Apple Technical White Paper

Best Practices for Deploying FileVault 2

Deploying OS X Full Disk Encryption Technology

August 2012 – OS X 10.7.4

Contents

Overview

Breaches in data security can appreciably cost a company customers, reputation, and significant fines or damages. When extra protection is needed for valuables, they're secured in a vault. The same rule applies to sensitive computer files—except instead of a physical vault, they're secured in a file vault.

When users store sensitive information on their computers, their corporate future could be at risk. For example, if an employee carries confidential financial data on a portable computer, losing it could allow someone to exploit that data, significantly damaging the organization. Using strong encryption can help protect those files and keep information safe.

If users have special privileges, or if they aren't restricted by policy to store data only in protected areas, they risk inadvertent data exposure. Encrypting the entire disk ensures that all information is protected without potential for accidental exposure. With FileVault enabled, an organization's information is safe on the hard drive.

Second-generation FileVault, FileVault 2, is full disk encryption (FDE) for data-at-rest protection built into OS X for any volume. FileVault 2 keeps all files on the hard drive secure—even when a computer is lost or stolen—with XTS-AES-128 data encryption at the disk level. With Apple's hardware and software integration, organizations can protect all user data in internal and external disk drives with technology that's transparent to the user, yet provides corporate IT access as needed.

Corporate IT departments have many options to protect sensitive data, but knowledge of option tradeoffs ensures that the best approach is deployed. One such tradeoff to consider is that FileVault 2 is restricted to password-based authentication.

Building on the strengths of FileVault 2 in OS X, an organization can ensure that sensitive data is protected from unauthorized access with malicious intent. Administrators have the foundational elements to deploy and use FileVault 2 in creative ways to meet even more complex data protection scenarios not discussed in this paper.

Gain Protection. Retain Simplicity.

Built-in Protection

FileVault 2 is the manageable encryption technology built into OS X that protects an entire volume. FileVault 2 provides full volume encryption that's easy for anyone to use with or without dedicated IT staff.

With FileVault 2, users don't need to worry about individually encrypting each file or storing files in specific encrypted containers—all the data on the entire volume is encrypted. This behind-the-scenes protection allows users to focus on their primary tasks and typical workflow without stopping to think about data protection.

What's in This Paper

To fully appreciate the power and simplicity of FileVault 2, it's important to understand its architecture, the implications of various configurations, and how to best manage these encrypted volumes to meet specific user and support needs.

Understanding the foundation of FileVault 2 and use of a FileVault Master Identity (FVMI) to enable IT departments to manipulate the protected storage is key for any large organization. Proper provisioning and use of the FVMI is central to any effective enterprise deployment.

It's equally important to cover several supporting technologies at a high level. The details of CoreStorage, manipulating keychains, and cryptographic algorithms are examples of such complexities mentioned only for clarity, but which aren't covered in depth in this paper.

Design Approach

FileVault 2 was designed by taking several key elements into account: the need to not only provide a reliable and safe architecture, but also to accommodate varying scenarios—from personal use to managed corporate access.

Usability

Ease of use for end users is central to FileVault 2. End users must be able to protect their data without being a security expert. Having a built-in safety net for personal recovery (such as storing a protected recovery key on Apple's servers) is also crucial to end users. This protection must be possible from anywhere without added complexity.

One traditional obstacle to usability with FDE is its historical reliance on the end user authenticating twice: once for "pre-boot authentication" to unlock the encrypted boot volume, and the second time to log in to the directory service account. If credentials get out of sync for any reason, the user experience is poor. Ensuring that users have safe access without repetitive authentication challenges and syncing password changes enhances usability. Users must be able to reset their passwords at any time without the need to re-encrypt the volume.

IT Access

Administrators fear the risk of losing control of data. It's important they have a mechanism to enable authorized access to end-user volumes in situations as simple as a forgotten password to more complex situations involving forensic access to volumes. Adding support for an institutional recovery method using a FVMI extends the capabilities beyond an individual's self-service recovery method with minimal, but specific, requirements for access and modifications by authorized administrators. Best practices in deploying the FVMI within a large organization depend on end-user and IT support capabilities and restrictions.

