



Non-Proprietary FIPS 140-2 Security Policy

Document Version: 1.2

Date: July 05, 2023

Table of Contents

1	Introduction	4
1.1	Module Description and Cryptographic Boundary	5
1.2	Mode of Operation.....	6
2	Cryptographic Functionality.....	7
2.1	FIPS 140-2 Approved Algorithms from Bounded Modules	7
2.2	Critical Security Parameters	8
3	Roles, Authentication and Services	8
3.1	Assumption of Roles.....	8
3.2	Authentication.....	8
3.3	Services.....	8
4	Self-Tests	10
5	Operational Environment	10
6	Security Rules and Guidance.....	11
7	Cryptographic Officer Guidance	11
8	References and Definitions.....	14

List of Tables

Table 1 – Security Level of Security Requirements4

Table 2 – Tested Configurations5

Table 3 – Approved Algorithms7

Table 4 – Allowed Non-Approved Algorithms7

Table 5 – Critical Security Parameters (CSPs)8

Table 6 – Services8

Table 7 – References14

Table 8 – Acronyms and Definitions15

List of Figures

Figure 1 – Module Logical diagram6

1 Introduction

The SecureData Engine Cryptographic Module (hereafter referred to as the “Module”) is a FIPS 140 software only cryptographic module from SecureAge Technology which provides transparent and automatic file and folder data encryption of individual files at their inception and without user deliberation, action, or even awareness. Inherent and Invisible PKI encryption ensures the confidentiality of the data at rest whether leaked, lost, or stolen. The Module is a Microsoft Windows file system filter driver intended for use by US and Canadian Federal agencies or other markets that require FIPS 140-2 validated file encryption.

The current version of the SecureData Engine is 8.0.5 and is implemented as the SecureData.sys kernel driver.

The FIPS 140-2 security levels for the Module are as follows:

Table 1 – Security Level of Security Requirements

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	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A
Overall	1

The Module is a software module and does not implement attack mitigations outside the scope of [140], hence [140] section 4.5 *Physical Security* and section 4.11 *Mitigation of Other Attacks* are not applicable per [140IG] G.3 *Partial Validations and Not Applicable Areas of FIPS 140-2*.

Operational testing was performed for the following configurations:

Table 2 – Tested Configurations

	OS	Platform	Processor	CNG.SYS Cert. #	CI.DLL Cert. #
1	Microsoft Windows 10 Pro 64-bit version 1803 build 10.0.17134	Microsoft Surface Pro 6	Intel® Core i5-8250U with AES-NI	#3196	#3195
2*	Microsoft Windows 10 Home 64- bit build 10.0.19041	Lenovo X1 Carbon	Intel® Core i7-8550U with AES-NI		
3*	Microsoft Windows 10 Pro 64-bit version 20H2 build 10.0.19042	Dell Vostro 220	Intel® Pentium Dual Core E5300 without AES-NI		

* Note: The module was tested on these platforms but as of the date of this document these versions of Microsoft Windows do not have a CMVP validation certificate. The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when used in an operational environment not listed on the validation certificate.

The Module conforms to [140IG] 6.1 *Single Operator Mode and Concurrent Operators*. The tested environments are restricted to a single operator (concurrent operators are explicitly excluded). The corresponding user mode application and Operating System that calls into the module to use the cryptographic services is the single user of the module.

1.1 Module Description and Cryptographic Boundary

The module is designed so it can support AES natively or by utilizing the AES-NI PAA capability if available. Per IG 1.21 it therefore defined as a Software/Firmware module Embodiment.

The Module conforms to [140IG] 1.16 *Software Module*:

- The physical cryptographic boundary is the General Purpose Computer (GPC) which wholly contains the module and operating system (i.e., physical embodiment is multi-chip standalone).
- The logical cryptographic boundary is the set of software components that implement the cryptographic mechanisms listed in Table 10. This is packaged as SecureData.sys.
- The module's ports are those of the GPC on which the module executes. All logical interfaces are directed through the modules logical interface (API).
- The power-up approved integrity test is performed without user action over all components within the logical boundary.
- Updates to the Module are provided as a complete replacement and constitute an entirely new module.

