Windows 7 BitLocker™ Drive Encryption Security Policy

For FIPS 140-2 Validation

For Windows 7

Document version 0.7
09/30/2010

1. Table of Contents

2. Introduction

Windows 7 BitLocker™ Drive Encryption is a data protection feature available in Windows® 7 Enterprise and Ultimate for client computers and in Windows Server 2008 R2. BitLocker is Microsoft's response to one of our top customer requests: address the very real threats of data theft or exposure from lost, stolen or inappropriately decommissioned PC hardware with a tightly integrated solution in the Windows Operating System.

BitLocker prevents a thief who boots another operating system or runs a software hacking tool from breaking Windows file and system protections or performing offline viewing of the files stored on the protected drive. This protection is achieved by encrypting the entire Windows volume. With BitLocker all user and system files are encrypted including the swap and hibernation files.

The feature ideally uses a Trusted Platform Module (TPM 1.2) to protect user data and to ensure that a PC running Windows 7 has not been tampered with while the system was offline; however, no TPM modules were included as part of this validation effort. Therefore, no claim is made about the security of any method of encrypting the VMK which uses a TPM. BitLocker provides both mobile and office enterprise information workers with enhanced data protection should their systems be lost or stolen, and more secure data deletion when it comes time to decommission those assets. BitLocker enhances data protection by bringing together two major sub-functions: full drive encryption and the integrity checking of early boot components.

Integrity checking the early boot components helps to ensure that data decryption is performed only if those components appear unmolested and that the encrypted drive is located in the original computer.
BitLocker offers the option to lock the normal boot process until the user supplies a PIN, much like an ATM card PIN, or inserts a USB flash drive that contains keying material. These additional security measures provide multi-factor authentication and assurance that the computer will not boot or resume from hibernation until the correct PIN or USB flash drive are presented.

In Windows 7 and Windows Server 2008 R2, three new features have been added to the product, including:

- Support for new file systems (FAT, FAT32, ExFAT).
- Support for removable data volumes: now any volume formatted using a supported file system can be protected, whether an external hard-drive or a flash stick.
- New key protectors: a password or a smartcard can now be used to protect data volumes.
- New recovery mechanism: a public-key-based key-protector can now be used by enterprise-designated Data Recovery Agents (DRA) to transparently protect all volumes and recover them without the need of a recovery key or recovery password.

2.1 List of Cryptographic Modules

BitLocker includes seven cryptographic modules that use the following cryptographic algorithms:

1. Hashing: SHA-1 (for TPM communications), SHA-256.
2. Keyed hash: HMAC, AES in CCM mode (128 and 256 bit).
3. Symmetric key encryption: AES in CBC mode (128 and 256 bit), with or without the use of Elephant Diffuser algorithm.
4. Asymmetric key encryption: RSA (2048 bit).

The modules performing cryptographic operations are (those in bold are included as part of this validation):

Pre-boot environment:

1) BOOTMGR
2) WINLOAD.EXE
3) **WINRESUME.EXE**

Post boot environment:

4) CI.DLL
5) CNG.SYS
6) **FVEVOL.SYS**
7) **DUMPFVE.SYS**
8) **FVEAPI.DLL**
9) BCRYPTPRIMITIVES.DLL
10) **WIN32_TPM.DLL**

2.2 Brief Module Description

This section briefly describes each module and the technical differences between them:

BOOTMGR
: This is the system boot manager, called by the bootstrapping code that resides in the boot sector. It locates the VMK (Volume Master Key) and the FVEK (Full Volume Encryption Key), it gets the authentication keys required (depending on the authentication scenario) and decrypts a portion of the disk so that the OS can be loaded. It then checks the integrity of the OS loader and launches it.

WINLOAD.EXE
: This is the OS loader. It loads the boot-critical driver image files and the OS kernel image file itself.

WINRESUME.EXE
: This is the filter that handles resuming from hibernation. At resume time, the data is decrypted as it is paged back into memory.

CI.DLL
: This component provides Code Integrity for the OS by cryptographically verifying the integrity of OS components each time they are loaded into memory.

CNG.SYS
: This is the main cryptographic provider for the OS itself.

DUMPFVE.SYS
: This is the BitLocker™ filter that sits in the system dump stack. Whenever the dump stack is called (in the event of a crash, or for hibernation), this filter ensures that all data is encrypted before it gets written to the disk (as a dump file or hibernation file)

FVEVOL.SYS
: This is the BitLocker™ driver. It performs disk conversion (encryption/decryption) and on-demand decryption of disk data.

FVEAPI.DLL
: This is the internal (un-exposed) API that controls the different BitLocker™ functions, in particular key generation and key management.

BCRYPTPRIMITIVES.DLL
: This Windows component provides cryptographic services to callers executing outside of the kernel space.

WIN32_TPM.DLL
: This is the WMI provider for the TPM API. It provides an interface for controlling TPM functionality.

