Microsoft Corporation

Windows OS Loader

FIPS 140-3 Non-Proprietary Security Policy Document

Microsoft Windows 11 version 22H2
(Pro, Enterprise, IoT Enterprise, Education, and Home Editions)

Microsoft Windows Server 2022
(Standard and Datacenter Editions)

Prepared By	Microsoft Corporation One Microsoft Way Redmond, WA 98052-6399
Document Version Number	1.0
Updated On	May 4, 2026

COPYRIGHT AND DISCLAIMER

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.

Microsoft, Active Directory, Azure, Visual Basic, Visual Studio, Windows, the Windows logo, Windows NT, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

1 General 6
1.1 Overview 6
1.2 Security Levels 6
2 Cryptographic Module Specification 6
2.1 Description 6
2.2 Tested and Vendor Affirmed Module Version and Identification 9
2.3 Excluded Components 11
2.4 Modes of Operation 11
2.5 Algorithms 11
2.6 Security Function Implementations 14
2.7 Algorithm Specific Information 17
2.7.1 FIPS 186-4 and 186-5 17
2.7.2 Legacy RSA Signature Verification 17
2.7.3 AES-XTS 17
2.7.4 AES-GCM 17
2.8 RBG and Entropy 17
2.9 Key Generation 18
2.10 Key Establishment 18
2.11 Industry Protocols 18
3 Cryptographic Module Interfaces 18
3.1 Ports and Interfaces 18
4 Roles, Services, and Authentication 19
4.1 Authentication Methods 19
4.2 Roles 19
4.3 Approved Services 20
4.4 Non-Approved Services 24
4.5 External Software/Firmware Loaded 24
5 Software/Firmware Security 24
5.1 Integrity Techniques 24
5.2 Initiate on Demand 26
6 Operational Environment 27
6.1 Operational Environment Type and Requirements 27
7 Physical Security 27
7.1 Mechanisms and Actions Required 27
8 Non-Invasive Security 27
9 Sensitive Security Parameters Management 27

9.1 Storage Areas 27
9.2 SSP Input-Output Methods 27
9.3 SSP Zeroization Methods 28
9.4 SSPs 29
9.5 Transitions 31
10 Self-Tests 31
10.1 Pre-Operational Self-Tests 31
10.2 Conditional Self-Tests 31
10.3 Periodic Self-Test Information 35
10.4 Error States 40
10.5 Operator Initiation of Self-Tests 40
11 Life-Cycle Assurance 41
11.1 Installation, Initialization, and Startup Procedures 41
11.2 Administrator Guidance 43
11.2.1 Verifying the Installed Windows Version 43
11.2.2 Verifying the Cryptographic Module Version and its Signature 43
11.3 Non-Administrator Guidance 45
11.4 Design and Rules 45
12 Mitigation of Other Attacks 45
12.1 Attack List 45
13 References 45

List of Tables

Table 1: Security Levels 6
Table 2: Module Software Components 7
Table 3: CPU Photographs 9
Table 4: Version Information 9
Table 5: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) 10
Table 6: Tested Module Identification – Hybrid Disjoint Hardware 10
Table 7: Tested Operational Environments - Software, Firmware, Hybrid 10
Table 8: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid 11
Table 9: Modes List and Description 11
Table 10: Approved Algorithms - 12
Table 11: Approved Algorithms - Existing Validated Module [EVM] 12
Table 12: Security Function Implementations 16
Table 13: Random Bit Generator (RBG) Certificates 18
Table 14: Entropy Certificates 18
Table 15: Entropy Sources 18
Table 16: Ports and Interfaces 19
Table 17: Roles 19
Table 18: Approved Services 23

Table 19: Existing Validated Module Details 25
Table 20: Storage Areas 27
Table 21: SSP Input-Output Methods 28
Table 22: SSP Zeroization Methods 28
Table 23: SSP Table 1 29
Table 24: SSP Table 2 30
Table 25: Pre-Operational Self-Tests 31
Table 26: Conditional Self-Tests 35
Table 27: Pre-Operational Periodic Information 36
Table 28: Conditional Periodic Information 40
Table 29: Error States 40
Table 30: Mitigation of Other Attacks 45

List of Figures

Figure 1: Module Boundary Diagram 7
Figure 2: Integrity Chain of Trust 26
Figure 3: Finite State Model 42

1 General

1.1 Overview

The Windows OS Loader (the “module”) is the component that loads the operating system kernel (ntoskrnl.exe) and other boot stage binary image files. The module implements approved cryptographic algorithms. This FIPS 140-3 Security Policy contains a specification of the rules under which the module must operate and describes how the module meets the requirements specified in Federal Information Processing Standards Publication 140-3 (FIPS PUB 140-3) and International Standard ISO/IEC 19790:2012 (Information technology – Security techniques – Security requirements for cryptographic modules). This document is intended for the FIPS 140-3 testing lab, the Cryptographic Module Validation Program (CMVP), and administrators and users of the module.

1.2 Security Levels

The overall security rating for the module is level 1. The table below lists the security levels of individual clauses for this validation. As a software-hybrid module executing in a modifiable environment, non-invasive security requirements are not applicable.

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	1
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	1
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Windows OS Loader module is a multi-chip standalone software-hybrid cryptographic module that consists of the binary WINLOAD.EFI. The Windows OS Loader loads the operating system kernel (ntoskrnl.exe) and other boot stage binary image files. The module is a part of BitLocker Drive Encryption, which is a data protection feature of the Windows operating system that encrypts data on a storage volume.

Module Type: Software-hybrid

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The software-hybrid cryptographic boundary for the Windows OS Loader consists of disjoint software and hardware components within the same physical perimeter of the host platform. The module's software component is the binary listed in the following table, and its hardware component is the CPU running on the host platform.

Software Component	Description
WINLOAD.EFI	Binary file that contains the module.

Table 2: Module Software Components

Tested Operational Environment's Physical Perimeter (TOEPP):

The Tested Operational Environment's Physical Perimeter (TOEPP) is the physical perimeter of the computer that contains the module.

The following block diagram illustrates the module's components, physical perimeter (TOEPP), and cryptographic boundary. The cryptographic boundary of the module is the module software component, WINLOAD.EFI. The WINLOAD.EFI binary is loaded from the OS volume in physical storage and executes in the computer memory. The control input, data input / output, and status output of the module exist within the computer memory as well.

