



Microsoft Windows

FIPS 140 Validation

Microsoft Windows Server 2019

Microsoft Azure Stack Edge

Microsoft Azure Stack Hub

Microsoft Azure Stack Edge Rugged

Non-Proprietary

Security Policy Document

Version Number	1.3
Updated On	January 23, 2024

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.

Complying with all applicable copyright laws is the responsibility of the user. This work is licensed under the Creative Commons Attribution-NoDerivs-NonCommercial License (which allows redistribution of the work). To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd-nc/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

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.

© 2024 Microsoft Corporation. All rights reserved.

Microsoft, Windows, the Windows logo, Windows Server, and BitLocker 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.

Version History

Version	Date	Summary of changes
1.0	November 4, 2020	Draft sent to NIST CMVP
1.1	November 3, 2022	Updates in response to NIST feedback
1.2	December 12, 2023	Updates to bounded module certificate information
1.3	January 23, 2024	Updates in response to NIST feedback

TABLE OF CONTENTS

<u>SECURITY POLICY DOCUMENT</u>	<u>1</u>
<u>VERSION HISTORY</u>	<u>3</u>
<u>1 INTRODUCTION</u>	<u>6</u>
1.1 LIST OF CRYPTOGRAPHIC MODULE BINARY EXECUTABLES.....	6
1.2 VALIDATED PLATFORMS.....	6
<u>2 CRYPTOGRAPHIC MODULE SPECIFICATION</u>	<u>7</u>
2.1 CRYPTOGRAPHIC BOUNDARY	8
2.2 FIPS 140-2 APPROVED ALGORITHMS	8
2.3 NON-APPROVED ALGORITHMS.....	8
2.4 FIPS 140-2 APPROVED ALGORITHMS FROM BOUNDED MODULES.....	8
2.5 CRYPTOGRAPHIC BYPASS	9
2.6 HARDWARE COMPONENTS OF THE CRYPTOGRAPHIC MODULE.....	9
<u>3 CRYPTOGRAPHIC MODULE PORTS AND INTERFACES</u>	<u>9</u>
3.1 CONTROL INPUT INTERFACE	9
3.1.1 GETFVECONTEXT	10
3.1.2 DUMPWRITE	10
3.2 STATUS OUTPUT INTERFACE	10
3.3 DATA OUTPUT INTERFACE.....	10
3.4 DATA INPUT INTERFACE	10
<u>4 ROLES, SERVICES AND AUTHENTICATION</u>	<u>11</u>
4.1 ROLES	11
4.2 SERVICES.....	11
4.3 AUTHENTICATION.....	12
<u>5 FINITE STATE MODEL</u>	<u>12</u>
5.1 SPECIFICATION	12

<u>6</u>	<u>OPERATIONAL ENVIRONMENT</u>	<u>13</u>
6.1	SINGLE OPERATOR	13
6.2	CRYPTOGRAPHIC ISOLATION	13
6.3	INTEGRITY CHAIN OF TRUST	13
<u>7</u>	<u>CRYPTOGRAPHIC KEY MANAGEMENT.....</u>	<u>15</u>
7.1	CRITICAL SECURITY PARAMETERS	15
7.2	ZEROIZATION	15
7.2.1	VOLATILE KEYS.....	15
7.2.2	PERSISTENT KEYS.....	15
7.3	ACCESS CONTROL POLICY.....	15
<u>8</u>	<u>SELF-TESTS.....</u>	<u>15</u>
8.1	POWER-ON SELF-TESTS.....	15
<u>9</u>	<u>DESIGN ASSURANCE.....</u>	<u>16</u>
<u>10</u>	<u>MITIGATION OF OTHER ATTACKS</u>	<u>16</u>
<u>11</u>	<u>SECURITY LEVELS</u>	<u>16</u>
<u>12</u>	<u>ADDITIONAL DETAILS</u>	<u>17</u>
<u>13</u>	<u>APPENDIX A – HOW TO VERIFY WINDOWS VERSIONS AND DIGITAL SIGNATURES</u>	<u>18</u>
13.1	HOW TO VERIFY WINDOWS VERSIONS	18
13.2	HOW TO VERIFY WINDOWS DIGITAL SIGNATURES.....	18

1 Introduction

BitLocker Drive Encryption is a data protection feature of the Windows Server operating system which encrypts data on a storage volume.

