Security Policy

for FIPS 140-2 Validation

Microsoft Windows 8

Microsoft Windows Server 2012

Microsoft Windows RT

Microsoft Surface Windows RT

Microsoft Surface Windows 8 Pro

Microsoft Windows Phone 8

Microsoft Windows Storage Server 2012

Boot Manager

DOCUMENT INFORMATION

Version Number	1.3
Updated On	December 17, 2014

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.

TABLE OF CONTENTS

1 Introduction

The Windows Boot Manager is the system boot manager, called by the bootstrapping code that resides in the boot sector. Boot Manager is responsible for capturing the credentials required to unlock the OS volume and for loading and verifying the integrity of the Windows OS Loader, Winload.exe, and Windows OS Resume, winresume.exe. The Boot Manager consists of these binary executables and the Certificate Directory containing the root public key certificate issued by Microsoft.

> Note: credentials only pertain to unlocking the OS volumes. Credentials are not used for authentication to control access to cryptographic module ports and services.

In Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8, the credentials supported for unlocking the OS volumes are:

- A startup key (stored on a USB flash drive; also known as an External key)
- A recovery key (stored on a USB flash drive; also known as an External key)
- An alpha-numeric PIN for Trusted Platform Module (TPM)+PIN or TPM+PIN+ Startup Key (SK) scenarios
- A password
- A key provided over a network by a trusted server

1.1 List of Cryptographic Module Binary Executables

Boot Manager crypto module binaries include:

- BOOTMGR
- bootmgr.exe
- bootmgfw.efi
- bootmgr.efi

The version numbers are:

Version 6.2.9200 for Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8

1.2 Brief Module Description

Both the older PC/AT BIOS and the newer Unified Extensible Firmware Interface (UEFI) boot methods are supported.

BOOTMGR and bootmgr.exe are the binary executables for booting PC/AT BIOS systems. BOOTMGR logically encapsulates bootmgr.exe.

Bootmgfw.efi and bootmgr.efi are the binary executables for booting Unified Extensible Firmware Interface (UEFI) systems. Bootmgfw.efi logically encapsulates bootmgr.efi.

1.3 Validated Platforms

The Boot Manager components listed in Section 1.1 were validated using the following machine configurations:

x86 Microsoft Windows 8 Enterprise – Dell Dimension C521 (AMD Athlon 64 X2 Dual Core)
x64 Microsoft Windows 8 Enterprise – Dell PowerEdge SC430 (Intel Pentium D without AES-NI)
x64-AES-NI Microsoft Windows 8 Enterprise – Intel Client Desktop (Intel Core i7 with AES-NI)
x64 Microsoft Windows Server 2012 – Dell PowerEdge SC430 (Intel Pentium D without AES-NI)
x64-AES-NI Microsoft Windows Server 2012 – Intel Client Desktop (Intel Core i7 with AES-NI)
ARMv7 Thumb-2 Microsoft Windows RT – NVIDIA Tegra 3 Tablet (NVIDIA Tegra 3 Quad-Core)
ARMv7 Thumb-2 Microsoft Windows RT – Qualcomm Tablet (Qualcomm Snapdragon S4)
ARMv7 Thumb-2 Microsoft Windows RT – Microsoft Surface Windows RT (NVIDIA Tegra 3 Quad-Core)
x64-AES-NI Microsoft Windows 8 Pro – Microsoft Surface Windows 8 Pro (Intel x64 Processor with AES-NI)
ARMv7 Thumb-2 Microsoft Windows Phone 8 – Windows Phone 8 (Qualcomm Snapdragon S4)
x64 Microsoft Windows Storage Server 2012 – Intel Maho Bay (Intel Core i7 without AES-NI)
x64-AES-NI Microsoft Windows Storage Server 2012 – Intel Maho Bay (Intel Core i7 with AES-NI)

Boot Manager maintains FIPS 140-2 validation compliance (according to FIPS 140-2 PUB Implementation Guidance G.5) on the following platforms:

x86 Microsoft Windows 8
x86 Microsoft Windows 8 Pro

x64 Microsoft Windows 8
x64 Microsoft Windows 8 Pro
x64 Microsoft Windows Server 2012 Datacenter

x64-AES-NI Microsoft Windows 8
x64-AES-NI Microsoft Windows 8 Pro
x64-AES-NI Microsoft Windows Server 2012 Datacenter

1.4 Cryptographic Boundary

The software cryptographic boundary for Boot Manager is defined as the binaries BOOTMGR, bootmgr.exe, bootmgfw.efi, and bootmgr.efi. The physical configuration of Boot Manager, as defined in FIPS-140-2, is multi-chip standalone.