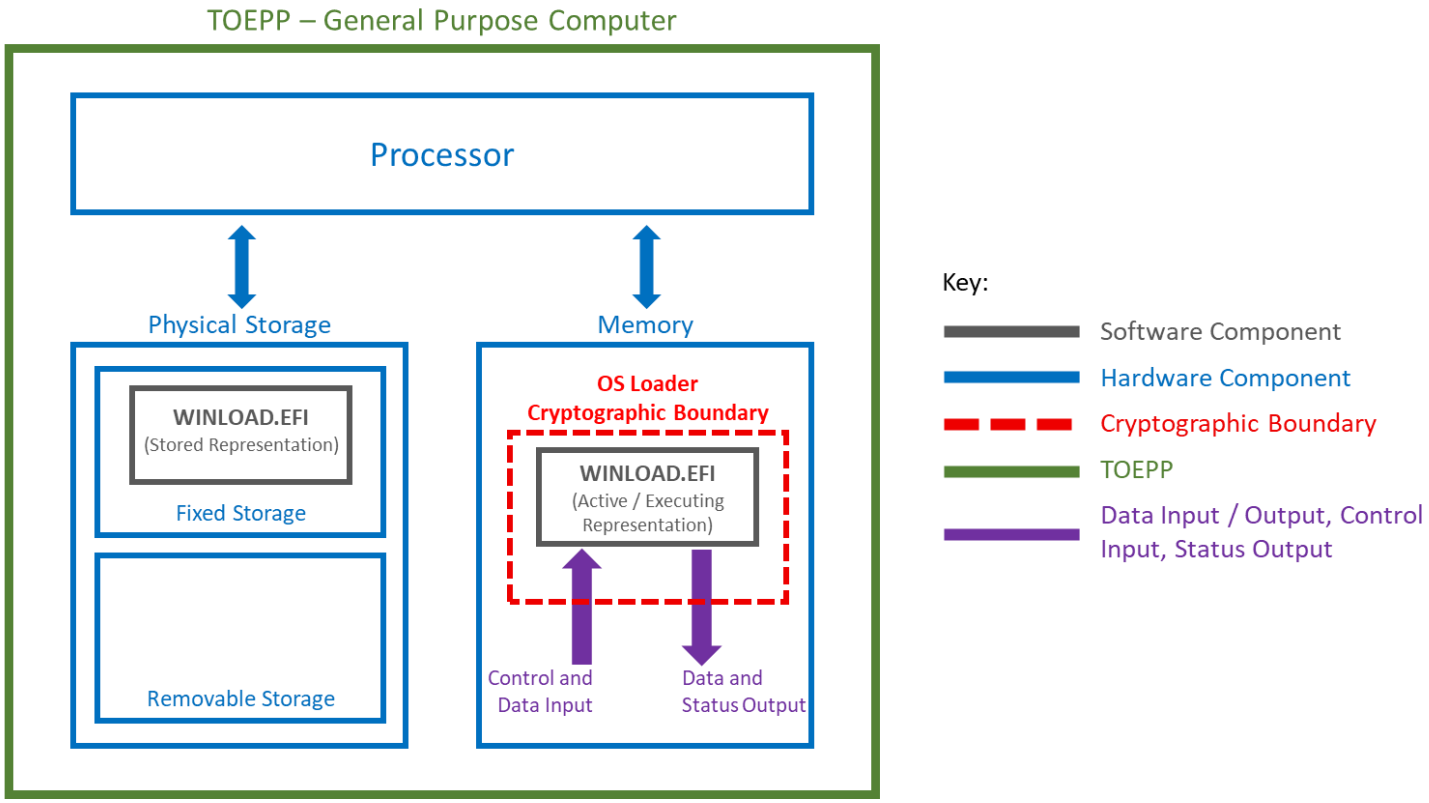
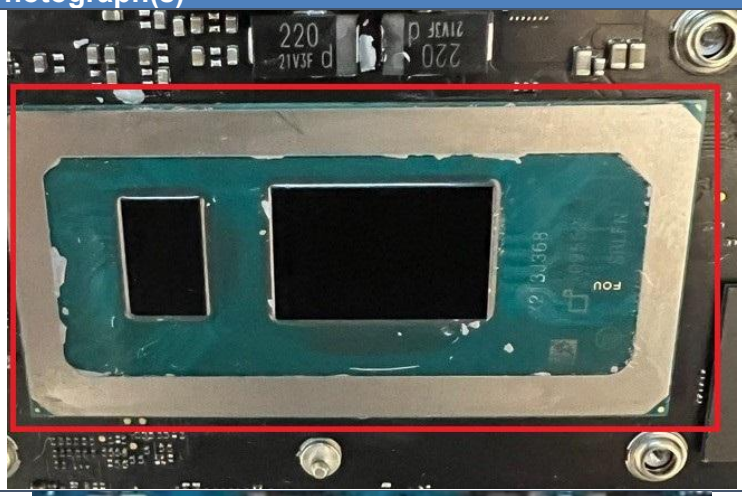
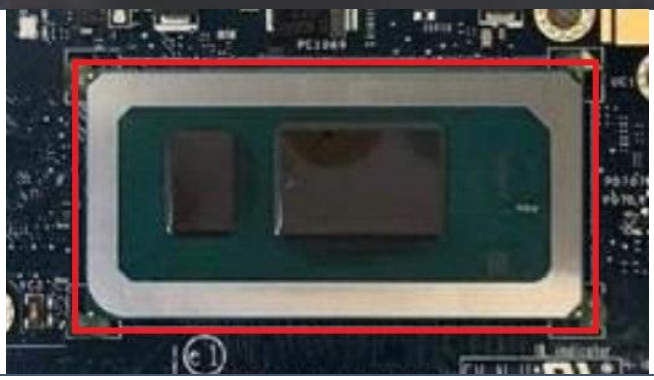


Figure 1: Module Boundary Diagram

The following table includes a photograph of the CPU of each computer listed in Section 2.2 Tested and Vendor Affirmed Module Version and Identification. The processor is highlighted by a red box for clarity. For laptop devices, the processor is shown as integrated into the motherboard. For server devices, the processor is shown both independently and as installed in the computer with its integral heat sink.

CPU	Photograph(s)
12th Gen Intel Core i7-1265U (Microsoft Surface Laptop 5)	
11th Gen Intel Core i5-11500H (HP ZBook Power G8)	
11th Gen Intel Core i7-1185G7 (Dell Latitude 7420)	

CPU	Photograph(s)
Intel Xeon Gold 6130 (Dell PowerEdge R640)	

Table 3: CPU Photographs

2.2 Tested and Vendor Affirmed Module Version and Identification

This validation includes the following Windows products and versions, each of which can be identified by its build number. The cryptographic module is a distinct implementation in each product build and for each processor architecture. Some Windows products may be installed as different editions; however, the cryptographic module is the same implementation in different editions of the same product.

Windows Product	Build	Edition(s) in Scope
Windows 11 version 22H2	10.0.22621.1	Enterprise Edition Home Edition IoT Enterprise Edition Pro Edition Education Edition
Windows Server 2022	10.0.20348.1668 (including the March 2023 updates)	Standard Edition Datacenter Edition

Table 4: Version Information

Tested Module Identification – Hardware:

N/A for this module.

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
WINLOAD.EFI (Windows 11 version 22H2)	Windows 11 version 22H2, build 10.0.22621.1	N/A	Yes
WINLOAD.EFI (Windows Server 2022)	Windows Server 2022, build 10.0.20348.1668	N/A	Yes

Table 5: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
Microsoft Surface Laptop 5	12th Gen Intel Core i7-1265U	N/A	12th Gen Intel Core i7-1265U	N/A
HP ZBook Power G8	11th Gen Intel Core i5-11500H	N/A	11th Gen Intel Core i5-11500H	N/A
Dell Latitude 7420	11th Gen Intel Core i7-1185G7	N/A	11th Gen Intel Core i7-1185G7	N/A
Dell PowerEdge R640	Intel Xeon Gold 6130	N/A	Intel Xeon Gold 6130	N/A

Table 6: Tested Module Identification – Hybrid Disjoint Hardware

Tested Operational Environments - Software, Firmware, Hybrid:

The operational environment for the module is the Windows operating system running on a supported hardware platform, as listed in the table below. All hardware platforms in the table below are 64-bit Intel architecture. The tested operational environments provide Processor Algorithm Acceleration (PAA) in the form of the Advanced Encryption Standard New Instructions (AES-NI).

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Windows 11 version 22H2, Enterprise Edition	Microsoft Surface Laptop 5	12th Gen Intel Core i7-1265U	Yes	N/A	Windows 11 version 22H2, build 10.0.22621.1
Windows 11 version 22H2, Home Edition	Microsoft Surface Laptop 5	12th Gen Intel Core i7-1265U	Yes	N/A	Windows 11 version 22H2, build 10.0.22621.1
Windows 11 version 22H2, IoT Enterprise Edition	Microsoft Surface Laptop 5	12th Gen Intel Core i7-1265U	Yes	N/A	Windows 11 version 22H2, build 10.0.22621.1
Windows 11 version 22H2, Pro Edition	HP ZBook Power G8	11th Gen Intel Core i5-11500H	Yes	N/A	Windows 11 version 22H2, build 10.0.22621.1
Windows 11 version 22H2, Education Edition	Dell Latitude 7420	11th Gen Intel Core i7-1185G7	Yes	N/A	Windows 11 version 22H2, build 10.0.22621.1
Windows Server 2022 Standard, including the March 2023 Updates	Dell PowerEdge R640	Intel Xeon Gold 6130	Yes	N/A	Windows Server 2022, build 10.0.20348.1668
Windows Server 2022 Datacenter, including the March 2023 Updates	Dell PowerEdge R640	Intel Xeon Gold 6130	Yes	N/A	Windows Server 2022, build 10.0.20348.1668

Table 7: Tested Operational Environments - Software, Firmware, Hybrid

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

Operating System	Hardware Platform
Any Microsoft operating system which is a relabeled version of the operating systems listed in Section 2.2.	Any UEFI-based x64 computer.

Table 8: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

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

The following caveat applies when operating the module on any vendor-affirmed operational environment:

No assurance of the minimum strength of generated SSPs (e.g., keys).