This security policy document describes the BitLocker Dump Filter cryptographic module which protects crash dump files on BitLocker encrypted computers. Other parts of BitLocker are described in the Security Policy Documents for Boot Manager, and Windows OS Loader.

The BitLocker Dump Filter is part of the system dump stack. Whenever the dump stack is called during a crash, this module ensures that all data is encrypted before written to storage as a dump file. For the purpose of this validation, BitLocker Dump Filter is classified as a Software-Hybrid cryptographic module because the validated platforms all implement the AES-NI instruction set.

1.1 List of Cryptographic Module Binary Executables

The BitLocker Dump Filter module contains the following binaries:

- DUMPFVE.SYS

The Windows builds covered by this validation are:

- Windows Server 2019 build 10.0.17763.10021 and 10.0.17763.10127

1.2 Validated Platforms


The editions covered by this validation are:




- Windows Server 2019 Datacenter Core

The BitLocker Dump Filter components listed in Section 1.1 were validated using the machine configurations and Windows Operating System versions specified in the table below.

All the computers for Windows Server listed in the table below are all 64-bit Intel architecture and implement the AES-NI instruction set but not the SHA Extensions.

Table 1 Validated Platforms

Computer	Windows Server 2019 Datacenter Core	Processor Image
Microsoft Azure Stack Edge - Dell XR2 - Intel Xeon Silver 4114	✓	

		wikichip.org
Microsoft Azure Stack Hub - Dell PowerEdge R640 - Intel Xeon Gold 6230	√	 wikichip.org
Microsoft Azure Stack Hub - Dell PowerEdge R840 - Intel Xeon Platinum 8260	√	 wikichip.org
Microsoft Azure Stack Edge Rugged - Rugged Mobile Appliance – Intel Xeon D-1559	√	 wikichip.org

2 Cryptographic Module Specification

BitLocker Dump Filter is a multi-chip standalone module that operates in FIPS-approved mode during normal operation of the computer and Windows operating system.

The following configurations and modes of operation will cause BitLocker Dump Filter to operate in a non-approved mode of operation:

- Boot Windows in Debug mode
- Boot Windows with Driver Signing disabled

2.1 Cryptographic Boundary

The software-hybrid cryptographic boundary for BitLocker Dump Filter consists of disjoint software and hardware components within the same physical boundary of the host platform. The software components are defined as the binary DUMPFVE.SYS, and the hardware components are the CPUs running on each host platform.

2.2 FIPS 140-2 Approved Algorithms

BitLocker Dump Filter implements the following FIPS 140-2 Approved algorithms:¹

Table 2 Approved Algorithms

Algorithm	Windows Server 2019 build 10.0.17763.10021	Windows Server 2019 build 10.0.17763.10127
FIPS 197 AES CBC 128 and 256 (Encrypt)	#C1577	#C2044
NIST SP 800-38E AES XTS 128 and 256 (Encrypt)	#C1577	#C2044
NIST SP 800-38C AES CCM 256 (Encrypt)	#C1583	#C2049

2.3 Non-Approved Algorithms

BitLocker Dump Filter implements only Approved algorithms.

2.4 FIPS 140-2 Approved Algorithms from Bounded Modules

A bounded module is a FIPS 140 module which provides cryptographic functionality that is relied on by a downstream module. As described in the [Integrity Chain of Trust](#) section, the BitLocker Dump Filter depends on the modules and algorithms below. Hypervisor Code Integrity (HVCI), also known as Memory Integrity, is a configurable Windows feature that determines whether the Code Integrity or Secure Kernel Code Integrity module is used to verify a binary image. When HVCI is enabled, the Secure Kernel Code Integrity module is used; when HVCI is disabled, the Code Integrity module is used.