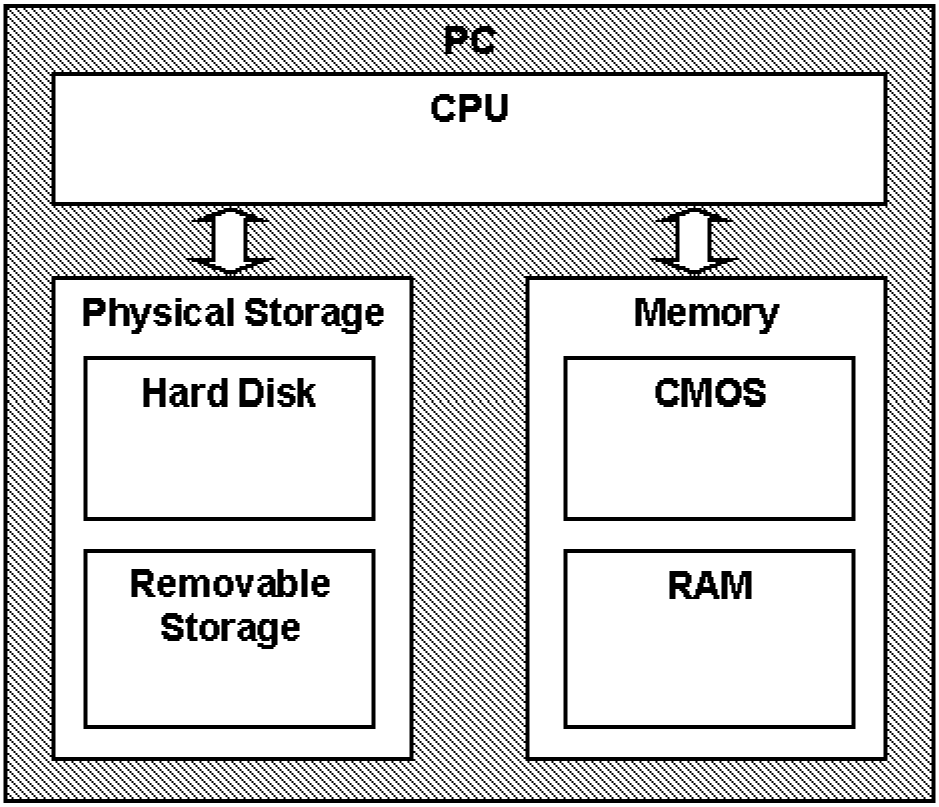
2 Security Policy

Boot Manager operates under several rules that encapsulate its security policy.

- Boot Manager is validated on the platforms listed in Section 1.3.
- Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8 are operating systems supporting a “single user” mode where there is only one interactive user during a logon session.

