BitLocker™ Drive Encryption Security Policy
For FIPS 140-2 Validation

© 2008 Microsoft Corporation – This non-proprietary Security Policy may be reproduced only in its original entirety (without revision)
2. Introduction

BitLocker™ Drive Encryption is a data protection feature available in Windows® Vista Enterprise and Ultimate for client computers. 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 Vista 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 can use either a Trusted Platform Module (TPM 1.2) or a USB key to protect user data and to ensure that a PC running Windows Vista has not been tampered with while the system was offline. 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 cryptographic integrity checking of early boot components.

Integrity checking the early boot components helps to ensure that data decryption is performed only if those components are 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.

2.1 List of BitLocker™ and Vista Cryptographic Modules

BitLocker™ Drive Encryption includes several cryptographic modules that operate in conjunction with the cryptographic modules of the Vista operating system. The BitLocker™ modules 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 the Elephant Diffuser algorithm

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

Pre-boot environment
1) BOOTMGR
2) WINLOAD.EXE
3) WINRESUME.EXE

Post boot environment
4) CI.DLL
5) KSECDD.SYS
6) FVEVOL.SYS
7) DUMPFVE.SYS
8) FVEAPI.DLL
9) BCRYPT.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.

**KSECDD.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.

**BCRYPT.DLL**
This Vista component provides cryptographic services to callers executing outside of the kernel space.

**WIN32_TPM.DLL**
This is the Windows Management Instrumentation (WMI) provider for the TPM API. It provides an interface for controlling TPM functionality.

2.3 Validated Platforms

The BitLocker™ components identified in section 4 have been validated on the Microsoft Vista Ultimate Edition, both x86 and x64. The Microsoft Vista Ultimate Edition is a superset of the Vista Enterprise Edition, which also includes BitLocker™ Drive Encryption. Thus, BitLocker™ maintains FIPS 140-2 compliance on both Vista 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 Vista 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 Vista and BitLocker™ cryptographic modules (KSECDD.SYS, DUMPFVE.SYS, FVEVOL.SYS, FVEAPI.DLL, BCRYPT.DLL, and
WIN32_TPM.DLL) when the Windows Vista Memory Manager attempts to load such cryptographic module.

4. **Cryptographic Boundary**

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 Vista and BitLocker™ Drive Encryption executes (as we define the module to as a multi-chip standalone module). Within the Microsoft Windows Vista 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 Vista BitLocker™ Drive Encryption cryptographic boundary includes the WINRESUME.EXE, DUMPFVE.SYS, FVEVOL.SYS, and FVEAPI.DLL components. These components, in addition with the other Vista 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 Vista Components**

In addition to the aforementioned BitLocker™ components, other Windows Vista operating system components provide integral to the operating of BitLocker™ Drive Encryption. The Windows Vista BOOTMGR (Cert. #888), WINLOAD.EXE (Cert. #889), KSECDD.SYS (Cert. #891), CI.DLL (Cert. #890), and BCRYPT.DLL (Cert. #892) 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 Vista 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 Vista operating system components, the BitLocker™ Drive Encryption validation is said to be bound to the Vista 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 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 Vista operating system successfully authenticates (through WinLogon) an operator. The Microsoft Vista 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.

5.1.2 User Role
The User Role has no access to any BitLocker configuration or control services; rather the User Role benefits from the seamless encryption and decryption of the user’s files written to and read from a BitLocker™ encrypted volume.

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

- Configuring BitLocker™ into FIPS-mode (see section 6.2 below)
- Start-up the BitLocker™ Setup Wizard
- Select / Create authentication methods (key protectors)
  - TPM, TPM+PIN, TPM+USB, USB
- Select / Create recovery key
- Manage keys
  - Copy keys (startup key, recovery key)
  - Reset PIN
- Disable/ Re-enable protection (go into and out of disabled mode) ¹
- Turn-off BitLocker™ (volume decryption)

5.2 Startup and Recovery Mechanisms
BitLocker™ incorporates four different startup methods and two recovery mechanisms (of which a subset is available when initializing BitLocker™ to operate in FIPS-mode):

- TPM-only authentication;
- TPM and PIN authentication;
- TPM + Startup key authentication;
- Startup key only authentication

The following recovery mechanisms are available:
- Recovery key authentication
- Recovery password authentication ¹

¹ Not available in FIPS mode.