2.3 Excluded Components

No components within the cryptographic boundary are claimed as excluded.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode	Normal operation of the computer, Windows OS, and module. This is the only mode claimed by the module.	Approved	N/A

Table 9: Modes List and Description

2.5 Algorithms

Approved Algorithms:

The tables below list the approved algorithms used in the module. The module may not use some of the capabilities described in each CAVP certificate. As the module has separate CAVP certificates for Windows 11 and Windows Server 2022, each approved algorithm is listed twice, with CAVP certificates #A4008 and #A3767 for Windows 11 and CAVP certificates #A4009 and #A3768 for Windows Server 2022. See Section 13 References for links to the standards referenced in the table below.

The Windows OS Loader cryptographic module relies on some functionality that is implemented in the Windows Boot Manager cryptographic module. Both modules are installed together as described in Section 11 Life-Cycle Assurance. FIPS 140-3 deems the Windows OS Loader module as binding to the Windows Boot Manager module (certificate #5404), which is referred to as an Existing Validated Module (EVM) in this document. See Section 5.1 Integrity Techniques for more information on the dependencies between Windows modules.

Table 10 below lists the approved algorithms in the Windows OS Loader module. Table 11 below lists the approved algorithms in the [EVM] Windows Boot Manager module that are used by the Windows OS Loader module.

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4008	Key Length - 128	SP 800-38A
AES-CBC	A4009	Key Length - 128	SP 800-38A
AES-CCM	A4008	Key Length - 128	SP 800-38C
AES-CCM	A4009	Key Length - 128	SP 800-38C
AES-GCM	A4008	Direction - Decrypt Key Length - 128	SP 800-38D
AES-GCM	A4009	Direction - Decrypt Key Length - 128	SP 800-38D
AES-XTS Testing Revision 2.0	A4008	Key Length - 128	SP 800-38E
AES-XTS Testing Revision 2.0	A4009	Key Length - 128	SP 800-38E
Counter DRBG	A4008	Prediction Resistance - No Mode - AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
Counter DRBG	A4009	Prediction Resistance - No Mode - AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
RSA SigVer (FIPS186-4)	A3767	Signature Type - PKCS 1.5 Modulo - 1024, 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A3768	Signature Type - PKCS 1.5 Modulo - 1024, 2048, 3072, 4096	FIPS 186-4
SHA-1	A4008	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA-1	A4009	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A4008	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A4009	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A4008	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A4009	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A4008	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A4009	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 10: Approved Algorithms -

Existing Validated Module [EVM]

Algorithm	CAVP Cert	Properties	Reference
RSA SigVer (FIPS186-4)	A3767	Signature Type - PKCS 1.5 Modulo - 1024, 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A3768	Signature Type - PKCS 1.5 Modulo - 1024, 2048, 3072, 4096	FIPS 186-4
SHA2-256	A4008	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A4009	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 11: Approved Algorithms - Existing Validated Module [EVM]

Vendor-Affirmed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

The table below lists the module's Security Function Implementations. The Algorithms column presents algorithm and key size information for each CAVP certificate listed in Section 2.5 Algorithms. Blank cells are either not applicable or optional according to the CMVP. As the module has separate CAVP certificates for Windows 11 and Windows Server 2022, each approved algorithm may be listed twice in the Algorithms column.

Name	Type	Description	Properties	Algorithms
BC1	BC-Auth BC-UnAuth	Symmetric block cipher decryption used by the Load the OS (Trusted Boot) service to decrypt data on the OS volume.		AES-CBC: (A4008, A4009) Key Length: 128 bits, 256 bits AES-XTS Testing Revision 2.0: (A4008, A4009) Key Length: 128 bits, 256 bits AES-CCM: (A4008, A4009) Key Length: 256 bits AES-GCM: (A4008, A4009) Key Length: 256 bits
DigSig-Legacy	DigSig-SigVer	RSA signature verification used by the module services (Secure boot, Unlocking the OS, and Loading and Verifying OS Loader) for integrity verification.		RSA SigVer (FIPS186-4): (A3767, A3768) RSA Modulus: 1024 bits (For legacy signature verification only) SHA-1: (A4008, A4009) SHA size: 160 bits (For legacy signature verification only)

Name	Type	Description	Properties	Algorithms
DigSig1	DigSig-SigVer	RSA signature verification used by the Load the OS (Trusted Boot) service to verify integrity of the components mentioned in Integrity Chain of Trust. SHA secure hash functions support this by verifying the RSA signature.		RSA SigVer (FIPS186-4): (A3768) RSA Moduli: 2048, 3072, and 4096 bits. RSA SigVer (FIPS186-4): (A3767) RSA moduli: 2048, 3072, and 4096 bits. SHA2-256: (A4008) SHA Size: 256 bits SHA2-256: (A4009) SHA size: 256 bits SHA2-384: (A4008, A4009) SHA size: 384 bits SHA2-512: (A4008, A4009) SHA size: 512 bits
DigSig2	DigSig-SigVer	RSA signature verification used for the module pre-operational self-test only, executed by the [EVM] Boot Manager module (IG 1.A Documentation Requirements 5). SHA secure hash functions support this by verifying the RSA signature.		RSA SigVer (FIPS186-4): (A3767, A3768) RSA Moduli: 2048, 3072, and 4096 bits. SHA2-256: (A4008, A4009) SHA Size: 256 bits SHA2-384: (A4008, A4009) SHA Size: 384 bits SHA2-512: (A4008, A4009) SHA Size: 512 bits
DRBG1	DRBG	Used by the Load the OS (Trusted Boot) service to generate output data as input for when the Kernel Mode Cryptographic Primitives Library module (CNG.sys) initializes its DRBG.		Counter DRBG: (A4008) Block cipher: AES-256 Counter DRBG: (A4009) Block cipher: AES-256 AES-CBC: (A4008, A4009)
ENT1	ENT-P	Entropy source used by the Collecting Initial Entropy service to initialize the DRBG (DRBG1), providing entropy to the deterministic random number generator.		

Name	Type	Description	Properties	Algorithms
SHS1	SHA	Secure hash functions used by the Measured Boot service to validate the health of early boot stage components.		SHA2-256: (A4008, A4009) SHA size: 256 bits SHA2-384: (A4008, A4009) SHA size: 384 bits SHA2-512: (A4008, A4009) SHA size: 512 bits

Table 12: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 FIPS 186-4 and 186-5

The module claims compliance with FIPS 186-5 for RSA signature verification, although the CAVP testing for RSA was completed against FIPS 186-4. Because the FIPS 186-4 RSA CAVP tests are mathematically identical to the FIPS 186-5 RSA CAVP tests, the module can claim a FIPS 186-5 compliance for these tests. The scope of FIPS 186-4 testing complies with IG C.K. Additional Comment 3. Although FIPS 186-4 has been superseded by FIPS 186-5, algorithms implemented under FIPS 186-4 remain approved for use under NIST SP 800-131A Rev. 2.

2.7.2 Legacy RSA Signature Verification

1. RSA signature verification using SHA-1 is used for legacy signature verification only.
2. 1024-bit RSA key is used for legacy signature verification only.
3. Algorithms designated as “Legacy” can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

2.7.3 AES-XTS

AES-XTS is approved only for storage applications such as BitLocker. The length of data encrypted does not exceed 2^{20} AES blocks. Key 1 and Key 2 are generated independently, as required by IG C.I., and the two keys are explicitly checked to ensure that they are not equal before use.

2.7.4 AES-GCM

AES-GCM is used only for decryption. The module does not generate AES-GCM initialization vectors (IVs).

2.8 RBG and Entropy

The Windows OS Loader entropy source consists of a Physical (P) noise source and conditioning chain contained in an Intel processor, supplemented with a SHA2-512 vetted conditioning component implemented in software by Microsoft. The Intel portion of the entropy source is accessed via the RDSEED machine instruction, where the Intel RDSEED component is outside the cryptographic module boundary but within the TOEPP. The software portion is implemented within the Windows OS Loader (winload.efi). The health tests for the entropy source are part of the continuous health tests in the Intel processor and satisfy the requirements of SP800-90B Section 4.

The RDSEED machine instruction provides 1024 bits of data to the SHA2-512 vetted conditioning component. Section 7 of the entropy source’s Public Use Document specifies the output of the vetted conditioning component as having full entropy. See section 13 References for links to the Public Use Documents associated with the entropy certificates listed in the table below. The Windows OS Loader AES-CTR-256 DRBG is seeded with 512 bits of output from the entropy source and therefore is seeded with full entropy before generating random strings.