Performance

With FileVault 2, users benefit from optimizations in architectural design, cryptographic algorithms, and use of hardware acceleration when available. The algorithm used for block encryption is XTS-AES-128, which has been optimized for 512-byte blocks. The conversion from plaintext to ciphertext and back is performed on the fly with negligible impact to the user experience since it relinquishes processing cycles to user activity. Software optimizations are further accelerated on systems with Intel Core i5 and i7 processors with hardware acceleration using the built-in AES-NI instruction set.

Best Practices for Deployment

With a wide variety of enterprise policies and approaches to protecting data, choosing a single deployment scenario for all users may not be the best approach. To help determine the best way to deploy FileVault 2, it's important to look at usage environments and how IT departments can support end users and recover access to the encrypted volume when needed.

The following deployment strategies brings its own pros and cons affecting setup, day-to-day usage, and the biggest challenge—recovery when needed.

Most large organizations decide to mix and match these practices to fit various categories of users' needs rather than standardizing on one—forcing a single approach across the organization. It's important to understand the usage scenarios of the organization's user base, along with the services, capabilities, and restrictions of the IT staff.

Deployment

Three deployment methodologies for consideration include:

Self-Service—One-to-one deployment with end users taking full control of their own system setup, use, and personal recovery.

- **Pros**: Very small draw on IT resources; full offline recovery is possible
- **Cons**: End-user responsibility and understanding; challenges with corporate key escrow; additional resources needed for compliance enforcement and monitoring

Cafe—Similar to Self-Service, but IT assists end users with a menu of options and services for initial setup, daily use, and volume recovery.

- **Pros**: IT oversight; expert guidance; use of proven tools when needed
- **Cons**: Additional resources needed for compliance enforcement and monitoring

Centralized—IT staff taking strict control over user system setup and management while ensuring all data is protected and recoverable at any time.

- **Pros**: IT assurance of compliance and key escrow
- **Cons**: Heaviest draw on IT resources for deployment and recovery planning

Self-Service Methodology

One-to-One Deployment

The Self-Service methodology is a one-to-one deployment that focuses on the use of FileVault 2 in an environment where end users play the major role in controlling their system setup, use, and recovery. This doesn't refer only to systems that may function alone, detached from any centralized corporate directory service, but also to the use of FileVault 2 by a single individual on a single system.

Target Environment

This methodology may be a good fit for organizations that have adequate controls in their data center and network services, and a user community that can benefit from the ability to drive much of their own usage. With this type of deployment, the end user typically takes control and responsibility for everything, with IT staff possibly providing guidance as needed, but largely playing a hands-off role.

Users with full administrator capabilities on their own systems, and users without network access requiring recovery support, are the perfect candidates for this methodology.

Deployment

With a Self-Service deployment, end users get freedom and flexibility to enable, modify, and disable FileVault 2 when and how it makes most sense to their workflow and needs.

A requirement for this deployment is that the individual must have the local rights to modify the system configuration and setup FileVault 2. These modifications would all be performed by using System Preferences > Security & Privacy > FileVault.

Enabling FileVault 2 requires the user to unlock the system preference with an account that can acquire the authorization right:

`system.preferences.security`.

Authorization rights are defined in the Authorization Database (`/etc/authorization`), which is, by default, granted to members of the local administrator group. If end users aren't full administrators of their system, they could be authorized to modify the settings in Security & Privacy preferences. However, that means users could modify any security setting on their system. There isn't a specific right for just FileVault management

that could be granted to an unprivileged user. However, privileged users could assist an end user who's enabling FileVault by entering their credentials for a one-time authorization.

Personal Recovery Key (PRK)

Setup can take one of two paths depending on corporate restrictions or recovery scenarios: either storing the PRK offline, or storing it protected on Apple servers. Corporate regulations may require that all encryption and recovery key storage be maintained inside the corporate controlled infrastructure. Other corporations may place a higher requirement on the need to recover at any time from anywhere, especially when completely disconnected from network resources.

Choosing offline storage means the randomly generated PRK must be written down or captured via a screenshot and protected off the computer. There's no automated system retention of the PRK, and its value could be lost permanently if not noted or written down during the FileVault enablement process.