© 2008 Microsoft Corporation

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 BitLocker™ Security Rules

The following are security rules imposed by BitLocker™:

1. BitLocker™ can only be initialized by a Crypto Officer.
2. BitLocker™ will only allow a Crypto Officer to perform key management operations
3. BitLocker™ will allow any User that possesses the appropriate authentication credentials to operate a computer that has a volume protected with this technology.

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 using 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 currently supported on Windows Vista Enterprise and Ultimate Editions (both 32-bit and 64-bit versions).
4. Windows Vista 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 registry key used for the FIPS-mode policy is:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\FIPSAlgorithmPolicy\Enabled

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.

© 2008 Microsoft Corporation
The registry key used for the TPM policy is:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\RequireValidTPM\Disabled

The policy controlling whether BitLocker will require the presence of a valid TPM device (that can be found here: Computer Configuration -> Local Computer Policy -> Administrative Templates -> Windows Components -> BitLocker Driver Encryption -> Control Panel Setup: Enable Advanced Startup Options) must be set if the computer lacks a TPM or if using the Startup Key only method.

Finally, 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.

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.

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 a time-expensive operation.

There are several different ways to encrypt the VMK:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>VMK blob</th>
<th>VMK encrypted</th>
<th>Available in</th>
</tr>
</thead>
</table>

© 2008 Microsoft Corporation
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.

Note that recovery passwords are disabled when operating in FIPS mode.

The diagram below shows the BitLocker™ Key Architecture.

---

2 In schemes using TPM, the VMK is considered to be in plaintext since no guarantee of cryptographic strength of the TPM implementation is provided by this module.
The keys 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 “disabled mode”. In this case, there is no security – it is as though the 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).

### 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
   - SK (TPM-less scenario)
3. TPM system and no user input required
4. User Interface (UI) Required (TPM system or no-TPM system)
   a. TPM+PIN
   b. SK/RK
   c. Recovery password

The following diagram illustrates the flow logic in the system.
7.2 Key Generation
All keys are generated using FIPS-compliant Random Number Generators (RNGs). The only exception is the nonce used for sending commands to the TPM, which requires nonces to be generated – in this case by the TPM’s RNG (random number generator).

<table>
<thead>
<tr>
<th>Key</th>
<th>Generated by</th>
<th>Algorithm used</th>
<th>Used by</th>
<th>FIPS Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVEK</td>
<td>FVE API</td>
<td>BCryptGenRandom</td>
<td>FVE driver</td>
<td>Yes</td>
</tr>
<tr>
<td>VMK</td>
<td>FVE API</td>
<td>BCryptGenRandom</td>
<td>Boot manager</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FVE API</td>
<td></td>
</tr>
<tr>
<td>SRK</td>
<td>TPM vendor</td>
<td>-</td>
<td>TPM</td>
<td>Yes</td>
</tr>
<tr>
<td>Intermediate key</td>
<td>FVE API</td>
<td>BCryptGenRandom</td>
<td>Boot manager</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FVE API</td>
<td></td>
</tr>
<tr>
<td>SK</td>
<td>FVE API</td>
<td>BCryptGenRandom</td>
<td>Boot manager</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FVE API</td>
<td></td>
</tr>
<tr>
<td>RK (similar to SK)</td>
<td>FVE API</td>
<td>BCryptGenRandom</td>
<td>Boot manager</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FVE API</td>
<td></td>
</tr>
<tr>
<td>Clear key</td>
<td>FVE API</td>
<td>BCryptGenRandom</td>
<td>Boot manager</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FVE API</td>
<td></td>
</tr>
<tr>
<td>PIN</td>
<td>User</td>
<td>-</td>
<td>Boot manager</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FVE API</td>
<td></td>
</tr>
<tr>
<td>TPM Nonce</td>
<td>TPM RNG</td>
<td>TPM vendor specific</td>
<td>Boot manager</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TPM WMI provider</td>
<td></td>
</tr>
</tbody>
</table>