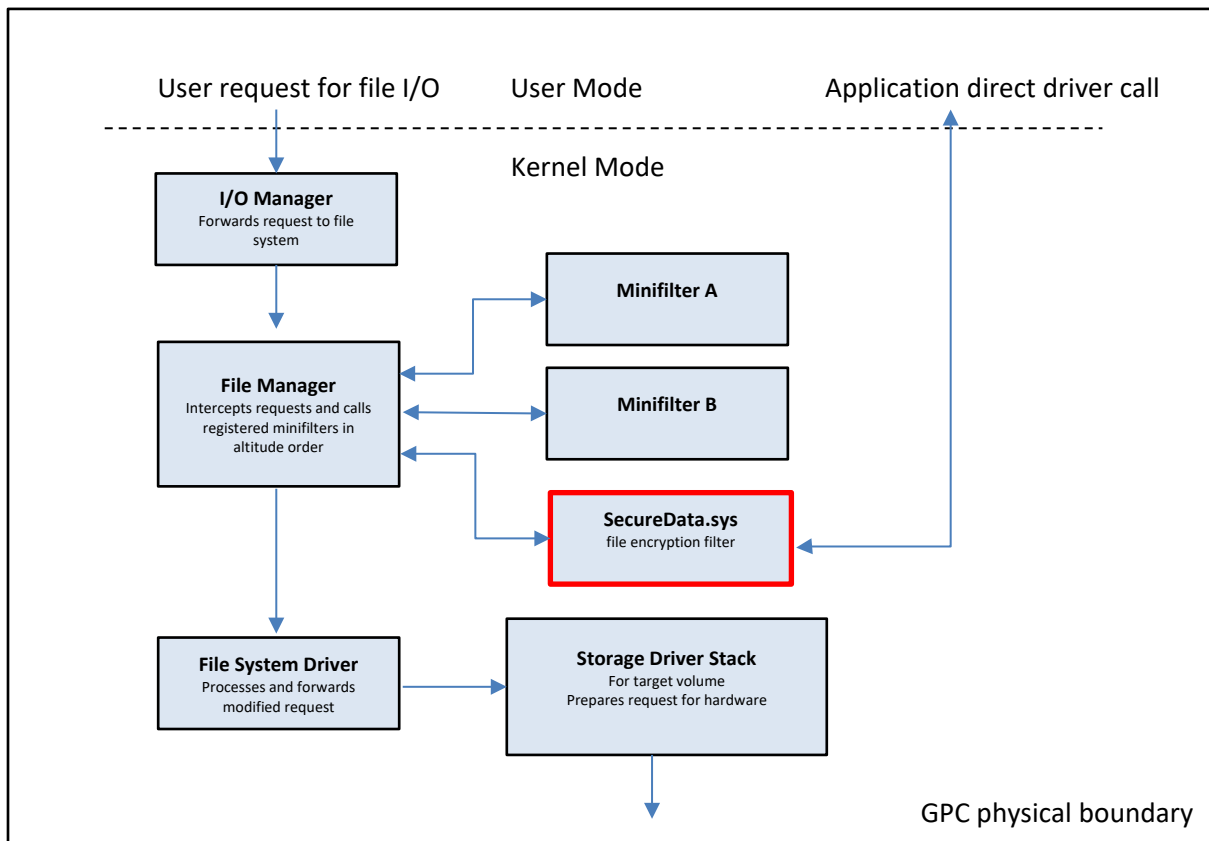


Figure 1 – Module Logical diagram

1.2 Mode of Operation

This module is bound to Microsoft modules described below in Section 2.1. As described in the Security Policy of the bound modules, the following configurations and modes of operation will cause the module to operate in a non-approved mode of operation:

- Boot Windows in Debug mode
- Boot Windows with Driver Signing disabled
- Windows enters the ACPI S4 power state

To operate in FIPS approved mode, this module must also be configured as directed in Section 7. All files on the target system encrypted using a previous version of the module should be unencrypted and re-encrypted using this version in order to operate in an approved mode.