Choosing protected storage of the PRK on Apple servers takes the user through a process of answering three security questions to generate a symmetric key used to wrap the PRK before submission for storage. The symmetric key can be unwrapped only with the exact same answers to those questions, known only by the end user. The answers shouldn't be too complex for the user to recall when needed.

Recovery

Daily workflow with FileVault 2 enabled doesn't change because users regularly log in to the system using their Directory Services account password. They enter their password at the pre-boot EFI-Login and the system initiates password-forwarding when the operating system is up and requiring login credentials. Password-forwarding eliminates the need for users to log in twice after a cold boot.

But what if users forget their account password or an IT department needs access to the system without an authorized FileVault user present? Use of the PRK provides this safety net.

Entering the wrong password three times at the pre-boot login prompts a message below the password field that states: "If you forgot your password, you can reset it using your recovery key." Users must click the triangle next to the message to reveal the Recovery Key field, along with the computer's serial number and record number. End users simply call AppleCare and provide the required information—including the exact responses to the three security questions provided at setup. Providing end users with AppleCare for support relieves IT departments from the day-to-day calls, allowing them to focus on enhancing overall corporate infrastructure.

Advantages

The Self-Service methodology has the biggest advantage in reducing IT costs and empowering end users. A positive user experience contributes significantly to corporate security compliance. A negative user experience caused by complicated regulations and resource-draining tasks drives end users to work around the protective security measures.

Choosing offline storage enables all types of scenarios, from the end user simply writing down the PRK and storing it in a secure location to a slightly

more integrated approach of submitting the PRK to a help desk service through a web form or phone call. All scenarios for offline storage allow the organization to retain and retrieve the PRK completely within their own control.

Choosing protected storage of the PRK on Apple servers gives end users flexibility to recover their systems anytime and anywhere, while also providing strong protection of the key. Failure by end users to succeed beyond the FDE pre-boot authentication means their computer isn't running an OS yet and remote access services aren't available, therefore help desk staff wouldn't be able to reach out over the network to assist them. Storing the PRK on Apple servers not only increases flexibility for the end user, but also ensures that the key is heavily guarded—without requiring dedicated corporate IT resources.

Disadvantages

The biggest challenges with the Self-Service methodology include automating the escrow of the PRK and ensuring that users remain compliant.

The offline storage approach doesn't provide the organization with an automated process to escrow the PRK back to a centralized server for later retrieval. Any desire for a sophisticated escrow and retrieval of the PRK requires organizations to perform their own integration with client management services or an in-house toolset. If required, the additional costs in time and money to integrate the PRK into existing enterprise escrow services may prohibit this approach.

Inexperienced users or those new to OS X may not be comfortable providing their own management and recovery. Many users switching from other platforms may be accustomed to calling the help desk for technical help. Shifting responsibility to end users may not provide the best user experience for these types of users.

Self-Service deployment prohibits an organization from taking advantage of the Centralized methodology, which uses an Institutional Recovery Key (IRK). These two methods are exclusive.

The Cafe or Centralized methodologies should be considered if the disadvantages of a Self-Service deployment exceeds the advantages to the end user.

Summary

The Self-Service methodology shifts the responsibility for setup, management, and recovery to end users, which significantly reduces IT costs and improves the user experience. Satisfied users are more likely to be compliant users, which further lowers IT costs.

Inexperienced users switching from other platforms may not be comfortable providing their own management and recovery. Shifting

complete responsibility to the end users may not provide the best user experience for these types of users. This methodology also requires administrative privileges to enable and disable FileVault 2, which may not be desirable or even allowed for corporately controlled assets.

Resetting account passwords and ensuring authorized access whenever needed—such as when users are completely detached from any network—provides a significant benefit to both end users and corporate IT.

The Self-Service methodology is the best fit for organizations that have adequate controls in place within their data center and network services, along with a user community that can benefit from the ability to drive much of their own usage.

Cafe Methodology

Assisted Deployment

The Cafe methodology is an assisted deployment that works well for organizations with user communities that need to control much of their own usage, but have strict corporate or industry mandates that require IT staff to assist end users in ensuring that data is properly protected. Using this approach, IT administrators can provide security expertise, deployment options, and recovery assistance as needed, but leave enablement and usage tasks to the end users.