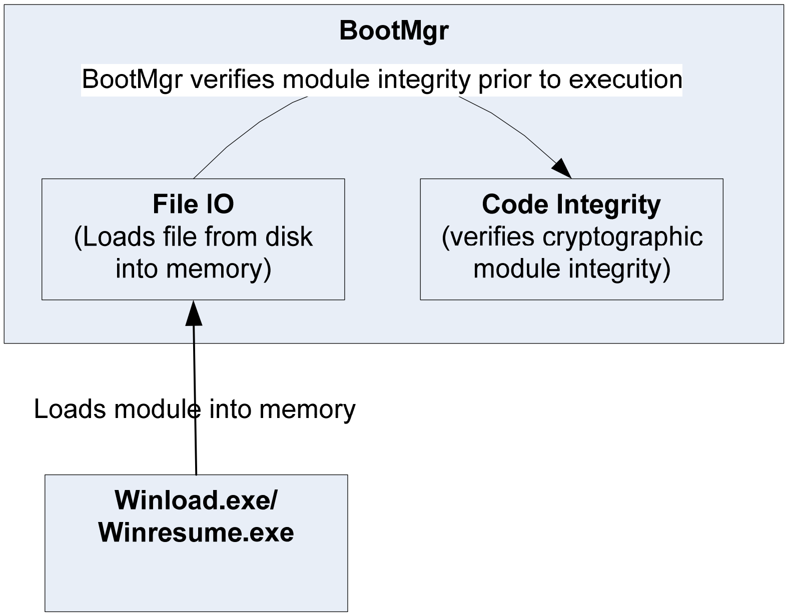
- Boot Manager is only in its Approved mode of operation when Windows is booted normally, meaning Debug mode is disabled and Driver Signing enforcement is enabled.
- The Debug mode status and Driver Signing enforcement status can be viewed by using the bcdedit tool.

The following diagram illustrates the master components of the Boot Manager module:



Boot Manager

The following diagram illustrates Boot Manager module interaction with the cryptographic module:


BootMgr
BootMgr verifies module integrity prior to execution
File IO
(Loads file from disk into memory)
Code Integrity
(verifies cryptographic module integrity)
Loads module into memory
Winload.exe/
Winresume.exe

- Boot Manager loads the Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8 operating system loader (Winload.exe) or winresume.exe, after it determines component's integrity using its cryptographic algorithm implementations using the FIPS 140-2 approved algorithms mentioned below. After the verified binary image file is loaded, Boot Manager passes the execution control to it and no longer executes until the next reboot. The Crypto officer and User have access to the services Boot Manager supports.
- If the integrity of components being loaded is not verified, Boot Manager does not transfer the execution from itself.
- Boot Manager has a service for the encryption and decryption functionality used with BitLocker® Drive Encryption[1] operations related to bootstrapping the Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8 operating system.
- The module provides a power-up self-tests service that is automatically executed when the module is loaded into memory, as well as, a show status service, that is automatically executed by the module to provide the status response of the module either via output to the general purpose computer (GPC) monitor or to log files.
- Boot Manager implements a self-integrity check using an RSA digital signature during its initialization process. Boot Manager will not complete its initialization if the signature is invalid.

2.1 FIPS 140-2 Approved Algorithms

- Boot Manager implements the following FIPS-140-2 Approved algorithms.
  - RSA PKCS#1 (v1.5) digital signature verification (Cert. # 1132)
  - SHS (Cert. # 1903)

[1] BitLocker is a registered trademark for the full volume encryption functionality in Windows. BitLocker is not a separate binary executable.

- AES (Certs. # 2196 and # 2198)
- HMAC (Cert. # 1347)

2.2 Non-Approved Algorithms

- Boot Manager implements the following non-approved algorithms:
  - MD5 – used for certificate chain authentication

2.3 Cryptographic Bypass

Cryptographic bypass is not supported by Boot Manager.

2.4 Machine Configurations

Boot Manager was tested using the machine configurations listed in Section 1.3 - Validated Platforms.

3 Operational Environment

The operational environment for Boot Manager is Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8 running on the hardware listed in Section 1.3 - Validated Platforms.

4 Integrity Chain of Trust

Boot Manager is the very start of the chain of trust. It cryptographically checks its own integrity during its startup. It then cryptographically checks the integrity of the Windows OS Loader (Winload.exe) or Windows OS Resume (Winresume.exe) before starting it.

5 Ports and Interfaces

5.1 Control Input Interface

The Boot Manager Control Input Interface is the set of internal functions responsible for reading control input. These input signals are read from various system locations and are not directly provided by the operator. Examples of the internal function calls include:

- BlBdDebuggerEnabled – Reads the system flag to determine if the boot debugger is enabled.
- BlXmiRead – Reads the operator selection from the Boot Selection menu.
- BlGetBootOptionBoolean – Reads control input from a protected area of the Boot Configuration Data registry.

The GPC's keyboard can also be used as control input when it is necessary for an operator to provide a response to a prompt for input or in response to an error indicator.