The module's DRBG is part of an SP800-90C compliant RBGC construction, as listed in the RBG certificate table below. The DRBGs in the chain are seeded and instantiated with 256-bits of security strength.

#	Vendor Name	Certificate Number
1	Microsoft Corporation	#G1

Table 13: Random Bit Generator (RBG) Certificates

Cert Number	Vendor Name
E189	Microsoft Corporation
E216	Microsoft Corporation

Table 14: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Windows OS Loader Entropy Source for Windows 11	Physical	1) Microsoft Windows 11 version 22H2 running on a 12th Gen Intel Core i7-1265U with AES-NI; 2) Microsoft Windows 11 version 22H2 running on an 11th Gen Intel i7-1185G7 with AES-NI; 3) Microsoft Windows 11 version 22H2 running on an 11th Gen Intel i5-11500H with AES-NI	512 bits	512-bit	CBC-MAC with AES-128 (#A2873, #A2668, #A1791); SHA2-512 (#A4008, #A4009)
Windows OS Loader Entropy Source for Windows Server 2022	Physical	1) Microsoft Windows Server 2022 running on an Intel Xeon Gold 6130 with AES-NI;	512 bits	512-bit	CBC-MAC with AES-128 (#A2873, #A2668, #A1791); SHA2-512 (#A4008, #A4009)

Table 15: Entropy Sources

2.9 Key Generation

N/A because the module does not generate cryptographic keys.

2.10 Key Establishment

N/A because the module does not perform key establishment.

2.11 Industry Protocols

N/A as none are claimed.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

As a software-hybrid module, the module has no physical ports of its own. The physical ports of the module are interpreted as those on the underlying hardware platform and control of them is outside the scope of the module.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	The Data Input Interface is represented by the BlFileReadEx function and the BlDeviceRead function. BlFileReadEx is responsible for reading the binary data of unverified components from the computer hard drive. In addition, the BitLocker Full Volume Encryption Key (FVEK) can also be entered into the module over the module's data input interface. BlDeviceRead is responsible for reading data directly from devices.
N/A	Data Output	The Data Output Interface is represented by the AhCreateLoadOptionsString function. OslArchTransferToKernel is responsible for transferring the execution from Windows OS Loader to the initial execution point of the Windows kernel. Data exits the module in the form of the initial instruction address of the Windows kernel. Data exits the module from the AhCreateLoadOptionsString function in the form of boot application parameters passed to the Windows kernel and, for computers with a TPM running in Virtual Secure Mode, in the form of keys sealed by the TPM.
N/A	Control Input	The Windows OS Loader Control Input Interface is the set of internal functions responsible for intercepting control input. These functions are: OslMain (WINLOAD) - Receives and parses the Boot Application parameters, which are passed to the module when execution is passed from Boot Manager. BlBdInitialize - Reads the system status to determine if a boot debugger is attached. BlInitializeLibrary - Parses Boot Application parameters. BlXmiRead - Reads the operator selection from the Windows OS Loader user interface.
N/A	Status Output	The Status Output Interface is the BlXmiWrite function that is responsible for displaying any integrity verification errors to the display. The Status Output Interface is also defined as the BlLogData responsible for writing the name of the corrupt driver to the bootlog.
N/A	Power	N/A

Table 16: Ports and Interfaces

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

4.2 Roles

The module claims a single role, Cryptographic Officer (CO). All services are accessible by this role.

Name	Type	Operator Type	Authentication Methods
Cryptographic Officer (CO)	Role	CO	None

Table 17: Roles

4.3 Approved Services

The module provides approved services only. The indicator for each service is the successful completion of the service. The table below provides additional indicator details to enable the operator to verify each service's successful completion.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Collect Initial Entropy	The module uses an SP 800-90B entropy source. At startup, this service collects the initial entropy for the system.	Initial entropy is collected and the Windows boot process continues.	This service is fully automatic and occurs after self-tests pass and, if the system volume is encrypted, after decryption of the required system volume sectors.	Initial entropy is collected for DRBG initialization.	ENT1	Cryptographic Officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Load the OS (Trusted Boot)	Validates the integrity of the following OS components and then loads them: the Windows OS kernel (ntoskrnl.exe), the Windows hypervisor (hvix64.exe), and other boot stage binary image files, including Code Integrity (ci.dll), Secure Kernel Code Integrity (skci.dll), and the Kernel Mode Cryptographic Primitives Library (cng.sys). This service also initializes the DRBG using the entropy input string, and then uses the DRBG output to provide random bits as input to the Kernel Mode Cryptographic Primitives Library (cng.sys).	The Windows boot process continues after integrity checks are complete.	This service is fully automatic after Boot Manager has loaded the Windows OS Loader.	After the verified kernel and boot stage binary image files are loaded, Windows OS Loader passes the execution control to the NT operating system kernel, and the Virtual Secure Mode kernel, and it terminates its own execution. If the integrity of any module is not verified, Windows OS Loader does not transfer the execution to the kernel and displays the boot failure page.	BC1 DigSig-Legacy DigSig1 DRBG1 SHS1	Cryptographic Officer (CO) - Microsoft Root Certificate Authority Public Key (This is not an SSP): E - SHA Hashes (This is not an SSP): E - Full Volume Encryption Key (FVEK): W,E - AES-CTR DRBG Entropy Input String and Nonce: G,E - AES-CTR DRBG Seed: G,E - AES-CTR DRBG V: G,E - AES-CTR DRBG Key: G,E
Measured Boot	Validates the early boot stage components using secure hash functions. Collects boot process log measurements from Windows and the computer's firmware, which can be sent to a trusted remote server for objective assessment of the health of early boot stage components.	Measured boot data is output, e.g. to a log file.	This service is fully automatic after Measured Boot has been enabled.	Measured boot data is made available to a remote trusted attestation server.	SHS1	Cryptographic Officer (CO) - SHA Hashes (This is not an SSP): W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Perform Cryptographic Algorithm Self-Tests	The module provides a power-up self-test service that is automatically executed when the module is loaded into memory.	Self-test success is indicated by module and algorithm availability; failure is indicated by an error.	This service is fully automatic, executed when the module is loaded into memory.	The availability of the module is the service output. See section 10 Self-Tests for more information.	BC1 DRBG1 DigSig-Legacy DigSig1 DigSig2 SHS1	Cryptographic Officer (CO)
Perform Pre-operational Software Integrity Test [EVM]	The pre-operational integrity test is executed by the [EVM] Boot Manager (IG 1.A Documentation Requirements 5) before the OS Loader module is loaded into memory.	Integrity test success is indicated by module and algorithm availability; failure is indicated by an error.	This service is fully automatic and executed by the EVM before the module is loaded into memory.	The availability of the module is the service output. See section 10 Self-Tests for more information.	DigSig2 SHS1	Cryptographic Officer (CO)
Perform Zeroization	Zeroizes cryptographic material.	Volatile keys are zeroized. See Section 9.3 SSP Zeroization Methods for more information.	This service is fully automatic.	Keys are zeroized and the module unloads from memory.	BC1 DRBG1	Cryptographic Officer (CO) - AES-CTR DRBG Entropy Input String and Nonce: Z - AES-CTR DRBG Seed: Z - AES-CTR DRBG V: Z - AES-CTR DRBG Key: Z - Full Volume Encryption Key (FVEK): Z
Show Status	Provides the module status response.	Provided via NTSTATUS by all functions of approved services.	This service is fully automatic and occurs when any function is called.	Status is output across the module's Status Output logical interface, to the computer monitor or to log files.	None	Cryptographic Officer (CO)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show Version	Provides the module version number.	Version information is provided in the Portable Executable (PE) header of each module binary. The PE header contains Windows-specific fields such as the major and minor version (10.0.22621.1 for Windows 11 and 10.0.20348.1668 for Windows Server). See the PE Format public documentation published on https://learn.microsoft.com/ for more information.	Module binary file.	Version information is embedded in the PE header of the binary.	None	Cryptographic Officer (CO)

Table 18: Approved Services

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

The module does not load external software or firmware. While it is running, the module is the only process running on the computer.

5 Software/Firmware Security

The secure installation, generation, and startup procedures of this module are part of the secure installation, configuration, and startup procedures of the Windows operating systems named in Section 2.2 Tested and Vendor Affirmed Module Version and Identification.