7.3 Key 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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Entry</th>
<th>Output</th>
<th>FIPS mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVEK</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>VMK</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>SRK</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Intermediate key</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>SK</td>
<td>USB device</td>
<td>USB device</td>
<td>Yes</td>
</tr>
<tr>
<td>RK (similar to SK)</td>
<td>USB device</td>
<td>USB device</td>
<td>Yes</td>
</tr>
<tr>
<td>Clear key</td>
<td>-</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>PIN</td>
<td>manually</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Recovery Password</td>
<td>manually</td>
<td>Text file or Active Directory Domain Services</td>
<td>No</td>
</tr>
</tbody>
</table>

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.

### 7.5 Key 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 Storage

Below is a chart that details where the different types of keys are stored.

<table>
<thead>
<tr>
<th>Key</th>
<th>Length</th>
<th>Algorithm in which is used</th>
<th>Visible to user</th>
<th>Storage Place (default)</th>
<th>Storage Form</th>
<th>FIPS Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVEK</td>
<td>128, 256</td>
<td>AES</td>
<td>No</td>
<td>On disk</td>
<td>Encrypted</td>
<td>Yes</td>
</tr>
<tr>
<td>VMK</td>
<td>256</td>
<td>AES</td>
<td>No</td>
<td>On disk</td>
<td>Encrypted</td>
<td>Yes</td>
</tr>
<tr>
<td>SRK</td>
<td>2048</td>
<td>RSA</td>
<td>No</td>
<td>TPM</td>
<td>Plaintext</td>
<td>Yes</td>
</tr>
<tr>
<td>Intermediate key</td>
<td>128, 256</td>
<td>AES</td>
<td>No</td>
<td>On disk</td>
<td>Encrypted</td>
<td>Yes</td>
</tr>
<tr>
<td>SK</td>
<td>256</td>
<td>AES</td>
<td>Yes</td>
<td>External device</td>
<td>Plaintext</td>
<td>Yes</td>
</tr>
<tr>
<td>RK (similar to SK)</td>
<td>256</td>
<td>AES</td>
<td>Yes</td>
<td>External device</td>
<td>Plaintext</td>
<td>Yes</td>
</tr>
<tr>
<td>Clear key</td>
<td>256</td>
<td>AES</td>
<td>No</td>
<td>On disk</td>
<td>Plaintext</td>
<td>Yes</td>
</tr>
<tr>
<td>PIN</td>
<td>4 to 20 digits</td>
<td>SHA256</td>
<td>Yes</td>
<td>AD</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>48 digits</td>
<td>AES</td>
<td>Yes</td>
<td>AD</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

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

### 7.7 Other Key-related Details

**Volume Move:**

After being moved to a different computer, if the volume is moved back to the original TPM it may reuse its original keys, provided they were not over-written. Each BitLocker™ instance will have a single set of keys – if the keys are stored and have not been removed from the original computer the volume should work without recovery when moved back to the original computer.

---

3 For FIPS purposes, the VMK is considered to be stored in plaintext whenever the TPM only, TPM + Startup Key, or TPM + PIN modes are used for protecting this value.
**External Keys**
External Keys are keys saved externally, on a USB flash drive. Startup Keys or Recovery Keys are examples of an External Key. External keys are saved as a .bek file, containing raw binary data of the key in clear.

At set-up time, the SK or RK can only be saved on removable drives. The file is stored as a system file and hidden to prevent inadvertent deletion. RKs can also be copied after creation, ensuring additional back-ups are available. The RK will only be required at boot time, so it is recommended that users remove it after that for security’s sake.

**PINs**
The user will be able to change PIN, but it will not be saved/escrowed programmatically. The RK or recovery password must be used if the PIN is lost. When the PIN scenario is enabled and the new encrypted VMK blob is stored, the system shall remove the encrypted VMK blob that was encrypted with only the TPM, if such blob is present – except for RK/RK or Recovery Password. In this manner, at next boot, the system will require the two-factor authentication, instead of working with only a TPM.