5.2 Status Output Interface

The Status Output Interface is the BlStatusPrint function that is responsible for displaying the integrity verification errors to the screen. The Status Output Interface is also defined as the BsdpWriteAtLogOffset responsible for writing the name of the corrupt driver to the bootlog.

5.3 Data Output Interface

The Data Output Interface includes the following functions: Archx86TransferTo32BitApplicationAsm, Archx86TransferTo64BitApplicationAsm, and Archpx64TransferTo64BitApplicationAsm. These functions are responsible for transferring the execution from Boot Manager to the initial execution point of the Windows OS Loader or Windows OS Resume. Data exits the module in the form of the initial instruction address of Winload.exe or Winresume.exe.

5.4 Data Input Interface

The Data Input Interface includes the BlFileReadEx function. BlFileReadEx is responsible for reading the binary data of unverified components from the computer hard drive.

Additionally, the GPC's USB port also forms a part of the Data Input interface. This interface is used to enter the Startup key or Recovery Key used by the BitLocker® Drive Encryption in Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8. The GPC's keyboard can also serve as a Data Input Interface when the method to protect the Volume Master Key (VMK) value relies on an operator supplied PIN.

6 Specification of Roles

Boot Manager supports both User and Cryptographic Officer roles (as defined in FIPS-140-2). Both roles have access to all services implemented in Boot Manager. Therefore, roles are assumed implicitly by booting the Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8 operating system.

Services available to the Cryptographic Officer role:

- Configure BitLocker into FIPS mode
- Unlock the operating system volume
- Boot the Windows operating system

Services available to the User role:

- Unlock the operating system volume
- Boot the Windows operating system

6.1 Maintenance Roles

Maintenance roles are not supported.

6.2 Multiple Concurrent Interactive Operators

There is only one interactive operator in Single User Mode. When run in this configuration, multiple concurrent interactive operators are not supported.

6.3 Show Status Services

The User and Cryptographic Officer roles have the same Show Status functionality, which is described in Section 5 Ports and Interfaces.

6.4 Self-Test Services

The User and Cryptographic Officer roles have the same Self-Test functionality, which is described in Section 10 Self-Tests.

6.5 Service Inputs / Outputs

The User and Cryptographic Officer roles have service inputs and outputs as specified in Section 5 Ports and Interfaces.

7 Services

Boot Manager services are:

1. the encryption and decryption functionality used with BitLocker Drive Encryption for file I/O that supports the bootstrapping of the Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8 operating system
2. loading and verifying the integrity of the Windows 8, Windows RT, Windows Server 2012, Windows Storage Server 2012, and Windows Phone 8 operating system loader (winload.exe) or winresume.exe.

The User and Cryptographic Officer roles have the same use of the encryption, decryption, and OS loading services. Boot Manager does not export any cryptographic functions that can be called or externally invoked.

8 Cryptographic Key Management

Note: authentication in the FIPS 140-2 standard pertains exclusively to controlling access to cryptographic module ports and services. This is different the use of the word authentication in a broader information security sense. The keys described here are credentials used to unlock Windows OS volumes.

Boot Manager does not store any secret or private cryptographic keys across power-cycles. However, it does use certain AES keys in support of the BitLocker feature in FIPS mode. These keys are:

- Full Volume Encryption Key (FVEK) – 256-bit AES key used to encrypt data on disk sectors.
- Volume Master Key (VMK) – 256-bit AES key used to decrypt the FVEK.
- External Key (ExK) or Clear Key (CC) – 256-bit AES key stored outside the cryptographic boundary (for example a USB device). This key is entered into the module via the USB port and is the only method used to decrypt the VMK that results in the VMK being considered encrypted, as other methods rely upon non-approved key derivation methods. Logically, the external key represents either a startup key or a recovery key.
- Intermediate Key (IK) – 256-bit AES key value that is stored encrypted and forms the basis of a key which decrypts the VMK, by combining with another 256-bit AES key via key derivation or XOR.
- Session Key (SK) – 256-bit AES key value that is stored in plaintext on disk and used to decrypt an IK transported over a trusted network to Boot Manager. Boot Manager does the actual decryption of the IK using AES-CCM.
- Network Key (NK) – 256-bit key used for AES decryption of the VMK.