When Memory Integrity, called HVCI in previous Windows Server versions, is not enabled, Code Integrity (module certificate [#4602](#)) provides:

- CAVP certificates #C1577 and #C2044 for FIPS 186-4 RSA PKCS#1 (v1.5) digital signature verification with 2048 moduli; supporting SHA-256
- CAVP certificates #C1577 and #C2044 for FIPS 180-4 SHS SHA-256

When Memory Integrity, called HVCI in previous Windows Server versions, is enabled, Secure Kernel Code Integrity (module certificate [#4640](#)) provides:

¹ This module may not use some of the capabilities described in each CAVP certificate.

- CAVP certificates #C1577 and #C2044 for FIPS 186-4 RSA PKCS#1 (v1.5) digital signature verification with 2048 moduli; supporting SHA-256
- CAVP certificates #C1577 and #C2044 for FIPS 180-4 SHS SHA-256

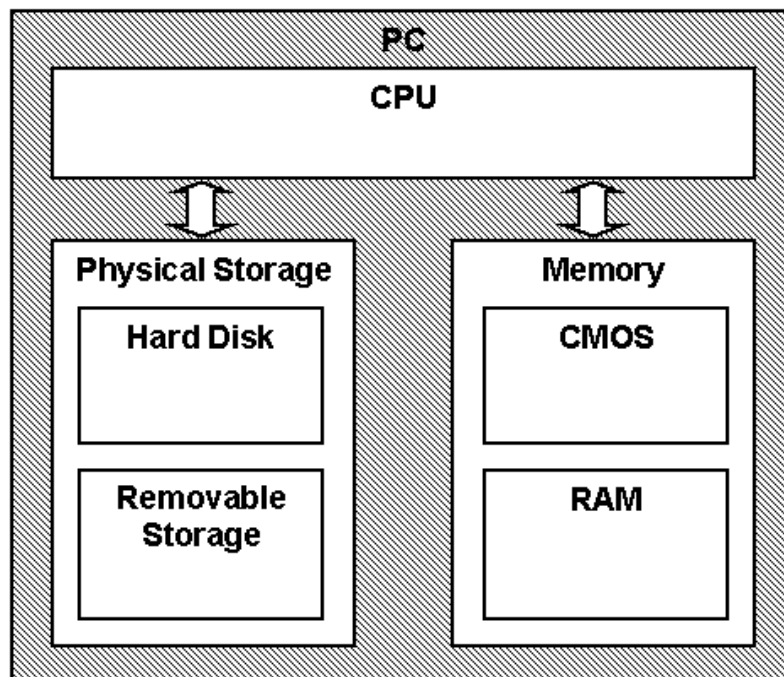
Note that the validated platforms listed in section 1.2 include processors that support the SHA Extensions. This module does not implement SHA, but the bounded modules may implement SHA and, therefore, use the SHA Extensions.

2.5 Cryptographic Bypass

Cryptographic bypass is not supported by BitLocker Dump Filter.

2.6 Hardware Components of the Cryptographic Module

The physical boundary of the module is the physical boundary of the computer that contains the module. The following diagram illustrates the hardware components used by the BitLocker Dump Filter module:



Note: The CPU provides Processor Algorithm Accelerator (PAA)

3 Cryptographic Module Ports and Interfaces

3.1 Control Input Interface

The BitLocker Dump Filter module's control input interface consists of parameter interfaces for the GetFveContext and DumpWrite functions. These interfaces are not exported, but rather, are internal to the cryptographic module.