5.1 Integrity Techniques

Windows uses several mechanisms to provide integrity verification depending on the stage in the boot sequence and also on the hardware and configuration. The algorithms used for integrity verification are included in section 2.5 Algorithms, Table 11.

- The [EVM] Boot Manager module (IG 1.A Documentation Requirements 5) checks the integrity of the WINLOAD.EFI component of the Windows OS Loader before it is loaded. To perform the module integrity test on demand, the user may restart the computer. See the table below for more information on the [EVM] Boot Manager.
- Windows binaries include a SHA2-256 hash of the binary signed with the 2048-bit Microsoft RSA code-signing key (i.e., the key associated with the Microsoft code-signing certificate). The integrity check uses the public key component of the Microsoft code signing certificate to verify the signed hash of the binary.
- The Windows OS Loader component verifies the integrity of multiple kernel mode cryptographic modules using the same process described above. The following binaries are verified:
    - CI.DLL.
    - CNG.SYS.
    - SKCI.DLL (When VSM / core isolation is enabled).
- Windows OS Loader also verifies the integrity of other early boot kernel mode drivers that are not cryptographic modules.

The following table provides the details of the [EVM] Boot Manager (IG 1.A Documentation Requirements 5).

EVM Name	CMVP Certificate	Version Details
Boot Manager (Windows 11 v 22H2 and Windows Server 2022)	#5404	Windows 11 version 22H2, build 10.0.22621.1 Windows Server 2022, build 10.0.20348.1668

Table 19: Existing Validated Module Details

The figure below shows the Integrity Chain of trust for the Windows builds and modules in scope for this validation.

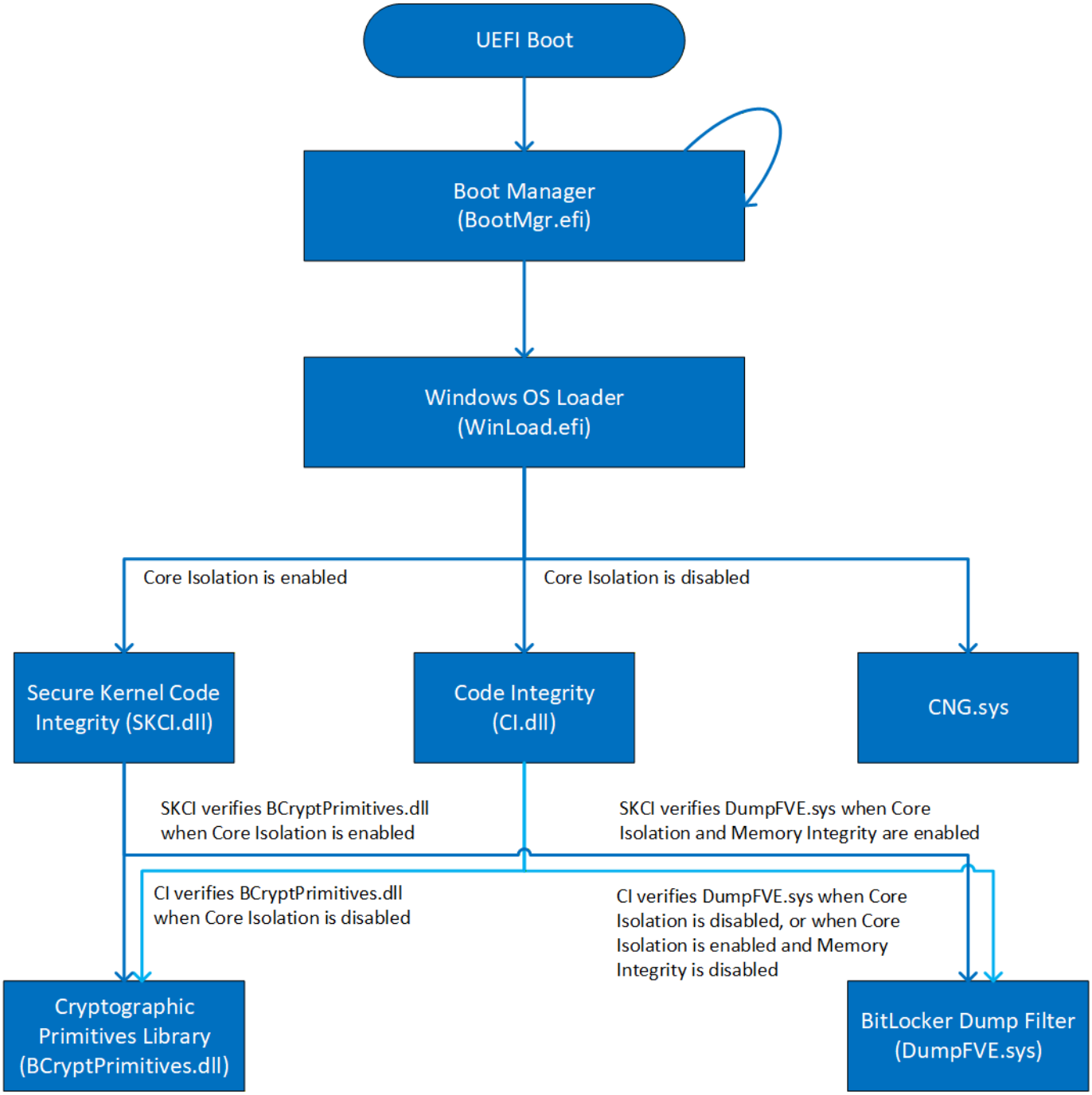


Figure 2: Integrity Chain of Trust

5.2 Initiate on Demand

To initiate the integrity test on demand, the operator may restart the computer.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The modifiable operational environment for the module is the Windows operating system running on a supported hardware platform, as listed in Section 2.2 Tested and Vendor Affirmed Module Version and Identification. During the operating system boot process there is no logged-on user, so the single operator requirement is met.

7 Physical Security

7.1 Mechanisms and Actions Required

The Windows OS Loader is a multi-chip standalone software-hybrid module whose host platforms meet the Level 1 physical security requirements. The host platform consists of production-grade physical security components that include standard passivation techniques and is entirely contained within a metal or hard plastic production-grade enclosure that may include doors or removable covers.

Tables identifying the voltage and temperature boundaries that trigger zeroization or shutdown are not included in this Security Policy as they are N/A for a Level 1 validation of a hybrid module.

8 Non-Invasive Security

N/A for this module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Hard Disk	Persistent SSPs are stored on the operating system volume (see: Hard Disk in the block diagram).	Static
RAM	Volatile SSPs are temporarily stored in the computer's memory (see: RAM in the block diagram).	Dynamic

Table 20: Storage Areas

9.2 SSP Input-Output Methods

The table below lists the single SSP input into the module. The SFI or Algorithm column is blank as the module does not use any SFIs or algorithms when the SSP is input.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Input in plaintext	[EVM] Boot manager	RAM	Plaintext	Manual	Electronic	

Table 21: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Procedural zeroization	Operators may choose to overwrite CSPs in non-volatile storage by reformatting and overwriting the storage media for the operating system. Operators may zeroize SSPs in volatile storage by powering off the host GPC.	Reformatting and overwriting the OS volume at least once ensures adequate zeroization. Operators may choose to overwrite multiple times at their discretion. SSPs in volatile storage are effectively overwritten with zeroes when power is lost.	Operators may use the Format.exe command together with the /P parameter to format and overwrite every sector on the volume with zeros. Operators may specify the number of overwrite passes using the /P parameter. See the public documentation for the Format command for more information. Operators may power off or reboot the host GPC to zeroize SSPs stored in volatile RAM.
Zeroization after use	The module overwrites the temporary storage area for volatile SSPs when the module is unloaded from memory, when control has been transferred to NTOSKRNL.exe.	Zeroization is performed by overwriting the memory area with zeros using a forced inline function to minimize execution time. Because the memory word is explicitly set to '0', it is not necessary to do a read-back test. Windows uses only one type of system memory.	None (programmatically executed by the module after CSPs are used).