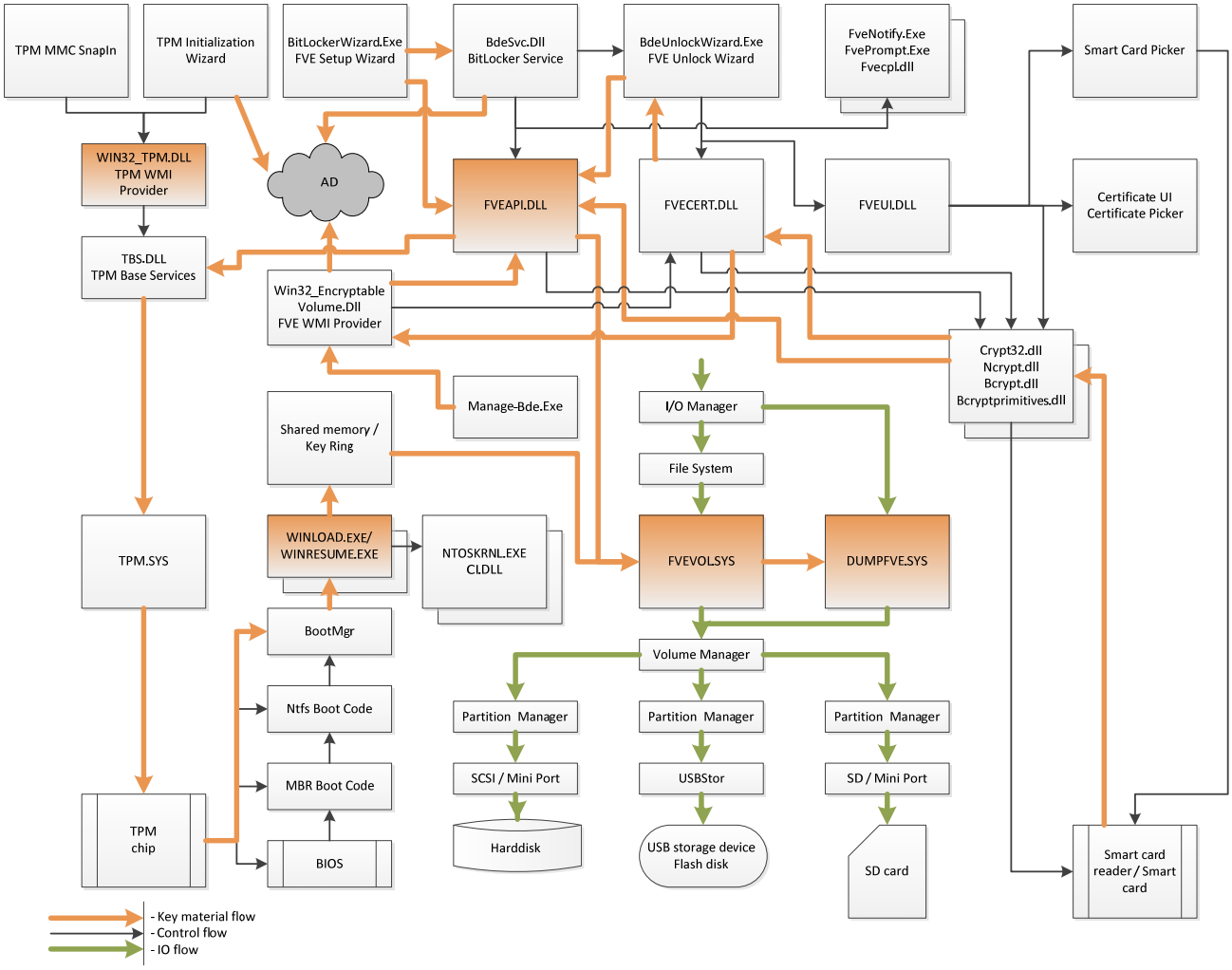


Figure 1 - Logical Operation of Module (orange components in the cryptographic boundary)

2.3 Validated Platforms

The BitLocker™ components (versions: 6.1.7600.16385, 6.1.7600.16429, and 6.1.7600.20536) identified in section 4 have been validated on the Microsoft Windows 7 Ultimate Edition on both x86 and x64. The Microsoft Windows 7 Ultimate Edition is a superset of the Windows 7 Enterprise Edition, which also includes BitLocker™ Drive Encryption. Thus, BitLocker™ maintains FIPS 140-2 compliance on both Windows 7 Enterprise and Ultimate Edition, for both x86 and x64 processor architectures.

3. Integrity Chain of Trust

The cryptographic integrity checking of early boot components in the Windows 7 and BitLocker™ cryptographic modules as follows:

1. BOOTMGR cryptographically checks its own integrity during its start up.
2. BOOTMGR then cryptographically checks the integrity of the OS loader (WINLOAD.EXE or WINRESUME.EXE if resuming from hibernation) before starting it.
3. WINLOAD.EXE cryptographically checks the integrity of CI.DLL before loading it.