3.1.1 GetFveContext

```
NTSTATUS GetFveContext(  
    __in PFILTER_EXTENSION Context,  
    __in ULONG MaxPagesPerWrite,  
    __inout_xcount(FveContext->StructureSize) PFVE_CONTEXT FveContext  
)
```

This function gets the BitLocker Full Volume Encryption Key for the storage volume. The Context parameter supplies the dump stack filter context. The FveContext parameter supplies the internal BitLocker context, which includes the BitLocker status and FVEK in this context so it can be used later when writing data to the volume.

3.1.2 DumpWrite

```
NTSTATUS DumpWrite(  
    PFILTER_EXTENSION Context,  
    PLARGE_INTEGER DiskByteOffset,  
    PMDL Mdl  
)
```

This function uses the FVEK from the Context parameter that is provided by the GetFveContext interface. The DiskByteOffset parameter is used to specify the location on the volume to receive the encrypted output data. The Mdl parameter points to the input data to be encrypted.

3.2 Status Output Interface

The BitLocker Dump Filter status output is a return value of type NTSTATUS that indicates whether the function completed successfully or not.

The BitLocker Dump Filter has no status output interface for self-test errors. If the self-tests pass, the module is loaded. If not, the dump filter securely zeroes out memory for any keys handed to it and unloads itself.

3.3 Data Output Interface

The Data Output Interface is the data returned from the DumpWrite function.

This function is responsible for providing the encrypted content for the crash dump file. Data exits the module in the form of encrypted blocks that may be written to a crash dump file on an encrypted volume.

3.4 Data Input Interface

The Data Input Interface includes the GetFveContext function and DumpWrite function. GetFveContext is responsible for reading the FVEK. DumpWrite accepts the memory blocks to encrypt with the FVEK and the target disk locations for the blocks as input.

4 Roles, Services and Authentication

4.1 Roles

BitLocker Dump Filter is a kernel-mode driver that does not interact with the user through any service therefore the module's functions are fully automatic and not configurable. FIPS 140 validations define formal "User" and "Cryptographic Officer" roles. Both roles can use any BitLocker Dump Filter service.

4.2 Services

BitLocker Dump Filter services are described below. This module does not export any cryptographic functions.

1. **Writing encrypted crash dump data** – This service is executed when the system crashes and must write the crash dump file to an encrypted volume.
2. **Show Status** – The module provides a show status service that is automatically executed by the module to provide the status response of the module either via output to the computer monitor or to log files.
3. **Self-Tests** - The module provides a power-up self-tests service that is automatically executed when the module is loaded into memory.
4. **Zeroizing Cryptographic Material** - This service is executed as part of the module shutdown. See [Cryptographic Key Management](#)

The following table maps the services to their corresponding algorithms and critical security parameters (CSPs) as described in Cryptographic Key Management.

Table 3 Services

Service	Algorithms	CSPs	Invocation
Writing encrypted crash dump data	FIPS 197 AES: AES CBC 128 and 256 bits AES XTS 128 and 256 bits ² AES CCM 256	Full Volume Encryption Key (FVEK)	This service is fully automatic.
Show Status	None	None	This service is fully automatic. This service is executed upon completion of the Control Input Interfaces.
Self-Tests	AES-CBC - Encrypt/Decrypt KATs AES-CCM - Encrypt/Decrypt KATs Software Integrity Test (2048-bit RSA with SHA-256) AES XTS KAT	None	This service is fully automatic.

² The length of the data unit does not exceed 2²⁰ AES blocks for storage applications such as BitLocker.

Zeroizing Cryptographic Material (see Section 9)	None	Full Volume Encryption Key (FVEK)	This service is fully automatic.
--	------	-----------------------------------	----------------------------------