Target Environment

This methodology provides the best compromise among user control, corporate policy, and safe protection and recovery of user accounts. Both IT staff and end users are involved in the process, and formal checks and balances can still be put into place to ensure compliance with corporate or industry requirements.

Dedicated and knowledgeable IT staff can develop training and guidance material for end users to follow when necessary. IT staff can also develop and provide local tools and services to monitor the state of encryption and prompt users to take action when appropriate.

Large organizations new to the platform might use this methodology to transition to a more managed environment while still providing a positive end-user experience. Users who work in a relatively free IT environment may resist stronger controls from a corporate IT department. This sharing of responsibilities reduces anxiety on both sides, giving time for improvements on the infrastructure side while improving the end-user experience.

Deployment

With an assisted deployment, end users still get significant freedom and flexibility, but share that responsibility to enable, modify, and disable FileVault with corporate IT departments. Shared freedom and responsibility require cooperation between end users and corporate IT departments.

Administrative users with full system access and intermittent network access are generally the best candidates for this methodology. The organization can still enable IT administrators to help users who don't have or want those broader responsibilities. The shared responsibility allows for flexibility in setup and recovery that may not fit into the Self-Service or Centralized methodologies.

Requirements for deployment setup depend on the recovery method chosen to best serve the user's workflow—Personal or Institutional. Although it's not possible to have both Personal and Institutional recovery methods active at the same time, it's possible to switch back and forth

between the two. This switching flexibility requires decrypting the volume under one method and re-encrypting under the other method.

Recovery

The recovery method allows for personal recovery using the Personal Recovery Key (PRK) and Institutional Recovery Key (IRK) to enable users to reset an account password on systems locked at the standard login window (not for pre-boot login).

Using a PRK allows for more granularity relating to storage and corporate IT involvement if necessary. Refer to the "Self-Service Methodology" section for more information on the PRK—and throughout this document for in-depth details.

Password Resets

Using an IRK can be an effective way for corporate IT departments to assist end users with password resets without controlling the whole experience. However, the IRK can't be used by end users for pre-boot recovery because the system doesn't have access to the volume where the IRK is stored. IT staff, such as from a help desk, can provide the preset Master Password to end users for keying at the standard login window text field. If an IRK is present and the user fails to log in at the standard login window, the Master Password can be communicated over the phone as a standard support call or even restricted to secure and authenticated access to a protected corporate website. This allows end users to collaborate with IT staff to reset their own password, which is especially useful in an offline situation.

The Master Password is the password protecting the FileVault Master Keychain (FVMK) containing the complete FileVault Master Identity (FVMI).

End users can also securely access and retrieve the PRK from a corporate-protected website by using the hidden Recovery HD. All systems running FileVault 2 have a hidden Recovery HD partition designed specifically for this purpose. This partition is not seen under the Startup Disk system preference or viewable by the Disk Utility application. End users can restart their Mac and immediately press Command–R on the keyboard. Once the Mac begins to start into Recovery, they can let go of the keys. Recovery will load with a restricted guest-only access environment and users can open Safari to access and authenticate to the corporate-protected web portal. Once authenticated, they can retrieve the PRK from corporate secured storage.

Advantages

The Cafe methodology has the biggest advantage in sharing responsibilities between end users and corporate IT departments. A positive user experience contributes significantly to corporate security compliance. This approach allows for selectively giving responsibility according to need and capabilities for both end users and IT staff.

IT departments can front-load the majority of resource requirements by developing focused guidance and a recovery key portal to give end users

controlled and protected access for performing the tasks themselves. IT staff can also improve guidance, recovery, and monitoring while keeping sensitive information themselves—no storage or use of keys outside the company infrastructure.

With the built-in Recovery HD and its restrictive guest-only access, end users can safely reach corporate-protected content without exposing sensitive information or jeopardizing the integrity of the system or encrypted volume.

Disadvantages

The biggest challenges with the Cafe methodology are determining what responsibilities to assign to each party and the IT resources needed to develop web-based guidance content with a secure key escrow and retrieval portal.