2 Cryptographic Functionality

The Module implements the following FIPS Approved functions certified under CAVP Cert. #[A1875](#).

Table 3 – Approved Algorithms

Algorithm	Mode	Description	Functions/Caveats
AES [197]	ECB, CBC [38A]	Key Sizes: 256	Encrypt, Decrypt
	XTS ¹ [38E]	Key Sizes: 256	Encrypt, Decrypt
CKG [IG D.12] Vendor Affirmed	[133] Section 7.1 Direct symmetric key generation using unmodified DRBG output		
DRBG [90A]	Hash	SHA-256, 384, 512	Deterministic Random Bit Generation Security Strength = 256
SHS [180]	SHA-1, SHA-256, SHA-384, SHA-512		Message Digest Generation

The Module utilizes the following non-approved but allowed functions.

Table 4 – Allowed Non-Approved Algorithms

Algorithm	Description
AES [197] CBC [38E] (no security claimed)	Non-compliant key generation provides obfuscation of API parameters.
HMAC [198] (no security claimed)	Hardcoded key provides integrity checking of file header.
KAS (no security claimed)	Non-compliant key agreement for obfuscation of API parameters.

Note the module implements a non-compliant version of AES-XTS which may be used if it is improperly configured (see Section 7).

2.1 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. The SecureData Engine depends on the following modules and algorithms:

- Microsoft Kernel Mode Cryptographic Primitives (CNG.SYS module FIPS 140-2 Certificate #3196) for obtaining random numbers via BCryptGenRandom and as a result utilizes the following algorithms:
 - AES-256 counter mode – CAVP Cert. #5847
 - AES-256 counter mode DRBG – CAVP Cert. #2435
 - Non-Approved but allowed NDRNG
- Microsoft Code Integrity (CI.DLL module FIPS 140-2 Certificate #3195) for software integrity test:
 - FIPS 180-4 SHS SHA-256 – CAVP Cert. #4633
 - FIPS 186-4 RSA PKCS#1 (v1.5) Sig Ver with 2048 moduli (SHA-256) – CAVP Cert. #3080

¹ Per IG A.9, the XTS algorithm implementation includes an explicit check to ensure Key₁ ≠ Key₂

2.2 Critical Security Parameters

All CSPs used by the Module are described in this section. All usage of these CSPs by the Module (including all CSP lifecycle states) is described in the services detailed in [Section 4](#).

Table 5 – Critical Security Parameters (CSPs)

CSP	Description / Usage
DRBG-EI	Hash_DRBG entropy input.
DRBG-State	Hash_DRBG internal seed and state (V and C)
Session Key	XTS-AES-256 key used to encrypt the file contents
User Key	XTS-AES-256 key used to encrypt the Session Key

3 Roles, Authentication and Services

3.1 Assumption of Roles

The module supports the following roles:

- Cryptographic Officer: performs the module installation and associated configuration.
- User: performs all normal operations and services.

3.2 Authentication

The module is a Level 1 software-only cryptographic module and does not implement authentication. The user role is implicitly assumed for the services provided.

3.3 Services

The module provides services directly to a user mode application as well as to the operating system to implement filter driver functionality for filesystem calls. Table 6 lists those services and identifies their access to Security Parameters. Note: Input and Output are from the perspective of the logical API boundary. The module does not Input or Output CSPs across the physical boundary.

- G = Generate: The service generates the CSP.
- O = Output: The service outputs the CSP.
- E = Execute: The service uses the CSP in an algorithm.
- I = Input: The service inputs the CSP.
- Z = Zeroize: The service zeroizes the CSP.