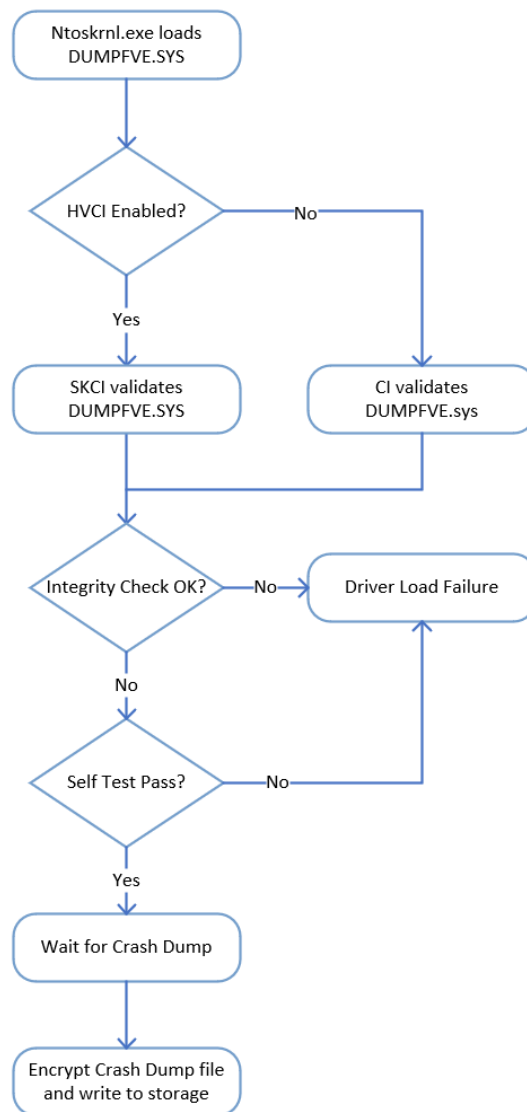
4.3 Authentication

The module does not provide authentication. Roles are implicitly assumed based on the services that are executed.

5 Finite State Model

5.1 Specification

The following diagram shows the finite state model for BitLocker Dump Filter:



6 Operational Environment

The operational environment for BitLocker Dump Filter is the Windows Server operating system running on a supported hardware platform listed in [section 1.2, Validated Platforms](#).

6.1 Single Operator

The BitLocker Dump Filter is loaded into kernel memory as part of the boot process before the logon component is initialized, and so there necessarily is a single operator for the module.