Most organizations already use web-based content for training, technology instruction, and guidance, so providing additional user content on FileVault may be negligible on resources. Any secure key escrow and retrieval portal requires specific development to capture and deliver either the PRK itself or the Master Password for the IRK.

The main drawback is the need for resources and development to handle the IT side of the shared responsibilities. An organization using a client management solution with solid OS X integration can frequently perform all or most of the IT tasks from within the vendor's toolset.

Summary

The Cafe methodology works well for organizations with a capable user community, but where limited IT staff involvement may be necessary. Using this approach, IT departments can provide security expertise, deployment options, and recovery assistance while having end users do the actual enablement and recovery.

Knowledgeable IT staff can develop training, guidance, and a Recovery Key Escrow/Retrieval portal for protected access by end users. Providing a flexible approach, the capture and retrieval of the recovery key can change according to the methodology deployed for each specific user. Inexperienced users switching from other platforms may feel more comfortable relying on IT staff for management and recovery.

The Cafe methodology is the best fit for organizations that have a strong IT resource pool, but need to balance that with the need or demand by end users to drive much of their own usage of FileVault 2. IT departments can play the role of subject-matter experts, giving advice and guidance to end users without restricting the end-user experience.

Centralized Methodology

IT-Controlled Deployment

The Centralized methodology is an IT-controlled deployment that may be the best choice for organizations that must give IT staff stringent control over user systems where all data on user devices must be protected using full disk encryption (FDE), yet recoverable only with IT involvement. This form of deployment requires IT staff to control all aspects of creation, management, and recovery of the encrypted volumes. Full IT involvement ensures proper deployment, auditing of systems, and enforced compliance with corporate or industry mandates.

Target Environment

This methodology is best for highly regulated or secure environments. Every phase of the process is controlled and performed by authorized IT staff to avoid any missteps by end users and to ensure adherence to all regulations at all times.

Large organizations with strong platform knowledge will embrace this methodology under a tightly managed environment while typically using equally strong client management tools. Many client management tools support the deployment and management of the FileVault Master Keychain (FVMK), FileVault Master Identity (FVMI), and related Apple-provided command line interface (CLI) tools such as `security` included in OS X.

An enterprise Certificate Authority (CA) for the provisioning and escrowing of the FMVI is crucial for FDE-solution deployments. With separation of responsibilities between security-focused administrators, workgroup managers, and help desk staff, authorization and access to security sensitive information must also be controlled.

Deployment

With the Centralized methodology, proper generation, protection, and management of the FVMI is critical to IT staff staging access to the fully encrypted volume. This is the most complex of the three methodologies and is typically performed completely by IT staff on behalf of the end users or by advanced OS X users. The toolsets used by the organization can be diverse as long as a few very specific details remain as required by OS X.

An understanding of the FVMI and the purpose of each related component determines the success and manageability of the FDE protection and access. The FVMI provides a pair of asymmetric keys (public and private) that are used to wrap and unwrap the Key Encryption Key (KEK) for authorized access to the encrypted volume. (See "Appendix A: Architectural Overview" for more information.) This IT access method is of particular interest because it doesn't require any authorized user credentials to unlock the fully encrypted volume.

The organization must first provision an FVMI either using its own enterprise CA, the built-in GUI method for "Setting Master Password," or CLI command `security` to create the FVMI-populated FileVaultMaster.keychain.

Provision the FVMI

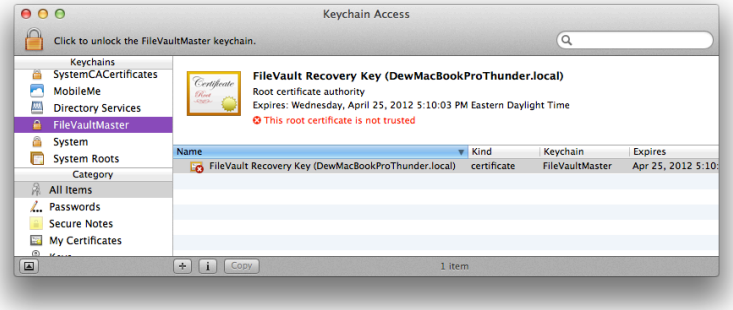
It's best for an organization to use its own enterprise CA for the proper provisioning of an FVMI. Private key escrow is the primary benefit. An FVMI must be provisioned according to the requirements in "Appendix A: Architectural Overview." Any deviation from the required attributes and attribute values prevent the proper use of the FVMI for key wrapping and unwrapping.