In addition to the keys listed above, when the module is not in FIPS mode, the module is able to use the following keys:

- Derived Key (DK) – 256-bit AES key value used to decrypt the VMK. The value is not stored long-term and is derived using a method defined by the system configuration. When the VMK is encrypted directly with this value, it is considered to be stored in plaintext.

The VMK is always stored in encrypted form; however, based on system configuration the method used to perform the encryption may use a non-approved key derivation method meaning that the VMK is considered plaintext when stored. When encrypted with the ExK/CC (identified above), the VMK is considered to be in an encrypted form, as no key derivation methods are used for the encrypting key. When keys from non-approved password-based key derivation methods are used to protect the VMK, it is considered plaintext. The VMK value can be zeroized by formatting the drive volume on which the key is stored.

Note that the FVEK is stored in encrypted form across power cycles, and thus not subject to the zeroization requirements, but can be zeroized in the same manner as the VMK. The ExK/CC, is stored only in memory and is zeroized by rebooting the OS.

Boot Manager also uses public keys stored on the computer hard disk to verify digital signatures using its implementation of RSA PKCS#1 (v1.5). These public keys are available to both roles. Zeroization is performed by deleting the Boot Manager module.

When authenticating with TPM+PIN or TPM+PIN+SK, the PIN is combined with a salt in a key derivation and then provided to the TPM as authentication data to release an intermediate key (IK). In addition, the PIN is combined with a different salt in a key derivation to create a derived key (DK) which is combined via XOR with the IK (and the ExK on the USB key, if applicable), to form the key that actually decrypts the VMK. The method of key derivation is considered non-approved and the VMK is considered plaintext when encrypted using a method relying on a PIN/password or the TPM alone.

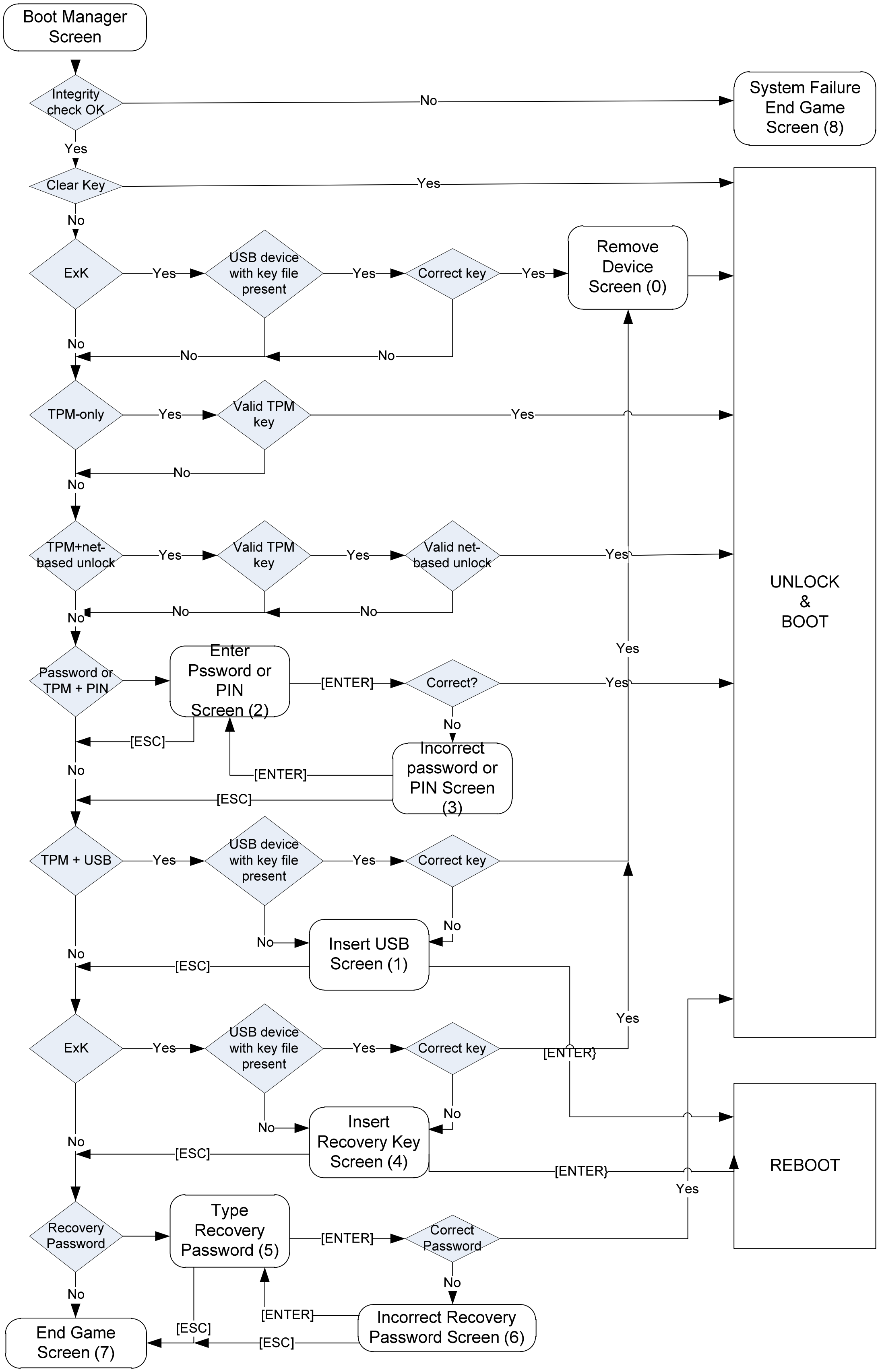
Note that recovery passwords are disabled in FIPS mode, therefore Derived Keys (DK) are disabled in FIPS mode.