4. CI.DLL cryptographically checks the integrity of the post-boot Windows and BitLocker™ cryptographic modules (CNG.SYS, DUMPFVE.SYS, FVEVOL.SYS, FVEAPI.DLL, BCRYPTPRIMITIVES.DLL, and WIN32_TPM.DLL) when the Windows Memory Manager attempts to load such cryptographic module.

4. Cryptographic Boundaries

4.1 Overall Cryptographic Boundary

For FIPS 140-2 purposes the cryptographic boundary is the physically contiguous enclosure of the computer system upon which Microsoft Windows 7 and BitLocker™ Drive Encryption executes (as we define the module to as a multi-chip standalone module). Within the Microsoft Windows 7 Operation System exists a second cryptographic boundary, drawn around those components responsible for providing BitLocker™ Drive Encryption functionality.

4.2 BitLocker™ Components Included in the Boundary

The Windows 7 BitLocker™ Drive Encryption cryptographic boundary includes the WINRESUME.EXE, DUMPFVE.SYS, FVEVOL.SYS, and FVEAPI.DLL components. These components, in addition with the other Windows 7 operating system components described below, provide the cryptography and functionality for full drive encryption and chain of trust integrity checking during the boot process.

4.3 Other Windows 7 Components

In addition to the aforementioned BitLocker™ components, other Windows 7 operating system components provide integral to the operating of BitLocker™ Drive Encryption. The Windows 7 Boot Manager (bootmgr) (Cert. #1319), Windows 7 Winload OS Loader (winload.exe) (Cert. #1326), Windows 7 Code Integrity (ci.dll) (Cert. #1327), Microsoft Windows 7 Kernel Mode Cryptographic Primitives Library (cng.sys) (Cert. #1328) and Microsoft Windows 7 Cryptographic Primitives Library (bcryptprimitives.dll) (Cert. #1329) provide supporting cryptographic services to the BitLocker™ Components as well as cryptographically assure the integrity of the BitLocker™ components (in addition to cryptographically ensuring the integrity of each component in the Windows boot sequence). The BitLocker™ Driver Encryption cryptographic boundary does not include these components as these components have been subjected to separate FIPS 140-2 validations to ensure compliance.

Because the BitLocker™ Drive Encryption components depend upon these other Windows 7 operating system components, the BitLocker™ Drive Encryption validation is said to be bound to the Windows 7 operating system, and requires it to remain compliant.

4.4 Other BitLocker™ Components

Beyond the BitLocker™ Drive Encryption components included in the cryptographic boundary, there exist other BitLocker™ components that are not included in the boundary. The non-cryptographic components of BitLocker™, for example, the BitLocker™ Setup Wizard that provides a friendly graphical user interface, are not suitable for inclusion into the cryptographic boundary as they provide no cryptography.

5. Roles, Services and Authentication

BitLocker™ provides two different, implicitly assumed roles and a set of services particular to each of the roles. As a FIPS 140-2 level 1 validated product, BitLocker™ itself does not provide any authentication;

however, as with all other Windows components, access to BitLocker™ is granted only after the Windows 7 operating system successfully authenticates (through WinLogon) an operator. The Microsoft Windows 7 operating system authenticates an operator's identity by verifying his credentials through WinLogon, at login time, and then implicitly assigns him either the Crypto-Officer or User role depending on the group permissions associated with the operator's ID.

5.1 Roles

BitLocker™ provides both a Crypto-officer (Administrator) and User Role.

5.1.1 User Role

The User Role has access to the unauthenticated services. Once the PC boots, the user will be able to log into the system. This means the User role has access to the Self-Tests and Show Status services. In addition, the user has access to the services discussed below.

For removable data volumes, a user will be able to perform any of the following services:

- Select / Create key protection methods (key protectors)
  - Password, smartcard
- Select / Create recovery key
- Manage keys
  - Reset password
  - Copy recovery key
  - Create / delete an auto-unlock key
- Turn-off BitLocker (volume decryption)

5.1.2 Crypto-officer Role

The Crypto-officer Role has access to the PC's administrative services, including BitLocker administration. The Crypto-officer must initialize BitLocker on a new PC upon receipt, by selecting the encryption and recovery methods to be used and launching the conversion (encryption) process. Once authenticated, the Crypto-officer can perform any of the following services:

- Configure BitLocker into FIPS mode
- Select / Create key protection methods (key protectors)
  - For OS volumes: TPM, TPM+PIN, TPM+USB+PIN, TPM+USB, USB
  - For data volumes (fixed or removable): password, smartcard
- Select / Create recovery key
- Manage keys
  - Copy keys (startup key, recovery key)
  - Reset PIN
- Disable/ Re-enable protection (go into and out of suspend mode)
- Turn-off BitLocker (volume decryption)
- Data volume management
  - Reset password
  - Copy / delete recovery key
  - Create / delete an auto-unlock key

5.2 Startup and Recovery Mechanisms

BitLocker incorporates five different startup methods and two recovery mechanisms (of which a subset is available when one initializes BitLocker™ to operate in FIPS mode):

- TPM-only authentication;
- TPM + PIN authentication;

- TPM + Startup key authentication;
- TPM + PIN + Startup key authentication;
- Startup key only authentication.
- For data volumes (fixed or removable)
    - Password authentication;
    - Public-key-based authentication (e.g. smartcard).

