The Dell Data Protection | Encryption (DDPE) provides Dell customers with a solution for protecting sensitive data while at rest. Protecting data “at rest” means the data is protected while residing on storage media attached to or residing in the system. The storage media can be any platform-supported media and is not restricted to the system boot volume. In addition to the system boot volume, examples of storage media for which protection is available include USB, hot-pluggable 1394, eSATA™ storage and optical drives. Protection is provided by encrypting the data.

There are a number of encryption solutions in today’s market. Each of them attempts to address the customer problem: self-encrypting hard disk drive (HDD) technology as provided by Seagate® Secure and new Trusted Computing Group™ Opal-compliant encrypted storage technology, which only protect data when stored on the system volume, and encrypted USB devices such as IronKey only protect data when stored on the encrypting USB device.

Software solutions bridge some of the gap in recognizing and accommodating different storage technologies, however, software solutions carry an overhead performance penalty, are not easy to use, often do not provide configurability and are not seamless. In addition, software encryption, by design, must expose encryption keys and un-encrypted as well as encrypted data in system memory, thus becoming a potential security target.

The Dell Data Protection | Encryption Full Volume Encryption (DDPE FVE), a new feature of our encryption solution portfolio helps tackle those issues and consists of five basic parts:

- A hardware component called an Hardware Encryption Accelerator.
- Encryption accelerator device driver – a WDM device driver interface to the encryption accelerator hardware.
- A software library layer that handles key management, key blob generation, key binding and provisioning and TPM functionality.
- Application software that provides the user interface, management and implementation software. The application consists of a local (1 to 1) and a remote (1 to many) management and configuration console for policy definition, encryption implementation and policy enforcement engines.
- BIOS support – supports boot process for initializing the encryption accelerator and TPM prior to booting the system and using the encryption accelerator’s capabilities during the boot process.

DDPE FVE implements the best features of each technology and avoids the weaknesses. In response to the performance penalty associated with a software solution, the Dell solution implements a hardware offload encryption accelerator that is capable of streaming data in excess of 3.0Gbs. The Hardware Encryption Accelerator provides encryption and hashing flexibility by offering multiple algorithms, strengths and modes. It does not care what the encryption key is or derived from as long as it conforms to the requirements of the algorithm. The accelerator is media agnostic: it does not care where the data comes from or where the data is going. Provide an encryption key, data and a specific algorithm and the accelerator will crypt the data. The accelerator is U.S. Federal Information Processing Standards (FIPS) 140-2 Level 3 Appendix C compliant. FIPS requires protecting secrets against attack, binding and ownership, encryption key protection and authentication within the hardware module.
The accelerator module implements the Suite B algorithms consisting of AES 128 and 256, Elliptic Curve Digital Signature Algorithm, SHA-256 and SHA-384. In addition to the Suite B complement, the accelerator also implements AES 192, RSA-2048, SHA-1, SHA-512, and 3DES. Not all of the accelerator’s capabilities are utilized in the FVE solution, but by using additional software, these features are available on an as needed basis.

The supporting software for DDP|E FVE is implemented in a traditional three layer approach. The application layer provides the user interface for configuration, management and information. Application components are also responsible for configuration of Data Leakage Protection (DLP) policies. Default templates and Wizards assume the bulk of the encryption complexity and enable ease of use aspects of encryption technology. Drill-down capabilities are available to users that require specific functionality.

The software library abstracts the mechanics of implementing key handling and binding functionality within the application. It also performs key management and local storage, TPM setup, binding and ownership and key load session management for accelerator tasks.

Additional functional capabilities within the library assist with user migration and recovery, back-up and restoration. These functions enable application software to easily perform these vital functions without being implementation-aware. As part of the initial setup of DDP|E FVE, the client-based policy enforcement engine and library create the necessary credential blobs to be migrated/moved or initiates user credentials in migration, restore or recovery sessions.

The Hardware Encryption Accelerator device driver is a kernel-mode driver written in compliance with Microsoft’s WDM specification. It is architected in such a manner as to minimize kernel mode/user mode transitions when performing requested tasks.

The device driver services two interfaces: a kernel-mode interface that is utilized by kernel-mode software used for full volume encryption and a user-mode interface used for on demand application and DLP software. Kernel requests are given priority in order to prevent stalls. Task requests are run to completion at which time the next request is initiated. If a kernel-mode and user-mode request both exist, the kernel request is executed.

BIOS provides boot services in support of FVE, local storage for the system volume encryption blob and power-on, pre-operating system authentication.