Boot Manager will seek appropriate keys to decrypt the encrypted (i.e. “BitLocker® protected”) volume at boot time and waking up from hibernation, in the following sequence:

1. Clear Key
2. No-TPM required and no user input required
    a. ExK (TPM-less startup key scenario)
3. TPM system and no user input required
    a. TPM
    b. TPM+USB
4. UI Required (TPM system or no-TPM system)
    a. TPM+PIN
    b. TPM+PIN+USB
    c. ExK (i.e. USB startup key or USB recovery key)
    d. Recovery password

The following diagram illustrates the flow logic in the system.

Boot Manager



8.1 Cryptographic Keys

The Boot Manager crypto module uses the following cryptographic keys in FIPS mode:

Cryptographic Key	Key Description
External Key (ExK) or Clear Key (CC)	Key used for AES decryption of the VMK.
Intermediate Key (IK)	Key used as the basis of another AES Key, such as the NK.
Session Key (SK)	Key used for AES decryption of the encrypted IK delivered over a trusted network during Network Unlock authentication[2].
Network Key (NK)	Key used for AES decryption of the VMK. Composed by XOR of an IK protected by the TPM and an IK delivered over a trusted network. Used in Network Unlock authentication.
Volume Master Key (VMK)	Key used for AES decryption of the FVEK
Full Volume Encryption Key (FVEK)	Key used for AES encryption/decryption of data on disk sectors
Asymmetric Public keys	Keys used for RSA PKCS#1 (v1.5) verification of digital signatures

The following table lists keys that are not used when the module is in FIPS mode:

Cryptographic Key	Key Description
Derived Key (DK)	Key used for AES decryption of the VMK. Derived Keys are used in Password and Recovery Password[3] authentication.

Details about the keys and network protocol used for Network Unlock authentication are in the Network Key Protector Unlock Protocol Specification [MS-NKPU], which is available at http://msdn.microsoft.com/en-us/library/hh537327(v=prot.10).

8.2 Critical Security Parameters

The Boot Manager crypto module does not have Critical Security Parameters (CSPs).

8.3 Security Relevant Data Items

The Boot Manager crypto module does not have Security Relevant Data Items (SRDI).

[2] Network Unlock authentication concerns BitLocker functionality. It is not referring to FIPS 140-2 standard authentication used to control access to the ports and services of the cryptographic module.