Table 6 – Services

Service	Description	DRBG-EI	DRBG-State	Session Key	User Key
IOCTL_SECUREDATA_SA	Indication of SecureAge software running				
IOCTL_SECUREDATA_INIT_RANDOM	Initialize the random generator	IE	G		
IOCTL_SECUREDATA_LOGIN	Send login keys to SecureData				I
IOCTL_SECUREDATA_LOGOUT	Remove user login keys				Z

Service	Description	DRBG- EI	DRBG- State	Session Key	User Key
IOCTL_SECUREDATA_SET_LOGIN_USER	Set the default user ID				
IOCTL_SECUREDATA_GET_LOGIN_INFO	Get user login status				
IOCTL_SECUREDATA_GET_FILE_INFO	Get file encryption status				
IOCTL_SECUREDATA_GET_USER_KEY_LENGTH	Get the key length defined for encrypting user key				
IOCTL_SECUREDATA_GET_CFG	Get filter configurations				
IOCTL_SECUREDATA_PROCESS_FILE	Encrypt/Decrypt/Copy a file	IE	GE	GEZ	EZ
IOCTL_SECUREDATA_MIGRATE_FILE	Migrate a file	IE	GE	GEZ	EZ
IOCTL_SECUREDATA_COUNT_FILES	Count files in a directory				
IOCTL_SECUREDATA_GET_NEXT_DIR	Get next folder in the enumeration process				
IOCTL_SECUREDATA_CIPHER_KEY_REQUEST	Decrypt user key request from kernel mode				I
IOCTL_SECUREDATA_IS_REMOVABLE_MEDIA	Check if the drive is a removable drive				
IOCTL_SECUREDATA_GET_MESSAGE	Get message for display on screen				
IOCTL_SECUREDATA_MOVE_FOLDER_GET_INFO	Get progress in moving folder				
IOCTL_SECUREDATA_MOVE_FOLDER_GET_CURRENT_FILE_INFO	Get current file progress in moving folder				
IOCTL_SECUREDATA_APPCTRL_PROMPT	Get message to prompt in application control				
IOCTL_SECUREDATA_SET_INSTALL	Inform driver of SecureAge software installation				
IOCTL_SECUREDATA_SET_UNINSTALL	Inform driver of SecureAge software uninstallation				
IOCTL_SECUREDATA_SECURE_PARAM	Setup protocols for encryption of IOCTL parameters				
IOCTL_SECUREDATA_ALGO_TEST	Run algorithm test using input JSON file				
IOCTL_SECUREDATA_ALGO_TEST_STATUS	Get the startup algorithm test status				
IOCTL_SECUREDATA_UPDATE_DRIVER	Update the driver file				
IRP_MJ_CREATE	Read file header after the file is created or opened	IE	IE	GIO	E
IRP_MJ_READ	Decrypt the data read			E	
IRP_MJ_WRITE	Encrypt the data to be written			E	
IRP_MJ_QUERY_INFORMATION	Adjust file attributes as appropriate				
IRP_MJ_SET_INFORMATION	Set file metadata (potentially (de)encrypt on move)			E	
IRP_MJ_DIRECTORY_CONTROL	Adjust folder and file metadata				
IRP_MJ_NETWORK_QUERY_OPEN	Control file operations if needed				
IRP_MJ_FILE_SYSTEM_CONTROL	Control file operations if needed				
IRP_MJ_DEVICE_CONTROL	Control file operations if needed				
IRP_MJ_CLEANUP	Write file header				
IRP_MJ_CLOSE	Control file operations if needed				
ZEROIZE	CO shutdown of GPC	Z	Z	Z	Z

4 Self-Tests

The module performs self-tests to ensure the proper operation of the module. Per FIPS 140-2 these are categorized as either power-up self-tests or conditional self-tests. Power up self-tests are available on demand by power cycling the module. Upon instantiation the kernel driver entry point calls the power up self-tests which results in their execution without operator interaction and prior to the use of any cryptographic functionality.

Software integrity is performed by bound module Microsoft Code Integrity over all software components using RSA 2048 Sig Ver (SHA-256).

The module then performs the following algorithm KATs on power-up.