The following recovery mechanisms are available:
- Recovery key authentication;
- Public-key-based (e.g. DRA): A public key distributed to all BitLocker-protected devices as configured by Group Policy.
- Recovery password authentication (not available in FIPS mode).

No TPM modules were included as part of this validation effort. Therefore, no claim is made about the security of any method of encrypting the VMK which uses a TPM.

6. Secure Operation and Security Rules

In order to operate BitLocker™, the operator should be aware of the security rules enforced by the module and should adhere to the physical security rules and secure operation rules required.

6.1 Security Rules

The security rules enforced by BitLocker include both the security rules that Microsoft has imposed and the security rules that result from the security requirements of FIPS 140-2.

6.1.1 Microsoft Security Rules

The following are security rules imposed by Microsoft:

1. BitLocker can only be initialized by a Crypto Officer on OS volumes and fixed, internal data volumes.
2. BitLocker will only allow a Crypto Officer to perform key management operations on OS volumes and fixed, internal data volumes.
3. BitLocker will allow any User that possesses the appropriate authentication credentials to operate a computer that has a volume protected with this technology, and to initialize and manage keys for removable data volumes.

6.1.2 FIPS 140-2 Security Rules

The following are security rules that result from the security requirements of FIPS 140-2:

1. BitLocker™ will not allow creation or use of a recovery password in FIPS mode as FIPS 140-2 prohibits password deriving keys for data encryption/decryption.
    a. A “recovery password” is a 48 digit value that can be used to recover an encrypted volume, in the event that the main authentication keys are lost, stolen or unusable.
2. BitLocker™ will only release keys to be stored on USB flash drives, if the Crypto Officer performs this operation
3. BitLocker™ Drive Encryption is supported on Windows 7 Enterprise and Ultimate Editions (both 32-bit and 64-bit versions).
4. Windows 7 is an operating system supporting a “single-user” mode where there is only one interactive user during a logon session.

5. BitLocker™ Drive Encryption provides a variety of different key management options to allow customers to implement a key storage, authentication, and backup/recovery scheme that meets their needs. When placed into FIPS-mode, BitLocker™ will only offer FIPS approved methods included as part of this validation.
6. BitLocker™ will only operate in its FIPS-mode once volume conversion (encryption) has completed and the volume is fully encrypted.

6.2 Enabling FIPS Mode

In order to allow the local administrator to enable or disable FIPS compliance, BitLocker™ complies with the "System Cryptography: Use FIPS compliant algorithms" policy. Additionally, the local administrator may need to enable the BitLocker to operate without a valid TPM device if the computer lacks a valid TPM device.

The FIPS mode policy (that can be found here: Computer Configuration -> Windows Settings -> Security Settings -> Local Policies -> Security Options -> System Cryptography: Use FIPS compliant algorithms) setting allows you to configure the computer to use only FIPS compliant algorithms.

The policy controlling whether BitLocker will require the presence of a valid TPM device (that can be found here: Computer Configuration -> Administrative Templates -> Windows Components -> BitLocker Drive Encryption -> Operating System Drives -> Required additional authentication at startup) must be set if the computer lacks a TPM or if using the Startup Key only method.

The administrator should select one of the three available options – "Use Bitlocker without Additional Keys", "Require PIN at every startup", or "Require Startup USB key at every startup" – under the "Set BitLocker Startup Preferences" setup wizard screen. If the administrator wishes to setup the TPM + PIN + USB mode, they would first initialize by opening a command prompt and run the command: `Cscript C:\windows\system32\manage-bde.wsf –protectors –add –tpsk`

By enabling these policy settings and specifying one of the available options, BitLocker™ will operate in its FIPS-mode and will only use FIPS compliant algorithms authentication and recovery mechanisms. This means that no recovery password can be created or consumed, and that only a recovery key can be used for recovery purposes. This is also valid for foreign or data volumes: you can recover/unlock such volume only with a recovery key, and not with a recovery password. Therefore, when operating in FIPS-compliant mode, the Setup wizard will disable the buttons allowing the creation, display or printing of a recovery password. The Save to Folder link will also be disabled on the save recovery key page. Additionally, the recovery wizard will only allow recovery of foreign volumes using a recovery key. In FIPS mode, it will grey-out the Disable mode link and the Recovery Password entry option. Only a recovery key can be used, and only to decrypt the volume.