[3] The Recovery Password method is not allowed to be used in FIPS mode.

8.4 Access Control Policy

The Boot Manager crypto module does not allow access to the cryptographic keys contained within it. For this reason, an access control table is not included in this document. Boot Manager receives keys from outside and then manages them appropriately once received. Boot Manager prevents access to its keys by zeroizing them.

9 Authentication

Boot Manager does not implement any authentication services as defined by the FIPS 140-2 standard, which is concerned exclusively with controlling access to cryptographic module ports and services. The User and Cryptographic Officer roles are assumed implicitly by booting the Windows operating system.

10 Self-Tests

10.1 Power-On Self-Tests

Boot Manager performs the following power-on (startup) self-tests.

- SHA (SHA-1/SHA-256) Known Answer Tests
- RSA Known Answer Tests
- Software Integrity Test – RSA PKCS#1 (v1.5) verify with public key
- AES Known Answer Tests
- HMAC-SHA-1 and HMAC-SHA-256 Known Answer Tests
- SHA-512 Known Answer Tests

If the self-test fails, the module will not load, the system will not boot, and status will be returned. If the status is not STATUS_SUCCESS, then that is the indicator a self-test failed.

11 Design Assurance

The secure installation, generation, and startup procedures of this cryptographic module are part of the overall Windows 8, Windows RT, Windows Server 2012, and Windows Storage Server 2012 operating system secure installation, configuration, and startup procedures. After the operating system has been installed, it must be configured by enabling the "System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing" policy setting followed by restarting the system. This procedure is all the crypto officer and user behavior necessary for the secure operation of this cryptographic module.

Windows Phone 8 does not use the same installation, configuration, and startup procedures as the Windows operating system on a computer, but rather, is securely installed and configured by the cellular telephone carrier.

The procedures required for maintaining security while distributing and delivering versions of a cryptographic module to authorized operators are:

1. The secure distribution method is via the physical medium for product installation delivered by Microsoft Corporation, which is a DVD in the case of Windows 8 and Windows Server 2012. In the case of Windows RT, Surface Windows RT, Surface Windows 8 Pro, Windows Phone 8, and Windows Storage Server 2012, the cryptographic module is already installed at the factory and is only distributed with the hardware.
2. An inspection of authenticity of the physical medium can be made by following the guidance at this Microsoft web site: http://www.microsoft.com/en-us/howtotell/default.aspx
3. The installed version of Windows 8, Windows RT, Windows Server 2012, and Windows Storage Server 2012 must be verified 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 by Microsoft certificates. The Windows Update client only accepts content whose 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.

12 Mitigation of Other Attacks

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

Algorithm	Protected Against	Mitigation	Comments
SHA1	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 used with AES NI

13 Additional Details

For the latest information on Microsoft Windows, check out the Microsoft web site at:
http://windows.microsoft.com
For more information about FIPS 140 evaluations of Microsoft products, please see:
http://technet.microsoft.com/en-us/library/cc750357.aspx

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

14.1 How to Verify Windows Versions

The installed version of Windows 8, Windows RT, Windows Server 2012, and Windows Storage Server 2012 must be verified to match the version that was validated using one of the following methods:

1. The ver command
    a. From Start, open the Search charm.
    b. In the search field type "cmd" and press the Enter key.
    c. The command window will open with a "C:\>" prompt.
    d. At the prompt, type "ver" and press the Enter key.
    e. You should see the answer "`Microsoft Windows  [Version 6.2.9200]`".
2. The systeminfo command
    a. From Start, open the Search charm.
    b. In the search field type "cmd" and press the Enter key.
    c. The command window will open with a "C:\>" prompt.
    d. At the prompt, type "systeminfo" and press the Enter key.
    e. Wait for the information to be loaded by the tool.
    f. Near the top of the output, you should see:

```
OS Name:                   Microsoft Windows 8 Enterprise
OS Version:                6.2.9200 N/A Build 9200
OS Manufacturer:           Microsoft Corporation
```

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

14.2 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: x.x.xxxx.xxxxx.
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.