- SHA-1, SHA-256, SHA-384, SHA-512
- AES-ECB-256 encrypt and decrypt
- AES-XTS-256 encrypt and decrypt
- Hash_DRBG Health Tests per SP 800-90A prior to first use
 - Known Answer Test
 - Instantiate Function
 - Generate Function

The module implements an APT on the BCryptGenRandom output from the bound module which is used to see the internal DRBG. The module does not perform any other critical functions tests.

All algorithm Known Answer Tests (KATs) are completed successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module enters the error state and outputs an error indicator as described below, otherwise it indicates successful completion, the indicator is also described below.

Status is returned through the API but is observable to the end user through the user mode tray application. The tray application “System Settings” tab shows 3 separate status:

- Cryptographic Algorithms:
- Random Number Generator:
- XTS Algorithm:

The Cryptographic Algorithms in use are listed and the font will normally be Blue if configured in Approved mode, otherwise green or purple. If a KAT has failed it will be Red. When the module has been configured for FIPS mode the XTS algorithm will be listed as “NIST AES-256 XTS”. If the KAT has passed this will show up in blue otherwise it will be red. In either error condition the module will no longer perform any cryptographic operations. This will cause all file access configured for encryption to fail.

The Random Number Generator will normally display “NIST DRBG” in blue. If the DRBG health tests fail the module will use the output of the BCryptGenRandom directly and the text “Error encountered in RNG” will be displayed in red color. If the BCryptGenRandom fails, the module will enter a hard error state and will no longer perform any cryptographic operations. This will cause all file access configured for encryption to fail.

5 Operational Environment

The Module has a modifiable operational environment under the FIPS 140-2 definitions. Module validation compliance is affirmed in accordance with [140IG] G.5 on any general purpose platform utilizing the Windows10 operational environment on an Intel x86 processor (thereby not requiring recompilation),

however per [IG] G.5 no claim can be made as to the correct operation of the module or the security strengths of the generated keys.

6 Security Rules and Guidance

This section documents the security rules for the secure operation of the cryptographic module to implement the security requirements of FIPS 140-2.

1. The module provides two implicitly assumed operator roles: User and Cryptographic Officer.
2. The Cryptographic Officer is the admin that installs the module.
3. The module does not perform authentication, roles are implicitly assumed.
4. An operator does not have access to any cryptographic services prior to assuming an authorized role.
5. The module allows the operator to initiate power-up self-tests by power cycling power or resetting the module.
6. Power up self-tests do not require any operator action.
7. All information is provided through the logical interfaces of the module which is inhibited during key generation, self-tests, zeroization, and error states.
8. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
9. The module does not persistently store plaintext keys or CSPs.
10. Zeroization of volatile memory is performed by powering off the module.
11. The module does not support concurrent operators.
12. The module does not support a maintenance interface or role.
13. The module does not support manual key entry.
14. The module does not have any proprietary external input/output devices used for entry/output of data.

7 Cryptographic Officer Guidance

Crypto Officers use the Installation instructions to install the Module in their environment. As part of that procedure the configuration of the module to operate in an approved mode requires the CO to perform the following:

Configure the Microsoft Kernel Mode Cryptographic Primitives Library (Cert. #3196) for FIPS mode:

Use the FIPS Local/Group Security Policy setting or a Mobile Device Management (MDM) to enable FIPS Approved mode for Kernel Mode Cryptographic Primitives Library. For all Windows versions listed in Table 2 of the validated platforms (except Windows 10 Mobile), use the FIPS Local/Group Security Policy. For Windows 10 Mobile, use Mobile Device Management (MDM).

The Windows operating system provides a group (or local) security policy setting, "System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing".