In Windows 7, users can suspend BitLocker (go into disabled mode) even when the FIPS GP is set. Furthermore, in Windows 7 users will be able to consume passwords when unlocking a data volume, provided the volume is mounted in RO mode once unlocked. In such case, if the FIPS flag is on, and a user uses a password to unlock a data volume (fixed or removable), the unlock wizard will allow the use of the password, the volume will be unlocked (can be unlocked with manage-bde/WMI as well) but the volume is mounted in RO mode. Note that when the FIPS flag is on, users will be able to create a data volume password key protector.

Additionally, in order to allow roaming down-level, in Windows 7 BitLocker will allow creation of a "hybrid volume", protected with a password, even when the FIPS flag is on. Because the password key protector has just been added (and therefore the volume was not unlocked with a passphrase (yet)), the user can still write to the volume. Once the volume is unplugged and plugged back in, the volume is mounted in RO mode, as described above, ensuring that if you unlock with the password, users get only RO access.

If the group policy requires a password and FIPS compliant mode is on, a sanity check error is presented. This is a continuation of the policy of not silently failing to provide requested actions. In non-FIPS mode, the default mode was “require password” if no policy registry key was set. In FIPS mode, the default mode is “no password” in order to avoid sanity check errors.

If FIPS policy is on (which prevents creation of a recovery password) the error returned by FVEAPI is: FVE_E_FIPS_PREVENTS_RECOVERY_PASSWORD.

Disallowing both recovery password and recovery key in FIPS mode will allow users to go through the wizard without a policy error.

To recapitulate:

- In FIPS mode, no recovery password can be created or consumed. This is because when consuming a recovery password, BitLocker does not place the volume in RO mode.
- In FIPS mode, only recovery keys can be used to recover OS volumes (recovery keys or DRAs can be used for data volumes).
- In FIPS mode an unlock password (a.k.a. passphrase) can only be created for hybrid volumes (when the GP for hybrid volumes is turned on). Once such a volume is protected it is RW until it is unplugged, at which point, if a password is used to unlock it, it will be mounted RO. Hybrid volumes will always, by design, be unlocked in RO mode on down-level platforms.
- In FIPS mode, an unlock password (a.k.a. passphrase) can be used to unlock a data volume. If a password was used to unlock a data volume, the data volume will be unlocked in RO mode.

Once the administrator has configured the policies as described above, and set up BitLocker™, BitLocker™ will begin encrypting the operating system volume. Once this conversion process is complete, BitLocker™ will be operating in FIPS-mode.

Additionally, it is recommended that domain administrators enable the FIPS policy before turning on BitLocker™. If FIPS mode is enabled after BitLocker™ was turned on, BitLocker™ must be turned off, and turned back on in order to remain compliant with the FIPS 140-2 requirements. This is because FIPS requires that all keys used in FIPS mode should have been created in FIPS mode.

7. Cryptographic Key Management

In order to achieve a higher level of security, without greatly affecting usability, BitLocker™ supports different types of cryptographic algorithms and encryption layers, including multi-factor authentication. Note that only a subset of options is available when operating in FIPS mode.

The main goal of BitLocker™ is to protect user data on the Operating System volume of the hard drive. To achieve this, disk sectors are encrypted with a Full Volume Encryption Key (FVEK), which is always encrypted with the Volume Master Key (VMK), which, in turn, is bound to the TPM (in TPM scenarios).

The VMK directly protects the FVEK and therefore, protecting the VMK becomes critical. Protecting the disk through the VMK allows the system to re-key easily when one of the other keys upstream in the chain is lost or compromised, especially since decrypting and re-encrypting the entire volume is expensive.

There are several different ways to encrypt the VMK:

Scenario	**VMK blob**	**Algorithm used to encrypt VMK**
Default (TPM-only)	SRK(VMK)	RSA
TPM and PIN	(SRK+SHA256(PIN))(VMK)	RSA

TPM and PIN and USB	XOR((SRK+SHA256(PIN)),SK)(VMK)	AES
TPM and USB (TPM+SK)	XOR(SRK(IK),SK)(VMK)	AES
Startup key (SK)	SK(VMK)	AES
Recovery key (RK)	RK(VMK)	AES
Recovery password[1]	(Chained-hashing(Password),Salt)(VMK)	AES
Data volume password[1]	(Chained-hashing(Password),Salt)(VMK)	AES
Public-key-based	IK(VMK) where IK is RSA or ECC-encrypted with the PK	AES
Clear key (CC)	CC(VMK)	AES
Auto-unlock key (AUK)	OS_VMK(IK(VMK)) or user_PK (form user cert store)	AES

No TPM modules were included as part of this validation effort. Therefore, no claim is made about the security of any method of encrypting the VMK which uses a TPM.

The SRK is the Storage Root Key held by the TPM. It is a 2048 bit RSA key pair generated when ownership of the TPM is taken. The SRK, referred to here as an RSA key, is actually the RSA public key; the private key member of the pair is never shown by the TPM. The SRK is stored within the non-volatile protected memory of the TPM and cannot be removed. This helps ensure that the private key material cannot be leaked, and prevents keys from being used on any platform other than the one they were created on. However, mechanisms are available to migrate keys from one TPM to another, for backup and disaster recovery purposes.