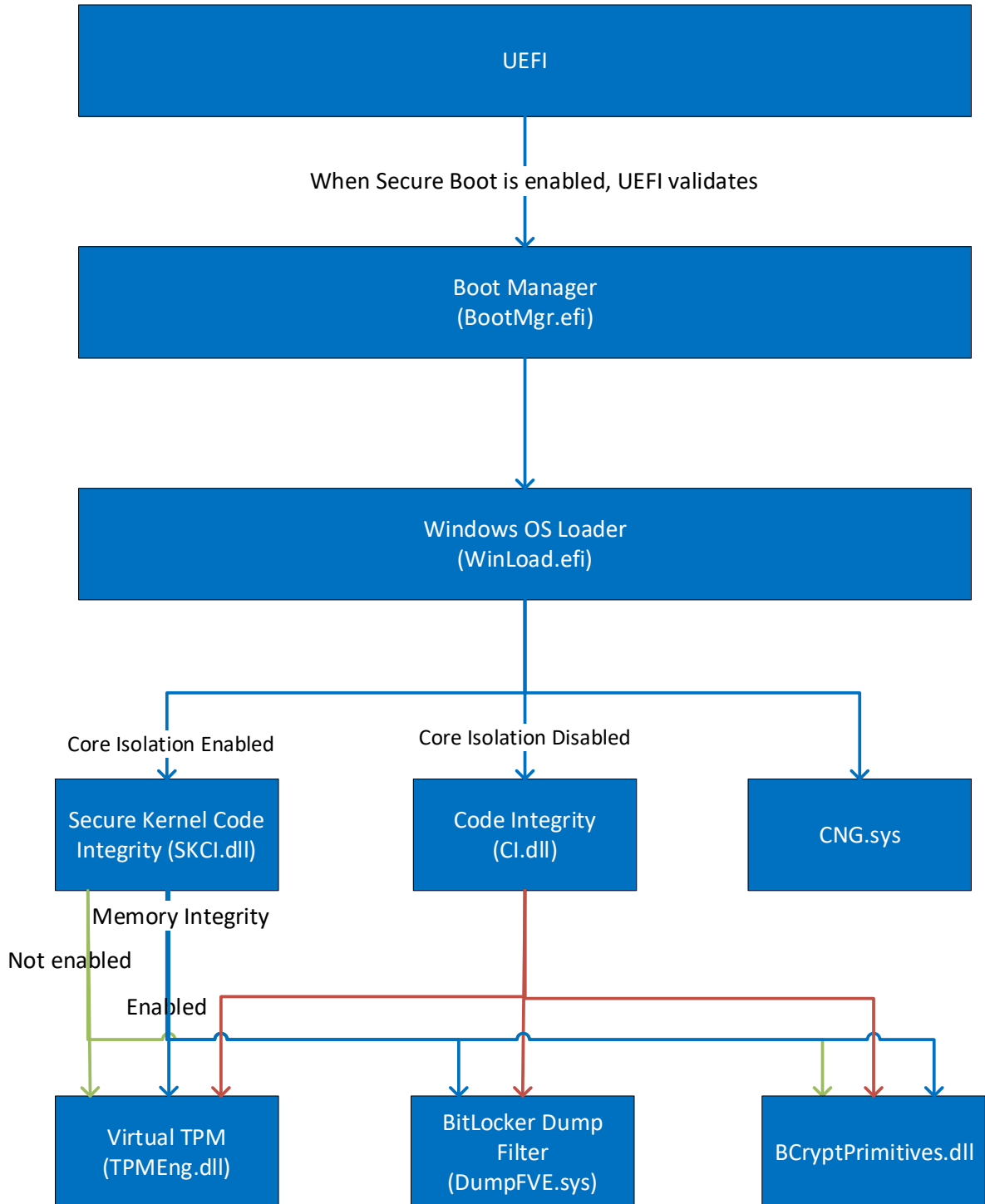
6.2 Cryptographic Isolation

In the Windows operating system, all kernel-mode modules, including DUMPFVE.SYS, are loaded into the Windows Kernel (ntoskrnl.exe) which executes as a single process. The Windows operating system environment enforces process isolation from user-mode processes including memory and CPU scheduling between the kernel and user-mode processes.

6.3 Integrity Chain of Trust

Windows uses several mechanisms to provide integrity verification depending on the stage in the OS boot sequence and also on the hardware and OS configuration. The following diagram describes the Integrity Chain of trust for each supported configuration for the following versions:

- Windows Server 2019 build 10.0.17763.10021 and 10.0.17763.10127



The integrity of the BitLocker Dump Filter module is checked according to the following:

- If Hypervisor Code Integrity (HVCI) is not enabled, then the Code Integrity module performs the integrity check.

- If Hypervisor Code Integrity (HVCI) is enabled then the Secure Kernel Code Integrity module performs the integrity check.

Windows binaries include a SHA-256 hash of the binary signed with the 2048 bit Microsoft RSA code-signing key (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.

7 Cryptographic Key Management

7.1 Critical Security Parameters

When the System Volume is encrypted with Bitlocker, BitLocker Dump Filter uses this critical security parameter (CSP):

- Full Volume Encryption Key (FVEK) - 128 or 256-bit AES key that is used to encrypt dump files.

The FVEK is passed to BitLocker Dump Filter by the Windows Kernel which received the FVEK from the Windows OS Loader modules.

7.2 Zeroization

7.2.1 Volatile Keys

The FVEK is zeroized when the module is unloaded as part of shutting down Windows.

7.2.2 Persistent Keys

BitLocker Dump Filter does not have any persistent keys.

7.3 Access Control Policy

The BitLocker Dump Filter does not allow access to the cryptographic keys contained within it, so an access control table is not included in this document. BitLocker Dump Filter receives keys from outside the module and then manages them appropriately once received. BitLocker Dump Filter prevents access to its keys by zeroizing them after use.