If the infrastructure and IT resources can support the complexity and granularity, it's best to provision an FVMI for every system that will be enabled for FileVault 2. This isn't a technical requirement for FileVault 2, but provisioning at this level allows the organization to selectively control keys for each system, potentially even from separate intermediate CAs. The opposite approach would be the issuance of a single FVMI for all systems enabled for FileVault. If the single FVMI were to be compromised for any reason, it would make all the organization's systems vulnerable—the single identity could unlock or decrypt all computers enabled with that same FVMI.

Deploy the FVMI

A keychain file named FileVaultMaster.keychain—which should ONLY contain the certificate component of the FVMI—should be pushed out to existing clients using client management tools and included within the disk image when imaging new systems. It must be placed on the file system at the following path: /Library/Keychains/FileVaultMaster.keychain.

Enabling FileVault 2 requires the user to unlock the system preference with an account that can acquire the authorization right (`system.preferences.security`). These rights are defined in the Authorization Database (`/etc/authorization`), which by default are granted to members of the local administrator group.



Only the certificate is included because the public key embedded in the certificate is all that's needed to enable IT-controlled FileVault 2. This mitigates any potential malicious compromise of the FVMI by not including the most sensitive component—the private key. The private key is needed only when performing a recovery on the encrypted volume. If the full FVMI were included on the system and an unauthorized individual were to gain

access to it, the individual would be able to unlock or decrypt that system and any other system enabled with the same FVMI.

OS X requires an administrator with rights to setup FileVault 2 using the GUI and modify the system configuration to turn on or off. This is performed by clicking Turn On FileVault from System Preferences > Security & Privacy > FileVault. Because the FVMK—a special credential store—is populated with only the certificate and is already staged at `/Library/Keychains/FileVaultMaster.keychain`, the administrator will be notified as shown in the following screenshot.

This notification indicates that OS X has recognized the appropriately deployed FVMK as the IRK and the public key from the certificate will be used to wrap the KEK and store it for authorized IT access at a later time. If there was no FileVaultMaster.keychain deployed, or the keychain contained the full FVMI, the process would default to setup of a PRK as noted in the "Self-Service Methodology" section. FileVault 2 recognizes either the PRK or the IRK, but not both.

Setting Master Password

An organization can also provision and deploy a FVMI directly on a device with "Setting Master Password." Setting the Master Password simply refers to requesting that the OS generate a random FVMI, store it inside a freshly created FileVaultMaster.keychain, and set the password for the FVMK to the password entered as the Master Password. Any desire or need to escrow the keychain or FVMI can be performed after the Master Password has been set.

Scripting FVMI Provisioning

Another method for provisioning an FVMI is by using the `security` CLI command. Use of the `security` command is explained in more detail later in this section.

Alternate Administrator Account

It's also highly suggested that an alternate administrator account be configured and enabled for FileVault 2. This can be used as yet another method for administrators to reset account passwords and unlock volumes without performing a recovery on the fully encrypted volume. This also allows IT staff to assist end users with password resets even with minimal knowledge of FileVault 2.

Recovery

Daily workflows aren't affected with FileVault 2 enabled because users log in to the system using their Directory Services account password. They enter their password at the pre-boot EFI-Login and the system initiates password-forwarding when the operating system is up and requiring user login credentials. Password-forwarding eliminates the need for users to log in twice after a cold boot.

What if IT needs access to the system without any authorized FileVault user present? Use of the IRK provides this safety net for IT staff.

Password Resets

To reset an account password, the encrypted boot volume must first be unlocked and active, which isn't possible if there's only one user. Because the computer first performs a pre-boot using EFI, it requires an authorized user to unlock the volume and continue the boot process. This is why it's important to have an alternate administrator account configured and enabled for FileVault 2 to perform a password reset.