The DDP|E FVE Process

The DDP|E FVE process begins with the decision at the management console to encrypt the system storage drive. The policy is sent to the client for enforcement.

In order to protect the encryption key, key blobs are created that bind the key to specific client elements.

In the initial state, the TPM and Hardware Encryption Accelerator are un-initialized and un-owned. The process begins by initializing and taking ownership of these two vital components. The components can be manually set up or default values will be initiated for ease of use and minimum direct touch.
After initialization of the system components, the crypt key is bound to the encryption accelerator hardware. The resulting blob is then bound to the platform TPM, producing a double-wrapped key blob.

The FVE encryption process encrypts all sectors on the system drive or all sectors except forensic data sectors. The management console enables opt-in to support these forensic data sectors. Therefore, the boot process must be supported within system BIOS. Once the key blob has been created, it is stored in BIOS environment space through an interface designed to support provisioning and releasing the blob within BIOS. This is one-way as BIOS is incapable of releasing a key blob to an external request of any kind. Key blobs can only be deleted or over-written. The only time the key resides in system memory in any form is during this creation process.

The next step in the process is to establish user authentication for the boot process and Windows. Authentication support depends on system capabilities, but must originate in the BIOS process. Successful BIOS user authentication releases the key blob to the platform authentication process using the TPM as the platform root of trust. Upon successful TPM authentication of the key blob, it is then instantiated into the encryption accelerator using its FIPS-approved authentication process. If any one of these authentication processes fail, the FVE crypt key is not loaded into the encryption accelerator hardware.

All that remains prior to encrypting the system volume is the backup of the key blob and components necessary to support recovery and migration. A backup/recovery package is created containing the necessary components and this package must be stored off of the platform storage. The recovery package can be stored on a USB storage device, network attached storage or other removable media, but it cannot be placed on the system’s resident storage. In the IT-managed case, the package is stored in the remote console’s key management capability. The system is now ready to begin encrypting the disk.

When all basic requirements have been fulfilled, a specific set of information is created on the system hard drive to inform the encryption system components that FVE is enabled and where FVE is within the encryption process. The process can begin immediately or be postponed until a specific time. The encryption process can be suspended if system performance becomes an issue. If suspended, encryption is only done during system idle time.

During the initial encryption process, the user is free to power off the platform, enter standby or hibernate the system. The process log keeps track of where the encryption process is, and will resume activity as necessary when powered on. When notified of an impending power cycle, the encryption process is immediately suspended, the log is updated and full attention is given to supporting a rapid power transition.

This process, by design, requires very little user or management interaction and happens in the background.
The FVE Cold Boot Process

When BIOS is ready to begin the boot process, BIOS verifies that FVE is enabled and the user has been successfully authenticated. The key blob is retrieved by BIOS-controlled storage and is verified by the TPM and Hardware Encryption Accelerator and, if successful, begins the boot process. Once the key is loaded within the accelerator, a binding session is established between the FVE BIOS support code and the accelerator and the accelerator configuration supporting FVE is locked down. It is only unlocked during a power transition, thereby disabling denial of service attacks by changing the accelerator’s configuration or stealing the session.

The boot process continues with BIOS examining the available storage devices. When the boot device is determined, the FVE encryption process log is loaded into BIOS. This enables the BIOS FVE support code to know when and when not to decrypt or encrypt blocks as they are read or written to the boot storage device.

At the transition from the BIOS to the operating system boot process, an operating system kernel component acquires the session handle from BIOS. The support code is written to allow this hand-off only one time during a power-on session. The appropriate operating system components and BIOS components share the responsibility of saving the encryption session identifier information securely to support resume from standby successfully and seamlessly without user intervention. Resuming from hibernation occurs in a fashion similar to the boot process, although some of the component parts change as required during the creation of the hibernation file and resuming from a hibernated system is handled outside the normal Windows file management system.

Migration and Recovery

In the event that the platform system board fails, the Encryption Accelerator fails or the user is moved to a new platform and retains the FVE encrypted drive, tools executing within a WinPE environment are available to move the encrypted system drive to a new environment. These tools, with the backup key blob, will restore the environment to a state necessary to support the encrypted drive without having to perform a decrypt/re-provision/re-encrypt process. In order to tear down the system, it is necessary to clear ownership of the TPM, the Encryption Accelerator and power-on authentication. Clearing ownership of the Encryption Accelerator is all that is necessary to clear the active FVE condition. However, there are a finite number of times Encryption Accelerator ownership can be cleared before the Encryption Accelerator is un-usable. It is not necessary to clear Encryption Accelerator ownership to accomplish the goal of disabling the FVE environment.