Note: The following configurations and modes of operation will cause the module to operate in a non-approved mode of operation:

- Boot Windows in Debug mode
- Boot Windows with Driver Signing disabled
- Windows enters the ACPI S4 power state

Configure the SecureAge Driver for FIPS installation:

For FIPS approved mode, the proper attributes must be set in the autoconf.ini file (located in the Win32 or x64 setup folder – e.g., C:\Setup\SecureAge v8\x64) and then run the installer. This can be done manually or by executing SecureDataCfg.exe (for 32-bit version), or SecureDataCfg64.exe (for 64-bit version):

Method 1: Explicitly specify the NIST AES-XTS algorithm ID = 38. Note: Currently the only other algorithm is a legacy SecureAge AES-XTS which is 37.

```
[SecureData]
```

```
KeyAlgo = 38
```

```
FileAlgo = 38
```

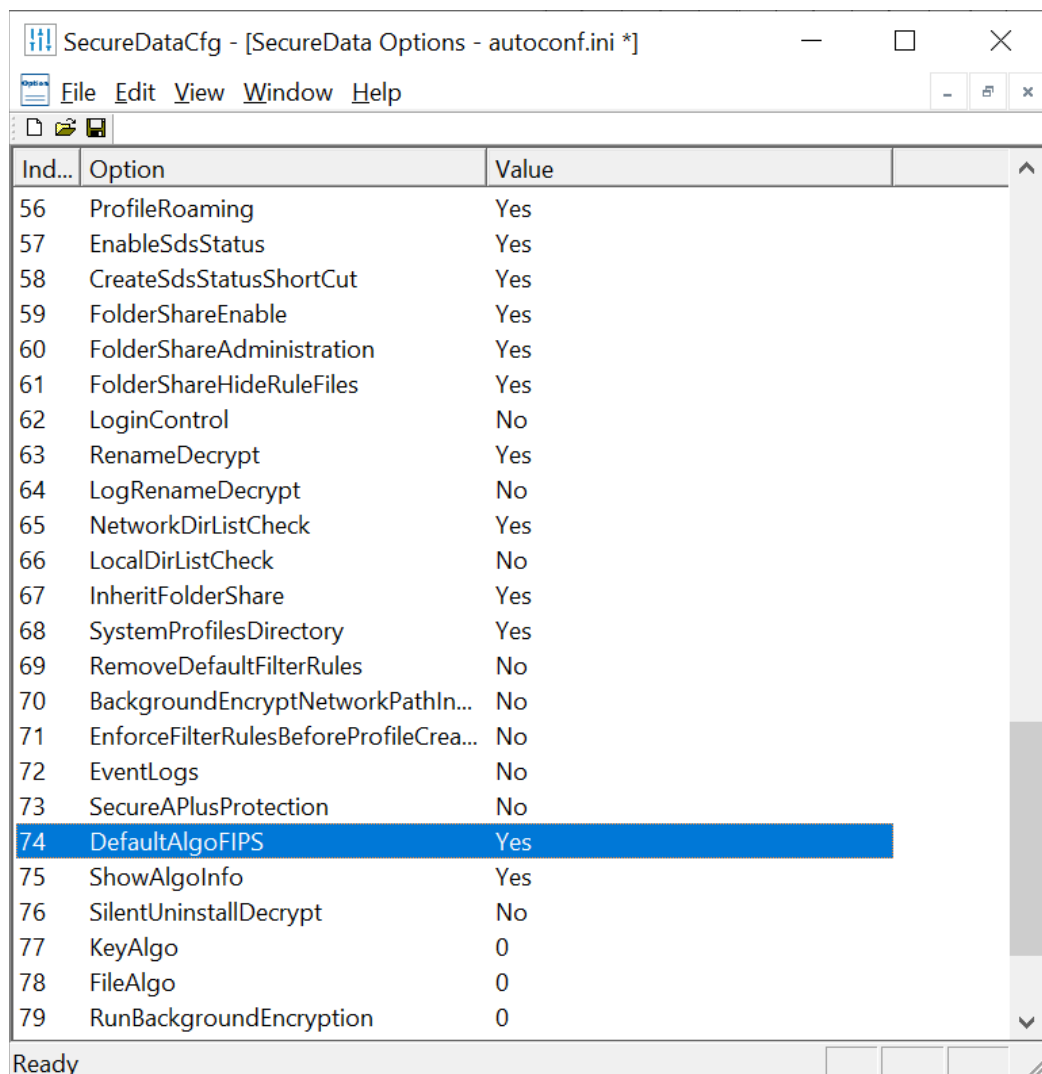
Method 2: Set the algorithm ID to zero to use default algorithms, but set the default to NIST approved by setting DefaultAlgoFIPS to true.

```
[SecureData]
```

```
DefaultAlgoFIPS = true
```

```
KeyAlgo = 0
```

```
FileAlgo = 0
```