All TPM key operations are based on the SRK. When ownership is taken of the TPM, the new owner is required to specify two pieces of authorization information, the ownership authorization and the SRK usage authorization. This SRK usage authorization will be required for each TPM operation. Since this is undesirable from a usability point of view, and since BitLocker requires that this information be known very early in the boot process, the TPM admin tools will set this usage authorization to a known value of all zeroes (20 bytes of 0). The SRK is re-keyed each time the owner changes.

In Windows 7, BitLocker has updated the way in which the VMK is protected when authenticating with TPM+PIN and TPM+PIN+SK. Specifically, the PIN is now not only used as authentication data by the TPM, but, in addition, it is also added in the computation of the encryption key protecting the VMK. This method of key derivation is considered non-approved and the VMK is considered plaintext when encrypted using a method relying on either the TPM or a PIN/password.

Note that recovery passwords are disabled in FIPS mode.

The keys (FVEK, VMK, and optionally, SK, RK, and/or Clear Key) are generated at set-up time, when BitLocker is enabled. They are stored after being generated. The FVEK and VMK are stored locally in three different places on the drive (beginning, middle, end) and cannot be changed. The only way to change these keys for an encrypted volume is to decrypt and re-encrypt the volume. Most of the keys (SK, RK, recovery password) – except the PIN – can be copied after the volume has been encrypted.

Intermediate keys (IK) are keys that are stored encrypted on the drive and become [part of] the basis of another key. IK is an intermediate symmetric key, 256-bit long, randomly generated, stored on disk encrypted with the SRK.

The clear key is a 256-bit AES key that encrypts the VMK; it is created when no other VMK-protecting key is present, in what is called “suspend (or disabled) mode”. In this case, there is no security – it is as though the

[1] Not allowed in FIPS mode

drive is not encrypted, the data is freely available. The clear key is stored as raw data on the disk, together with the VMK.

If a computer does have a [detectable] TPM, BitLocker™ will offer to use the TPM. An external device to hold a key may still be used, but the key on the external device will be a "partial key" (for two-layer authentication) or a recovery key (RK).

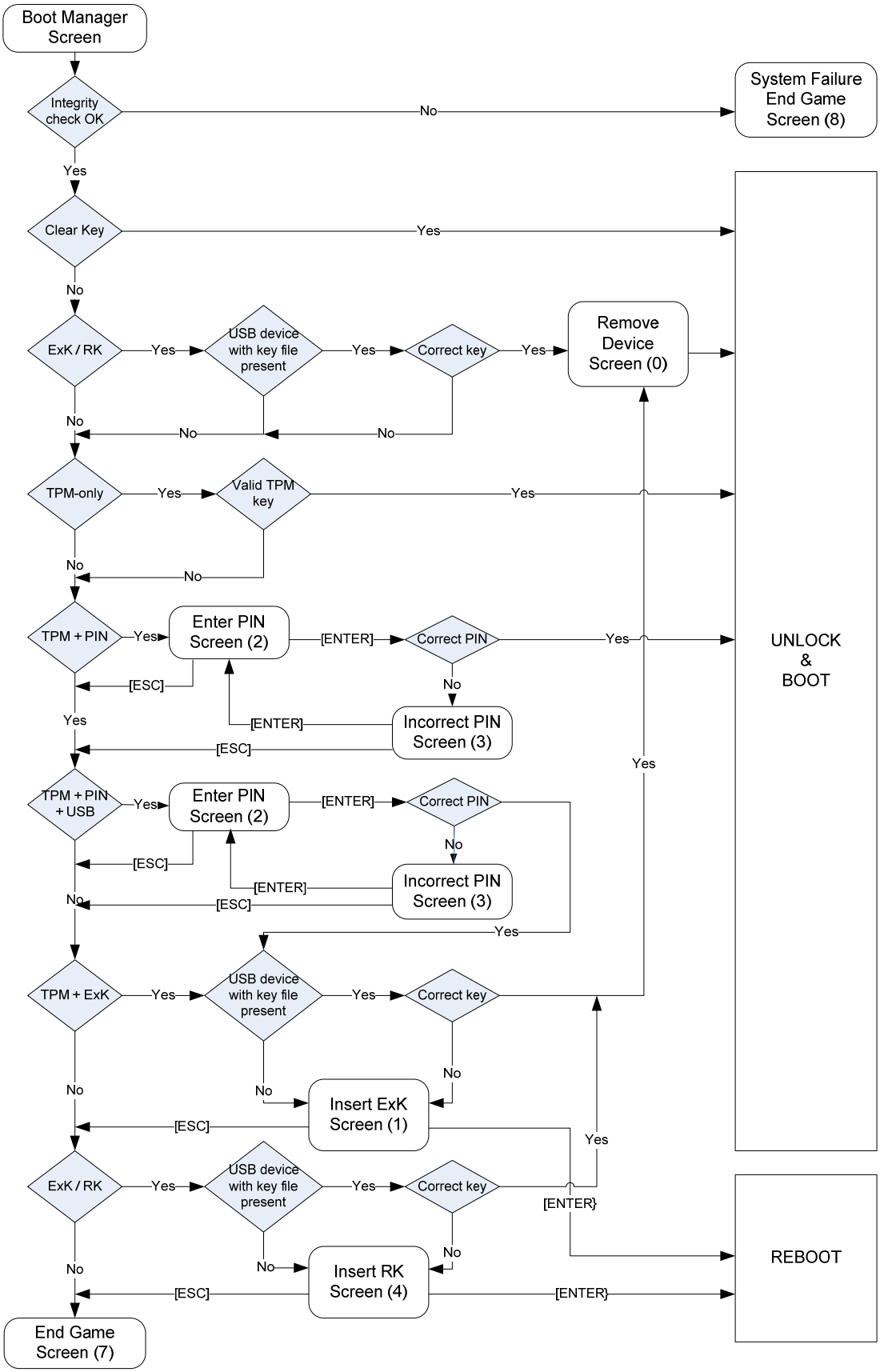
Data volumes can be protected using a password, a public-key-based method (such as a smartcard), external key (recovery key) and/or a recovery password. Data volumes can be "automatically unlocked": when a BitLocker-protected volume is mounted BitLocker automatically unlocks the volume if it finds an auto-unlock key (AUK).