Table 22: SSP Zeroization Methods

9.4 SSPs

The tables below list the SSPs used by the module. Per the CMVP, public keys and file hashes used for the module integrity check are not considered SSPs and, as such, have no input method assigned and are categorized as neither PSP nor CSP. Blank cells are either not applicable or optional according to the CMVP.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES-CTR DRBG Entropy Input String and Nonce	Provides entropy material for AES-CTR DRBG.	Size: 512-bit - Strength: 512-bit	Secret Value - CSP	ESV		DRBG1
AES-CTR DRBG Seed	Provides seed material for AES-CTR DRBG output.	Size: 512-bit - Strength: 512-bit	Secret Value - CSP	DRBG1		DRBG1
AES-CTR DRBG V	Provides entropy material for AES-CTR DRBG output.	Size: 128-bit - Strength: 128-bit	Secret Value - CSP	DRBG1		DRBG1
AES-CTR DRBG Key	Provides entropy material for AES-CTR DRBG output.	Size: 256-bit - Strength: 256-bit	Secret Value - CSP	DRBG1		DRBG1
Full Volume Encryption Key (FVEK)	An AES key used for encryption / decryption of data on disk sectors. This key is stored encrypted on the system volume. The Boot Manager decrypts the FVEK with the VMK using AES-CCM.	Size: 128-bit or 256-bit (administrator configurable) - Strength: 128 or 256 bits (administrator configurable)	Symmetric Key (AES) - CSP	Externally generated by BitLocker runtime components outside of the cryptographic boundary.		BC1
Microsoft Root Certificate Authority Public Key (This is not an SSP)	Public key used for RSA PKCS #1 (v1.5) verification of digital signatures.	Size: 2048-bit - Strength: 112 bits	Asymmetric Public Key (RSA) - Neither	Generated external to the module by the Microsoft root CA and pre-loaded from the manufacturer.		DigSig1 DigSig2 DigSig-Legacy
SHA Hashes (This is not an SSP)	File hashes used to verify the integrity of binaries.	Size: 160 (legacy use), 256, 384, or 512 bits - Strength: 80 (legacy use), 128, 192, or 256 bits	Secure Hash - Neither	Generated external to the module, and added as part of module binaries when the binary files are written to disk.		SHS1

Table 23: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES-CTR DRBG Entropy Input String and Nonce		RAM:Plaintext	Until zeroized	Zeroization after use	
AES-CTR DRBG Seed		RAM:Plaintext	Until zeroized	Zeroization after use	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES-CTR DRBG V		RAM:Plaintext	Until zeroized	Zeroization after use	
AES-CTR DRBG Key		RAM:Plaintext	Until zeroized	Zeroization after use	
Full Volume Encryption Key (FVEK)	Input in plaintext	RAM:Plaintext	Until zeroized	Procedural zeroization	
Microsoft Root Certificate Authority Public Key (This is not an SSP)		Hard Disk:Plaintext		Procedural zeroization	
SHA Hashes (This is not an SSP)		Hard Disk:Plaintext		Procedural zeroization	

Table 24: SSP Table 2

9.5 Transitions

The following transition timelines apply to the approved algorithms named in the bullets below:

- The SHA-1 algorithm will become disallowed for applying cryptographic protection starting January 1, 2031, however, its use for legacy digital signature verification will continue to be allowed.
- RSA with a security strength of less than 128 bits will become deprecated starting January 1, 2031, however, its use for legacy digital signature verification will continue to be allowed.
- FIPS 186-4 has been superseded by FIPS 186-5. This transition began on July 25, 2023, and concluded on February 3, 2024. Although testing for this module was completed against FIPS 186-4, it claims compliance with FIPS 186-5 for RSA signature verification – see Section 2.7.1 FIPS 186-4 and 186-5 for more information.

10 Self-Tests

Windows performs tests automatically to ensure integrity and correct functionality. The module will not perform cryptographic functions while in its self-test or error states. If a self-test fails, the module enters the error state and future cryptographic function calls fail. If the self-test passes, cryptographic functions are available for use. As the module has separate CAVP certificates for Windows 11 and Windows Server 2022, the self-test tables below list two identical rows for each algorithm self-test, one for Windows 11 and one for Windows Server 2022.

10.1 Pre-Operational Self-Tests

The Windows Boot Manager module (EVM) checks the integrity of WINLOAD.EFI before it is loaded. The algorithms used for the pre-operational self-tests pass their own algorithm self-tests before integrity verification is performed. See Section 5.1 Integrity Techniques for additional information on the EVM and how the module's integrity is checked.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
RSA SigVer (FIPS186-4) (A3767)	Signature verification using RSA PKCS#1v1.5 with 2048-bit key and SHA2-256 (Windows 11 version 22H2).	Software Integrity	SW/FW Integrity	Perform Cryptographic Algorithm Self-Tests service runs.	[EVM] Signature verification.
RSA SigVer (FIPS186-4) (A3768)	Signature verification using RSA PKCS#1v1.5 with 2048-bit key and SHA2-256 (Windows Server 2022).	Software Integrity	SW/FW Integrity	Perform Cryptographic Algorithm Self-Tests service runs.	[EVM] Signature verification.

Table 25: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The Windows OS Loader module performs the conditional cryptographic algorithm self-tests automatically when the module is loaded into memory, after the pre-operational software integrity tests described above have completed.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A4008) - Encrypt	128-bit AES key (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Encrypt KAT for AES.	Run at every module initialization after the module integrity is verified.
AES-CBC (A4008) - Decrypt	128-bit AES key (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for AES	Run at every module initialization after the module integrity is verified.
AES-CBC (A4009) - Encrypt	128-bit AES key (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Encrypt KAT for AES.	Run at every module initialization after the module integrity is verified.
AES-CBC (A4009) - Decrypt	128-bit AES key (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for AES.	Run at every module initialization after the module integrity is verified.
AES-CCM (A4008) - Encrypt	128-bit AES key (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Encrypt KAT for AES	Run at every module initialization after the module integrity is verified.
AES-CCM (A4008) - Decrypt	128-bit AES key (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for AES.	Run at every module initialization after the module integrity is verified.
AES-CCM (A4009) - Encrypt	128-bit AES key (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Encrypt KAT for AES	Run at every module initialization after the module integrity is verified.
AES-CCM (A4009) - Decrypt	128-bit AES key (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for AES.	Run at every module initialization after the module integrity is verified.
AES-GCM (A4008) - Decrypt	128-bit AES key (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for AES.	Run at every module initialization after the module integrity is verified.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A4009) - Decrypt	128-bit AES key (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for AES.	Run at every module initialization after the module integrity is verified.
AES-XTS Testing Revision 2.0 (A4008) - Encrypt	128-bit AES key (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Encrypt KAT for AES-XTS	Run at every module initialization after the module integrity is verified.
AES-XTS Testing Revision 2.0 (A4008) - Decrypt	128-bit AES key (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for XTS-AES	Run at every module initialization after the module integrity is verified.
AES-XTS Testing Revision 2.0 (A4008) - Key Equivalence Test	128-bit AES key (Windows 11 version 22H2)	Key Equivalence Test	CAST	Cryptographic operations execute.	XTS-AES Key_1 Key_2 check in compliance with FIPS 140-3 IG C.I.	Checks AES-XTS keys before using the algorithm.
AES-XTS Testing Revision 2.0 (A4009) - Encrypt	128-bit AES key (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Encrypt KAT for AES-XTS	Run at every module initialization after the module integrity is verified.
AES-XTS Testing Revision 2.0 (A4009) - Decrypt	128-bit AES key (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Decrypt KAT for AES-XTS	Run at every module initialization after the module integrity is verified.
AES-XTS Testing Revision 2.0 (A4009) - Key Equivalence Test	128-bit AES key (Windows Server 2022)	Key Equivalence Test	CAST	Cryptographic operations execute.	XTS-AES Key_1 Key_2 check in compliance with FIPS 140-3 IG C.I.	Checks AES-XTS keys before using the algorithm.
Counter DRBG (A4008)	AES-256 with derivation function and without prediction resistance (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Instantiate, generate, and reseed	Run at every module initialization after the module integrity is verified.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG (A4009)	AES-256 with derivation function and without prediction resistance (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Instantiate, generate, and reseed	Run at every module initialization after the module integrity is verified.
RSA SigVer (FIPS186-4) (A3767)	RSA PKCS#1v1.5 with 2048-bit key and SHA2-256 (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Signature verification.	Run at every module initialization after the module integrity is verified.
RSA SigVer (FIPS186-4) (A3768)	RSA PKCS#1v1.5 with 2048-bit key and SHA2-256 (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Signature verification	Run at every module initialization after the module integrity is verified.
SHA-1 (A4008)	160 bits (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute	Secure hash	Run at every module initialization after the module integrity is verified.
SHA-1 (A4009)	160 bits (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Secure hash	Run at every module initialization after the module integrity is verified.
SHA2-256 (A4008)	256 bits (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Secure hash	Run at every module initialization after the module integrity is verified.
SHA2-256 (A4009)	256 bits (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Secure hash	Run at every module initialization after the module integrity is verified.
SHA2-384 (A4008)	384 bits (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Secure hash	Run at every module initialization after the module integrity is verified.
SHA2-384 (A4009)	384 bits (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Secure hash	Run at every module initialization after the module integrity is verified.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-512 (A4008)	512 bits (Windows 11 version 22H2)	KAT	CAST	Cryptographic operations execute.	Secure hash	Run at every module initialization after the module integrity is verified.
SHA2-512 (A4009)	512 bits (Windows Server 2022)	KAT	CAST	Cryptographic operations execute.	Secure hash	Run at every module initialization after the module integrity is verified.
[EVM] RSA SigVer (FIPS186-4) (A3767)	RSA PKCS#1v1.5 2048-bit key with SHA2-256.	KAT	CAST	The module's integrity verification executes.	[EVM] Signature verification.	Run at every module initialization before the module integrity is verified.
[EVM] RSA SigVer (FIPS186-4) (A3768)	RSA PKCS#1v1.5 2048-bit key with SHA2-256.	KAT	CAST	The module's integrity verification executes.	[EVM] Signature verification.	Run at every module initialization before the module integrity is verified.
[EVM] Intel-based Entropy Source (E216)	256-bit entropy input (Windows 11 22H2)	Fault detection	CAST	The entropy source is instantiated and a status is returned via the interface.	[EVM] Start-up and continuous noise source health tests and start-up logic integrity self-test.	Run at start-up and continuously.
[EVM] Intel-based Entropy Source (E189)	256-bit entropy input (Windows Server 2022)	Fault detection	CAST	The entropy source is instantiated and a status is returned via the interface.	[EVM] Start-up and continuous noise source health tests and start-up logic integrity self-test.	Run at start-up and continuously.

