Demand / On Startup	Manually, by reboot of host device
SHA2-256 (A7732)	KAT	CAST	On Demand / On Startup	Manually, by reboot of host device

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC Decrypt (A7732)	KAT	CAST	On Demand / On Startup	Manually, by reboot of host device
AES-CFB8 Decrypt (A7732)	KAT	CAST	On Demand / On Startup	Manually, by reboot of host device

Table 18: Conditional Periodic Information

The operator can perform pre-operational and conditional self-tests on demand by power-cycling the host platform.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	Module terminates operation, Host device must be restarted for kernel and UEFI drivers. For the User Mode, the application using the module needs to be restarted for recovery.	Module fails any Pre-Operational Self-Tests or Conditional Self-Tests or Software Integrity Test.	Reboot host platform for Kernel and UEFI drivers. Restart the applications for User Mode.	Module terminates operation

Table 19: Error States

When the module fails any self-test, the module will immediately become unavailable to the host system. When in the error state, the module interfaces will not be available and therefore, all cryptographic operation is inhibited.

10.5 Operator Initiation of Self-Tests

The operator can perform the conditional self-tests on demand by power-cycling the host platform.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is installed along with Trellix Data Encryption products when they are installed in FIPS mode.

11.2 Administrator Guidance

The Crypto Officer should ensure that the module is running in the approved mode of operation before use. This can be determined by checking the device log to confirm the output of the 'Show Module's Versioning Information' and 'Show Status' services as listed in the 'Approved Services' section of this document.

11.3 Non-Administrator Guidance

In case the module's power is lost and then restored, the key used for AES encryption and decryption shall be provided to the module again.

12 Mitigation of Other Attacks

The module does not implement any security mechanisms which protect against other attacks.