7.1 Flow Logic

BitLocker will look (at boot time and waking up from hibernation) for appropriate keys to decrypt the volume in the following sequence:

1. Clear Key
2. No-TPM required and no user input required
    a. SK (TPM-less scenario)
3. TPM system and no user input required
    a. TPM
    b. TPM+SK
4. UI Required (TPM system or no-TPM system)
    a. TPM+PIN
    b. TPM+PIN+USB
    c. SK/RK
    d. Recovery password

The following diagram illustrates the flow logic in the system.



For data volumes, individual unlocking clients (e.g. the boot environment, the unlock wizard) implement the flow logic themselves. The nominal exception to this is that on volume mount, the driver will attempt to unlock via clear key, and then via driver-based auto-unlock. Assuming neither of those succeed, the driver launches the BitLocker service, which launches the unlock wizard in the user's context. The unlock wizard then attempts DPAPI auto-unlock, and if that fails, shows UI options to the user based on what protectors exist on the volume.

7.2 Key Generation

All keys are generated using FIPS-compliant Random Number Generators (RNGs).

Key	Generated by	Algorithm used	Used by
FVEK	FVE API	BCryptGenRandom	FVE driver
VMK	FVE API	BCryptGenRandom	Boot manager FVE API
Intermediate key	FVE API	BCryptGenRandom	Boot manager FVE API
SK	FVE API	BCryptGenRandom	Boot manager FVE API
RK (similar to SK)	FVE API	BCryptGenRandom	Boot manager FVE API
Clear key	FVE API	BCryptGenRandom	Boot manager FVE API
Public-key-based	FVE API	-	FVE API

7.3 Key and CSP Entry and Output

The raw data encryption keys are all internal and never leave the computer. The only inputs by the user are the PIN or the recovery password (which was generated by the system, but manually entered in recovery mode). The keys that are exported are:

Key	Entry	Output
FVEK	-	-
VMK	-	-
SRK	-	-
Intermediate key	-	-
SK	USB Device	USB Device
RK (similar to SK)	USB Device	USB Device
Auto-Unlock/Clear key	-	-
PIN	Manually via GPC Keyboard	-
Recovery Password	Manually via GPC Keyboard	Active Directory Domain Services
Data volume password	Manually via GPC Keyboard	-
Public-key-based KP	Smartcard device	-

7.4 Key Distribution

As outlined in the previous section, keys reside on the local computer or on a USB device. Only recovery passwords are electronically stored (if so indicated by Group Policy) in an Active Directory Domain Services server. When needed, a user calls helpdesk and a domain administrator can read the AD entry once the user

has been properly identified. At this point the key can be transmitted back to the user either verbally, or any other form approved by the enterprise security policy in place.

Data Recovery Agents (DRA) can also store a private key (on a smartcard or in a certificate) in a safe place, in order to use the DRA key protector, when set by GP to recover access to a volume.

7.5 Key and CSP Zeroization

All keys are zeroized in memory (by overwriting once with 0s) once they are used and no longer needed. Additionally, on shutdown, the FVEK is also zeroized. Furthermore, when turning off BitLocker™, the metadata sections are zeroized by overwriting the disk three times (once with 0s, once with 1s and once with encrypted 0s – which effectively outputs a random pattern).

7.6 Key and CSP Storage

Below is a chart that details where the different types of keys are stored [encrypted]. Keys highlighted below are not directly controlled by or owned by the BitLocker™ module.