Table 26: Conditional Self-Tests

10.3 Periodic Self-Test Information

The tables below present the periodic self-test information for the module. The set of self-tests presented in tables 25 and 26 below is identical to the set of self-tests presented in tables 23 and 24 above. The Windows Boot Manager module checks the integrity of WINLOAD.EFI before it is loaded. See Section 5.1 Integrity Techniques for additional information on the EVM and how the module’s integrity is checked

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-4) (A3767)	Software Integrity	SW/FW Integrity	Pre-operational integrity test is run at every module initialization before the CASTs (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer).
RSA SigVer (FIPS186-4) (A3768)	Software Integrity	SW/FW Integrity	Pre-operational integrity test is run at every module initialization before the CASTs (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer).

Table 27: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A4008) - Encrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
AES-CBC (A4008) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
AES-CBC (A4009) - Encrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
AES-CBC (A4009) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
AES-CCM (A4008) - Encrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CCM (A4008) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
AES-CCM (A4009) - Encrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
AES-CCM (A4009) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
AES-GCM (A4008) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
AES-GCM (A4009) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
AES-XTS Testing Revision 2.0 (A4008) - Encrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
AES-XTS Testing Revision 2.0 (A4008) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-XTS Testing Revision 2.0 (A4008) - Key Equivalence Test	Key Equivalence Test	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
AES-XTS Testing Revision 2.0 (A4009) - Encrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
AES-XTS Testing Revision 2.0 (A4009) - Decrypt	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
AES-XTS Testing Revision 2.0 (A4009) - Key Equivalence Test	Key Equivalence Test	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
Counter DRBG (A4008)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
Counter DRBG (A4009)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
RSA SigVer (FIPS186-4) (A3767)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-4) (A3768)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
SHA-1 (A4008)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
SHA-1 (A4009)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
SHA2-256 (A4008)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
SHA2-256 (A4009)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
SHA2-384 (A4008)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
SHA2-384 (A4009)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022)	Manual on-demand (operator initiated by rebooting the computer)

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-512 (A4008)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows 11 version 22H2).	Manual on-demand (operator initiated by rebooting the computer)
SHA2-512 (A4009)	KAT	CAST	Conditional CASTs are run at every module initialization after the module integrity and the boot policy file integrity is verified (Windows Server 2022).	Manual on-demand (operator initiated by rebooting the computer)
[EVM] RSA SigVer (FIPS186-4) (A3767)	KAT	CAST	Conditional CAST is run before module's integrity verification.	Manual on-demand (operator initiated by rebooting the computer)
[EVM] RSA SigVer (FIPS186-4) (A3768)	KAT	CAST	Conditional CAST is run before module's integrity verification.	Manual on-demand (operator initiated by rebooting the computer)
[EVM] Intel-based Entropy Source (E216)	Fault detection	CAST	Continuously checks the health of entropy output (Windows 11 22H2).	Manual on-demand (operator initiated by rebooting the computer).
[EVM] Intel-based Entropy Source (E189)	Fault detection	CAST	Continuously checks the health of entropy output (Windows Server 2022)..	Manual on-demand (operator initiated by rebooting the computer).

Table 28: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Boot Fail	Boot failure	1) Error state occurs if the OS Loader integrity test fails; 2) If any of the cryptographic algorithm self-tests fail; 3) If any of the signature verifications fail for the Code Integrity module (CI.DLL), Secure Kernel Code Integrity module (SKCI.DLL), Kernel Mode Cryptographic Primitives Library module (CNG.SYS), or operating system kernel (NTOSKRNL.EXE); 4) If any of the health tests fail for the entropy source thus failing to instantiate the DRBG, or if the DRBG KAT fails.	Restart the computer	Boot failure blue screen error message

Table 29: Error States

10.5 Operator Initiation of Self-Tests

To perform the module self-tests on demand, including running the approved services Perform Pre-operational Software Integrity Test [EVM] and Perform Cryptographic Algorithm Self-Tests, the user may reboot the computer.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Windows operating system must be pre-installed on a computer by an OEM, installed by the end-user, by an organization's IT administrator, or updated from a previous Windows version downloaded from Windows Update. An inspection of authenticity can be made by following the guidance at this Microsoft web site: https://www.microsoft.com/en-us/howtotell/default.aspx.

For Windows Updates, the client only accepts binaries signed by Microsoft certificates. The Windows Update client only accepts content whose SHA2 hash matches the SHA2 hash specified in the metadata. All metadata communication is done over a Transport Layer Security (TLS) port. Using TLS ensures that the client is communicating with the real server and so prevents a malicious TLS server from communicating to the TLS client. The version and digital signature of new cryptographic module must be verified to match the version that was validated. See Section 11.2 Administrator Guidance for details on how to do this.

Module initialization occurs automatically as part of the Windows boot process. The finite state model diagram below visualizes the initialization process along with other module states. Every state of the module can transition to the power-off state through power-cycle/rebooting the host machine.