When the alternate administrator unlocks the volume and accesses the system, a standard password reset on the user's account (via Users & Groups preferences) can be performed. When the password is reset from OS X, the corresponding user access to FileVault 2 is also reset to use the new password—a Derived Encryption Key (DEK) generated from the user's password.

If the user's password is reset from the external Directory Services side (for example, in Active Directory), the user's new password won't unlock the volume at pre-boot. Again, this would require an alternate administrator to unlock the volume, log out, and turn the system over to the user to enter the new credentials at the login window.

FVMI-Based Recovery

To perform any recovery or forensic-like access to the fully encrypted volume at any time requires the full FVMI—both the certificate (with embedded public key), and the private key in a valid keychain (for example, FVMI.keychain). This assumes all steps to acquire the FVMI from the enterprise CA have already been performed by the authorized IT administrator. It's easiest to create an empty keychain file and automate the importing of the FVMI. A keychain can be created and the FVMI imported using several different techniques, including fully scripted using the `security` command. One method using just the Terminal application and CLI commands would be:

1. Create a new FVMI keychain and store on a USB memory stick.

   a. `# security create-keychain -P /Volumes/<PathToFile>/FVMI.keychain`

b. This creates the keychain named FVMI.keychain at the entered path.

c. When prompted, enter the desired password to protect the keychain.

2. Import the FVMI exported from the enterprise CA (that is, FVMI.p12).

a. `# security import /PathTo/FVMI.p12 -k /Volumes/<PathToFile>/FVMI.keychain -f pkcs12 -P`

b. Enter the passwords protecting the keychain and the .p12 files as prompted.

3. Boot the computer you wish to recover into the Recovery HD.

a. Press Command–R on the keyboard to boot into the Recovery HD.

2. Open Terminal from the Utilities menu.

3. Identify the Logical Volume UUID (LvUUID) of the encrypted volume.

a. See "Appendix A: Architectural Overview" for more information on CoreStorage volume management.

4. Insert the USB memory stick.

5. Unlock the FVMI keychain.

a. `# security unlock-keychain -p <password>`

`/Volumes/<PathToFile>/FVMI.keychain`

6. Unlock and mount the CoreStorage encrypted target volume.

a. `# diskutil cs unlockVolume <LvUUID>`

`-recoverykeychain`

`/Volumes/<PathToFile>/FVMI.keychain`

After completing the steps, the volume is unlocked and mounted for full access by any tools or services. This allows for any updating and modifications to the files on the volume (for example, patching system or application files).

It's not possible to perform the unlock with the FVMI if attempting to boot from the same encrypted volume. This is why it's necessary to first boot from an alternative drive such as the built-in Recovery HD. The Recovery HD is an immutable image, validated for integrity, and mounted as a read-only boot volume. Any integrity issue would prevent its use and then default to an Internet-based Recovery—if it's a newer OS X–based system and supports Internet Recovery in firmware. This also prevents anyone from modifying the hidden Recovery HD, such as by installing software or altering the configuration.

Advantages

The Centralized methodology offers the biggest advantage in providing full control for IT staff to pre-stage the FVMI and access the encrypted volume at any time. A positive user experience contributes to corporate security compliance, but in this case, IT departments have stringent control of all phases using FileVault 2. Users are typically not involved in any phase of the management of FileVault 2 other than just the day-to-day use.

Using its own enterprise CA for the provisioning and private key escrow of the FVMI is what makes this approach so appealing to IT departments. As long as the CA can issue identities to match the requirements noted in "Appendix A: Architectural Overview," the organization can choose what level of granularity fits their security needs and IT resources. With FileVault 2, an organization isn't locked into a single approach or one method for escrowing keys.

Staging the FVMI onto an image destined for restoration to new systems, or pushing it out to all existing OS X systems, lets IT departments always access those volumes without an authorized user's credentials. Much of the process can be scripted and integrated into existing workflows for imaging and deployment.

Disadvantages

This is the most complex of the three methodologies and requires IT staff or advanced OS X user involvement throughout all phases of managing FileVault 2. The toolsets used by the organization can help, but no solution can fully enable FileVault 2 without the end users being present to enter their account password.

With OS X, push-button automation isn't provided in generating and escrowing the FVMI across multiple systems from a centralized location (that is, no built-in console support for the data center). If following best practices, it does require an organization to use multiple enterprise services and integrate the features through internal development or use of additional IT resources.