8 Self-Tests

8.1 Power-On Self-Tests

The BitLocker Dump Filter implements Known Answer Test (KAT) functions each time the module is loaded. The module performs the following KATs:

- AES-CBC 128 and 256 Encrypt/Decrypt Known Answer Tests
- AES-CCM 256 Encrypt/Decrypt Known Answer Tests
- XTS-AES 128 and 256 Encrypt/Decrypt Known Answer Tests

If the self-test fails, the module will not load and a status code STATUS_FAIL_CHECK will be returned.

9 Design Assurance

The secure installation, generation, and startup procedures of this cryptographic module are part of the overall operating system secure installation, configuration, and startup procedures for the Windows Server operating system.

The Windows Server 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 Server version downloaded from Windows Update.

An inspection of authenticity of the physical medium can be made by following the guidance at this Microsoft web site: <https://www.microsoft.com/en-us/howtotell/default.aspx>

The installed version of Windows Server must be checked to match the version that was validated. See [Appendix A](#) for details on how to do this.

For Windows Updates, the client only accepts binaries signed with Microsoft certificates. The Windows Update client only accepts content whose signed SHA-2 hash matches the SHA-2 hash specified in the metadata. All metadata communication is done over a Secure Sockets Layer (SSL) port. Using SSL ensures that the client is communicating with the real server and so prevents a spoof server from sending the client harmful requests. The version and digital signature of new cryptographic module releases must be verified to match the version that was validated. See [Appendix A](#) for details on how to do this.

10 Mitigation of Other Attacks

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

Table 10 Mitigation of Other Attacks

Algorithm	Protected Against	Mitigation
AES	Timing Analysis Attack	Constant time implementation
	Cache Attack	Memory Access pattern is independent of any confidential data
		Protected Against Cache attacks only when used with AES NI

11 Security Levels

The security level for each FIPS 140-2 security requirement is given in the following table.

Table 11 Security Levels

Security Requirement	Security Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	1
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	2
Mitigation of Other Attacks	1

BitLocker Dump Filter is a multi-chip standalone software-hybrid module whose host platforms meet the level 1 physical security requirements. The module consists of production-grade 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.

12 Additional Details

For the latest information on Microsoft Windows, check out the Microsoft web site at:

<https://www.microsoft.com/en-us/windows>

For more information about FIPS 140 validations of Microsoft products, please see:

<https://docs.microsoft.com/en-us/windows/security/threat-protection/fips-140-validation>

13 Appendix A – How to Verify Windows Versions and Digital Signatures

13.1 How to Verify Windows Versions

The installed version of Windows Server OEs must be verified to match the version that was validated using the following method:

1. In the Search box type "cmd" and open the Command Prompt desktop app.
2. The command window will open.
3. At the prompt, enter "ver".
4. The version information will be displayed in a format like this:
Microsoft Windows [Version 10.0.xxxxx]

If the version number reported by the utility matches the expected output, then the installed version has been validated to be correct.

13.2 How to Verify Windows Digital Signatures

After performing a Windows Update that includes changes to a cryptographic module, the digital signature and file version of the binary executable file must be verified. This is done like so:

1. Open a new window in Windows Explorer.
2. Type "C:\Windows\" in the file path field at the top of the window.
3. Type the cryptographic module binary executable file name (for example, "CNG.SYS") in the search field at the top right of the window, then press the Enter key.
4. The file will appear in the window.
5. Right click on the file's icon.
6. Select Properties from the menu and the Properties window opens.
7. Select the Details tab.
8. Note the File version Property and its value, which has a number in this format: xx.x.xxxxx.xxxx.
9. If the file version number matches one of the version numbers that appear at the start of this security policy document, then the version number has been verified.
10. Select the Digital Signatures tab.
11. In the Signature list, select the Microsoft Windows signer.
12. Click the Details button.
13. 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.