The screenshot shows a window titled "SecureDataCfg - [SecureData Options - autoconf.ini *]". The window contains a table with the following data:

Ind...	Option	Value
56	ProfileRoaming	Yes
57	EnableSdsStatus	Yes
58	CreateSdsStatusShortCut	Yes
59	FolderShareEnable	Yes
60	FolderShareAdministration	Yes
61	FolderShareHideRuleFiles	Yes
62	LoginControl	No
63	RenameDecrypt	Yes
64	LogRenameDecrypt	No
65	NetworkDirListCheck	Yes
66	LocalDirListCheck	No
67	InheritFolderShare	Yes
68	SystemProfilesDirectory	Yes
69	RemoveDefaultFilterRules	No
70	BackgroundEncryptNetworkPathIn...	No
71	EnforceFilterRulesBeforeProfileCrea...	No
72	EventLogs	No
73	SecureAPlusProtection	No
74	DefaultAlgoFIPS	Yes
75	ShowAlgoInfo	Yes
76	SilentUninstallDecrypt	No
77	KeyAlgo	0
78	FileAlgo	0
79	RunBackgroundEncryption	0

To apply the configuration, it is necessary to run the installer once.

Note if the module is installed for the first time on a system and configured for FIPS mode then it will only perform approved services (i.e., utilize approved algorithms). However, if files already exist which are encrypted using a non-approved algorithm (from a previous version or an improperly configured installation) they should be decrypted then re-encrypted, otherwise they will remain encrypted with the non-approved algorithm.

8 References and Definitions

The following standards are referred to in this Security Policy.

Table 7 – References

Abbreviation	Full Specification Name
[FIPS140-2]	<i>Security Requirements for Cryptographic Modules, May 25, 2001</i>
[IG]	<i>Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program</i>
[108]	<i>NIST Special Publication 800-108, Recommendation for Key Derivation Using Pseudorandom Functions (Revised), October 2009</i>
[131A]	<i>NIST Special Publication 800-131A Revision 2, Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, March 2019</i>
[132]	<i>NIST Special Publication 800-132, Recommendation for Password-Based Key Derivation, Part 1: Storage Applications, December 2010</i>
[133]	<i>NIST Special Publication 800-133 Revision 2, Recommendation for Cryptographic Key Generation, June 2020</i>
[135]	<i>National Institute of Standards and Technology, Recommendation for Existing Application-Specific Key Derivation Functions, Special Publication 800-135rev1, December 2011.</i>
[180]	<i>National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, August, 2015</i>
[186]	<i>National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4, July 2013.</i>
[197]	<i>National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 26, 2001</i>
[198]	<i>National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, July, 2008</i>
[38A]	<i>National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001</i>
[38B]	<i>National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, Special Publication 800-38B, May 2005</i>
[38C]	<i>National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, Special Publication 800-38C, May 2004</i>
[38D]	<i>National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, Special Publication 800-38D, November 2007</i>

Abbreviation	Full Specification Name
[38E]	<i>National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices, Special Publication 800-38E, January 2010</i>
[38F]	<i>National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, Special Publication 800-38F, December 2012</i>
[56Ar3]	<i>NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, April 2018</i>
[56Br2]	<i>NIST Special Publication 800-56B Revision 2, Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography, March 2019</i>
[90A]	<i>NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90A, June 2015.</i>
[90B]	<i>NIST Special Publication 800-90B Revision 1, Recommendation for the Entropy Sources Used for Random Bit Generation, Special Publication 800-90B, January 2018.</i>

Table 8 – Acronyms and Definitions

Acronym	Definition
IOCTL	Input/Output Control
MJ	Major
IRP	Input/Output Request Packet