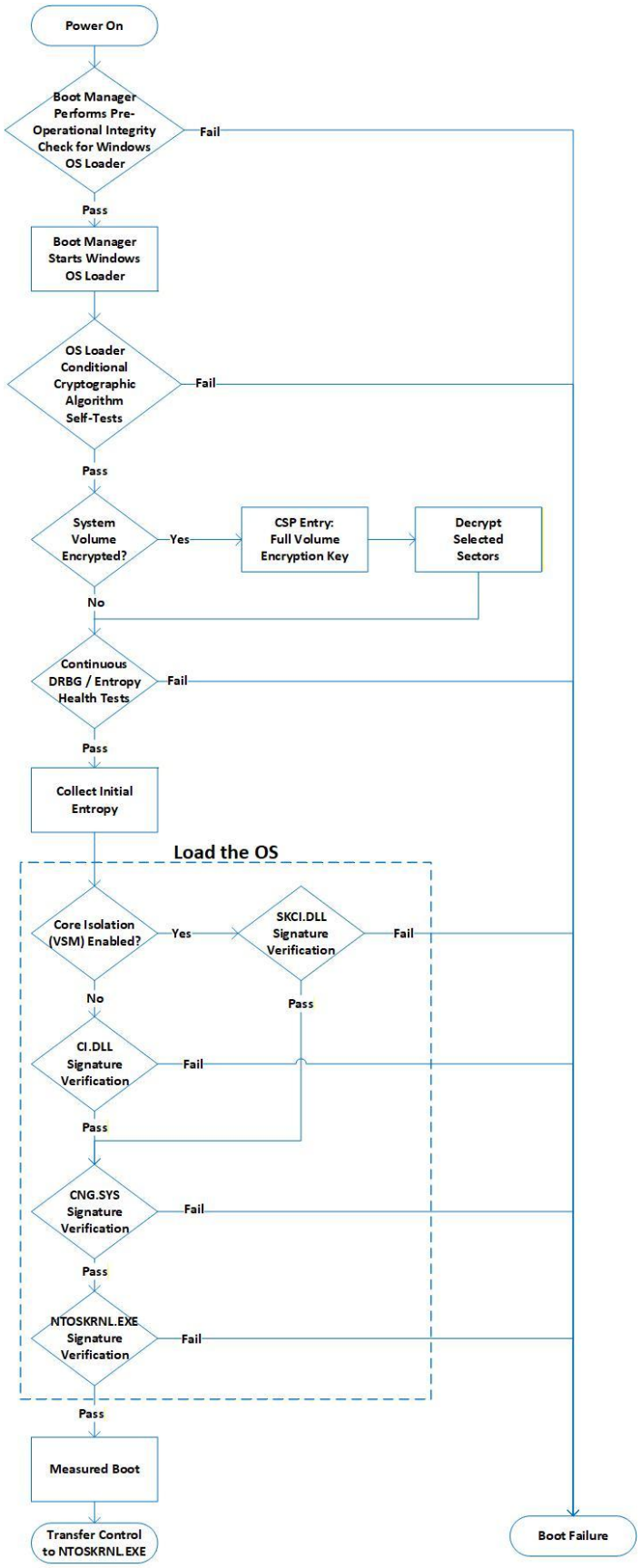


Figure 3: Finite State Model

11.2 Administrator Guidance

The installed version of Windows must be checked to match the version that was validated. See Section 11.2.1 Verifying the Installed Windows Version below for details on how to do this. To sanitize the module, the operator should reformat the hard drive or wipe the device as part of unenrollment for Azure Active Directory.

11.2.1 Verifying the Installed Windows Version

The following methods may be used to check the installed version of Windows against the version number listed in 2.2 Tested and Vendor Affirmed Module Version and Identification.

Using the Windows command prompt or Windows PowerShell (local or remote):

- Open a command prompt or PowerShell window.
- At the prompt, type **systeminfo** and press the Enter key.
- Near the top of the output, information like the following is displayed. The OS Version field lists the installed Windows version. Compare this version number against the version number listed in 2.2 Tested and Vendor Affirmed Module Version and Identification.

```
OS Name:                   Microsoft Windows 11 Enterprise
OS Version:                10.0.xxxxx N/A Build xxxxx
OS Manufacturer:           Microsoft Corporation
```

For Windows installations without a user interface, e.g., Windows Server with the Core Installation option, a server management solution may also be used to validate the installed Windows version. For example, the Overview page of Windows Admin Center Server Manager lists the version number under the Operating System category. For more information, see Manage Servers with Windows Admin Center.

11.2.2 Verifying the Cryptographic Module Version and its Signature

To confirm the version number or digital signature of the module, locate the module binary or binaries named in 2.2 Tested and Vendor Affirmed Module Version and Identification in their default installation location. The list below identifies the default install locations for the Windows cryptographic module binaries for a system where Windows has been installed on the C: drive.

- WINLOAD.EFI - C:\Windows\System32\

To validate the module version number, use Windows Explorer or PowerShell (local or remote).

- Using Windows Explorer:
    - Open Windows Explorer and navigate to the folder where the binary is installed, referencing the list above for the correct location.
    - Find the file in the folder and **right click** on the file's icon.
    - Select **Properties** from the context menu.
    - Select the **Details** tab.

- Compare the version number in the **File version** field against the version identified in 2.2 Tested and Vendor Affirmed Module Version and Identification.

- Using PowerShell:
  - Open a **PowerShell** window.
  - Use the **Get-ItemProperty cmdlet** together with the path of the cryptographic module binary identified above, formatting the output as a list. For example, if the binary path is `C:\Windows\System32\winload.efi`, the PowerShell command is:

    ```
    Get-ItemProperty -Path "C:\Windows\System32\winload.efi" | Format-List
    ```

  - The cmdlet will return output that summarizes file property details, including the version number. If the version number listed in the **VersionInfo / ProductVersion** field matches one of the version numbers identified in 2.2 Tested and Vendor Affirmed Module Version and Identification, then the module version has been verified.
  - Full documentation for the Get-ItemProperty cmdlet may be found at: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-itemproperty.

To validate the Windows digital signature for the module binary, use Windows Explorer or PowerShell (local or remote).

- Using Windows Explorer:
  - Open Windows Explorer and navigate to the folder where the binary is installed, referencing the list at the beginning of this section for the correct location.
  - Find the file in the folder and **right click** on the file's icon.
  - Select **Properties** from the context menu.
  - Select the **Digital Signatures** tab.
  - In the **Signature** list, select the **Microsoft Windows** signer.
  - Click the **Details** button.
  - Under the Digital Signature Information, you should see: "`This digital signature is OK.`" If that condition is true then the digital signature has been verified.

- Using PowerShell:
  - Open a **PowerShell** window.
  - Use the **Get-AuthenticodeSignature cmdlet** together with the path the path of the cryptographic module binary identified at the beginning of this section. For example, if the binary path is `C:\Windows\System32\winload.efi`, the PowerShell command is:

```
Get-AuthenticodeSignature -FilePath "C:\Windows\System32\winload.efi"
```

- The cmdlet will return output that summarizes signature details. If the signature is valid, the **Status** field will show `"Valid"` and the **StatusMessage** field will show `"Signature verified."`
- Full documentation for the Get-AuthenticodeSignature cmdlet may be found at: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/get-authenticodesignature.

11.3 Non-Administrator Guidance

The module implements a single role only, Cryptographic Officer. See the Administrator Guidance above.

11.4 Design and Rules

The module is a software-hybrid cryptographic module that provides cryptographic services within the Operational Environments listed in Section 2.2 Tested and Vendor Affirmed Module Version and Identification, Table 7. The other sections of this Security Policy provide additional details on the design of the module and its rules of operation.

12 Mitigation of Other Attacks

12.1 Attack List

The following table lists the mitigations of other attacks for this cryptographic module.

Algorithm	Protected Against	Mitigation
SHA-1	Timing Analysis Attack	Constant time implementation.
	Cache Attack	Memory access pattern is independent of any confidential data.
SHA2	Timing Analysis Attack	Constant time implementation.
	Cache Attack	Memory access pattern is independent of any confidential data.
AES	Timing Analysis Attack	Constant time implementation.
	Cache Attack	Memory access pattern is independent of any confidential data.
		Protected against cache attacks only when running on a processor that implements AES-NI.

Table 30: Mitigation of Other Attacks

13 References

- FIPS 140-3, Security Requirements for Cryptographic Modules, https://csrc.nist.gov/publications/detail/fips/140/3/final
- FIPS 180-4, Secure Hash Standard (SHS), https://csrc.nist.gov/publications/detail/fips/180/4/final

- FIPS 186-4, Digital Signature Standard (DSS), https://csrc.nist.gov/publications/detail/fips/186/4/final
- FIPS 186-5, Digital Signature Standard (DSS), https://csrc.nist.gov/pubs/fips/186-5/final
- FIPS 197, Advanced Encryption Standard (AES), https://csrc.nist.gov/publications/detail/fips/197/final
- NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, https://csrc.nist.gov/publications/detail/sp/800-38a/final
- NIST SP 800-38C, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality, https://csrc.nist.gov/publications/detail/sp/800-38c/final
- NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, https://csrc.nist.gov/publications/detail/sp/800-38d/final
- NIST SP 800-38E, Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices, https://csrc.nist.gov/publications/detail/sp/800-38e/final
- NIST SP 800-90A Rev. 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final
- NIST SP 800-90B, Recommendation for the Entropy Sources Used for Random Bit Generation, https://csrc.nist.gov/publications/detail/sp/800-90b/final
- Public Use Document for Microsoft Windows OS Loader Entropy Source for Microsoft Windows 11 22H2 (certificate E189), https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/entropy/E189_PublicUse.pdf
- Public Use Document for Microsoft Windows OS Loader Entropy Source for Microsoft Windows Server 2022 (certificate E216), https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/entropy/E216_PublicUse.pdf