Multiple PINs on a given computer are not supported in this version (the UI does not handle this, and WMI use is discouraged).

**Recovery**
Recovery is called during boot if BitLocker™ authentication fails (for data or non-OS volumes, when the BitLocker™-protected volume fails to be mounted).

A recovery key (RK) stored on a USB flash drive can be used in the recovery sequence to unlock the volume.

The recovery password consists of a 48 character numeric string that is generated and displayed by the system and must be manually entered by the user to recover the volume. After being generated at set-up time, the recovery password can be viewed, saved locally or printed by the user and is sent to the AD server, if required by Group Policy. If AD is not available, and AD password saving is required, then the BitLocker™ set-up is halted.

Zero or more recovery passwords are supported. The maximum number of all key protectors is limited by the maximum size of the metadata structure that can contain them, which is 16K. So for practical purposes, the limit is about 10 recovery password blobs.

If GP requires the recovery password to be sent to AD, then a recovery password is automatically generated, regardless of the Recovery Password UI Policy. If AD is not available, the set-up process is stopped. If AD is not required and the user sees the Recovery Password Creation screen, she will need to click on one of the links (see the password, save it, print it) to actually generate the password. Just clicking Next (without clicking on any of the links) will not generate a password.

BitLocker™ supports the Windows FIPS Group Policy. This GP is off by default, but can be enabled by a local administrator or, through scripting, by a domain administrator. When set, recovery passwords cannot be created or consumed. The FVE API ensures that calls (from either the UI wizard or WMI scripts) to create or consume recovery password-based key protectors will fail.

### 7.8 Administration Aspects
The Domain Administrator has full control over the enabling or disabling of BitLocker™ and most of its components. The administrator will be able to specify through Group Policy (GP):

---

4 A per-volume GUID is also generated and sent up with the recovery password.

© 2008 Microsoft Corporation
a. Create recovery password: Must, Cannot, Optional (default)
b. Store recovery password in Active Directory Domain Services: Yes, No (default)
   If AD is not available and it is required BitLocker™ Setup up will be stopped.
c. Create recovery key: Must (default), Cannot, Optional
   Default media path
d. Store recovery key to a folder/network location: Must, Cannot, Optional (default)
e. Create external key: Must, Cannot, Optional (default)
f. Create PIN: Must, Cannot, Optional (default)
g. Display warning when no recovery mechanisms have been set
h. System Cryptography: Use FIPS compliant algorithms: Enabled, Disabled (default)

If the registry keys controlling these settings have not been set-up by GP, a default will be used. The default is to make choices optional, leaving them to the user.

8. **FIPS Self-tests**

8.1 **Algorithm implementation conformance testing**
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™:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Key Size</th>
<th>Mode</th>
<th>Cert #</th>
<th>FIPS approved</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>128, 256</td>
<td>CBC</td>
<td>715</td>
<td>Yes</td>
</tr>
<tr>
<td>AES</td>
<td>128, 256</td>
<td>CCM</td>
<td>715</td>
<td>Yes</td>
</tr>
<tr>
<td>SHA-1</td>
<td>160</td>
<td>Byte oriented</td>
<td>737</td>
<td>Yes</td>
</tr>
<tr>
<td>SHA-256</td>
<td>256</td>
<td>Byte oriented</td>
<td>737</td>
<td>Yes</td>
</tr>
<tr>
<td>HMAC-SHA-1</td>
<td>KS &lt; BS</td>
<td>N/A</td>
<td>386</td>
<td>Yes</td>
</tr>
<tr>
<td>HMAC-SHA-256</td>
<td>KS &lt; BS</td>
<td>N/A</td>
<td>386</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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**
Vista 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.

8.5 **Continuous RNG checks design**
In the FVE API, all RNG is performed using BCryptGenRandom (in bcrypt.dll).
For the latest information on Windows Vista, check out the Microsoft web site at http://www.microsoft.com.