Key	Length	Algorithm in which is used	Visible to user	Storage Place (default)	Storage Form
FVEK	128, 256, 512	AES	No	On disk	Encrypted
VMK	256	AES	No	On disk	Encrypted[2]
SRK	2048	RSA	No	TPM	Plaintext
Intermediate key	128, 256	AES	No	On disk	Encrypted
SK	256	AES	Yes	External device	Plaintext
RK (similar to SK)	256	AES	Yes	External device	Plaintext
Clear key	256	AES	No	On disk	Plaintext
PIN	4 to 20 characters	SHA256	Yes	-	-
Recovery password	48 digits	AES	Yes	AD	Plaintext
Data volume password	8 to 100 characters	AES	Yes	-	-
Public-key-based KP	Whatever the smartcard API supports (e.g. 4096)	RSA or ECC	No	Smartcard or certificate	Plaintext

On disk, keys are stored in BitLocker™ metadata sections on the OS volume. The FVE metadata sections are present in three places on the OS volume and stored unencrypted.

- At boot time, only the first copy of the metadata is used. The second and third copies are kept in sync by fvevol.sys – done on first access to encrypted disk.
- Backup copies two and three are used for recovery purposes: using the restore tool
  - Tool will scan the disk for other metadata
- BIOS parameter block (BPB) points to first copy
  - the space used for the pointer was previously used to point to backup Master file table (MFT), which doesn't really move
  - no space for more pointers
- Each copy then points to a subsequent copy

[2] For FIPS purposes, the VMK is considered to be stored in plaintext whenever the TPM only, TPM + Startup Key, TPM + PIN + Startup Key, or TPM + PIN modes are used for protecting this value. The VMK for Data Volumes is always considered plaintext whether using the password or public key protection mechanisms.

8. FIPS Self Checks

8.1 Algorithm implementation conformance tests design

In order to ensure that each of cryptographic algorithms used by BitLocker™ is implemented correctly, each cryptographic algorithm was subjected to CAVP conformance testing. These conformance tests were conducted during the FIPS 140-2 validation process, and the following certificates were issued for the cryptographic algorithms employed by BitLocker™:

Algorithm	Key Size	Mode	Cert #	FIPS approved
AES	128, 256	CBC	1168	Yes
AES	128, 256	CCM	1177	Yes
SHA-1	160	Byte oriented	1081	Yes
SHA-256	256	Byte oriented	1081	Yes
HMAC-SHA-1	KS < BS	N/A	675	Yes
HMAC-SHA-256	KS < BS	N/A	675	Yes

Note that the implementation of ECC and RSA algorithms used for performing smartcard and DRA-related operations exist outside the BitLocker cryptographic boundary. BitLocker™ relies upon the FIPS validated cryptography of CNG and BCRYPTPRIMTIVES. The cryptographic functions used, include:

Algorithm	Key Size	Mode	Cert #	FIPS approved
ECC	256, 384, 512	N/A	141	Yes
RSA	1024, 2048, 4096	N/A	560	Yes

8.2 Power-on self-test (KAT) design

BitLocker™ implements Known Answer Test functions in each cryptographic component for each algorithm employed by that component. These tests run each time a module is powered up (after being powered-off, reset, rebooted, etc.).

8.3 Integrity check design

Windows 7 and BitLocker™ implement integrity checking starting with the boot manager and then continuing to check each subsequent component. The boot manager performs a cryptographic verification of its own integrity before checking the integrity of the next component in the boot process (winload.exe or winresume.exe, if coming from hibernation). Once control is passed to the OS loader, it will check CI.dll, and once the computer boots, Code Integrity (CI.dll) will perform integrity checks on all other modules.

8.4 Continuous RNG checks design

In the FVE API, all RNG is performed using BCryptGenRandom, which uses bcryptprimitives.dll.

8.5 BitLocker bypass tests

When going from a secure state to an insecure state, the system is allowing a bypass of security and a bypass test, consisting of two independent actions, need to be taken.

The fvevol.sys driver has a “raw” mode of operation and a “filtering” mode of operation. When operating in the raw mode, the driver passes through the data as-is without performing any encryption or decryption. When operating in filtering mode, the driver encrypts data it writes and decrypts data it reads.

There are two distinct circumstances under which the fvevol.sys driver performs bypass tests. For the first, the FveInitialDataRead() routine determines a volume's FVE status; this happens before the fvevol.sys driver processes any I/O events. This routine in turn calls the FveInitialDataReadPhase2() routine, which directly calls the self test component after setting the driver state to either raw or filtering mode. The second test occurs when the driver converts the first sector of the disk from encrypted to decrypted or decrypted to encrypted during the conversion process.

The bypass test validates that the internal filter calls to encrypt and to decrypt are transforming the data. Specifically, for an arbitrary M, it verifies that E{M} != M and D{E{M}} = M, where E{M} denotes encrypting message M and D{M} denotes decrypting message M.

For the latest information on Windows 7, check out the Microsoft web site at http://www.microsoft.com.