Summary

The Centralized methodology is the best choice for organizations that must have IT staff retain stringent control over user systems and ensure that FDE is recoverable only under IT control. This form of deployment requires IT staff to perform all aspects of creation, management, and recovery of the encrypted volumes.

Although full IT involvement ensures proper deployment, auditing of systems, and enforced compliance to corporate or industry mandates, it means increased IT costs and more complex planning and execution.

IT staff must become knowledgeable about the provisioning, escrowing, and use of the FVMI for proper staging and recovery using the IRK.

Compliance

There are usually several industry mandates enforced within an organization related to deployment of computers and the integration with data-at-rest protection. Two such requirements discussed most often are the need for FIPS 140-2 Conformance Validation and support of Assistive Technology (commonly referred to as Section 508 or Accessibility).

FIPS 140-2 Conformance Validation

The U.S. and Canadian governments maintain a validation program, the Cryptographic Module Validation Program (CMVP), within the National Institute of Standards and Technology (NIST) and Communications Security Establishment Canada (CSEC), for the proper review and validation of cryptographic modules used by commercial products. The validation performed on the cryptographic module itself and any operating system, service, or application that uses a validated cryptographic module is said to be FIPS 140-2 Compliant.

OS X Lion 10.7 is the first OS release to implement FileVault 2 (FDE). The new kernel-based cryptographic module, CoreCrypto—upon which FileVault 2 is built—wasn't submitted for FIPS 140-2 Conformance Validation under OS X Lion.

The cryptographic module used by OS X Snow Leopard 10.6 for services such as encrypted disk images, S/MIME, and SecureTransport achieved FIPS 140-2 validation on March 9, 2011. Apple has achieved revalidation of that module still present in OS X Lion. Any use of Legacy FileVault alone, or on top of FileVault 2, while running OS X Lion would use the revalidated module and provide FIPS 140-2 compliance, but would incur a performance hit.

The cryptographic modules in OS X have since been submitted and are expected to achieve validation for the OS X Mountain Lion 10.8 release. FileVault 2 under OS X Mountain Lion 10.8 will be FIPS 140-2 compliant.

Section 508 (Accessibility)

The architectural approach of FDE with pre-boot authentication prevents support for accessibility and may prevent a user with disabilities from using the FDE services. At the point of pre-boot, this device doesn't yet have the OS running, therefore any technologies reliant on the OS (such as accessibility) aren't yet available for use by the device. Organizations requiring accessibility support are better served by continuing to use Legacy FileVault on OS X Lion for protection of the user's home directory while enabling the use of accessibility services.

Conclusion

Breaches in data security can damage an organization in many ways. Protecting individual and corporate sensitive data is crucial, and that protection is improved when data is encrypted and out of the reach of unauthorized individuals. If users have special privileges, or aren't restricted by policy to store data only in protected areas, organizations risk inadvertent data exposure by storing files outside an encrypted boundary. Encrypting the entire disk ensures that all information is protected. With FileVault 2 enabled, all information is safe on the hard drive.

Apple developed and continues to advance the FileVault 2 capabilities for all customers with three core design goals: usability, IT access, and performance.

FileVault 2 exemplifies Apple's hardware and software integration, helping individuals and organizations protect all user data on internal and external storage that's virtually transparent to the user and, in some cases, without the need for IT resources. End users must be able to protect their data without becoming security experts.

End users and administrators fear losing control of data. Personal and Institutional Recovery methods for controlled and authorized access to an encrypted volume provide the safety that everyone needs. It's critical to IT departments to have access to end-user volumes under all kinds of scenarios. Integrating support for an Institutional Recovery Key (IRK) with a FileVault Master Identity (FVMI) extends the capabilities beyond just an individual's self-service recovery method using the Personal Recovery Key (PRK).

Performance had limited data-at-rest solutions in the past, but FileVault 2 shines where it means the most—architectural design, optimized cryptography, and use of hardware acceleration. FileVault 2 enhances the individual's and organization's experience while providing the expected level of protection. This fact alone sets it apart from the pack.

Corporate IT departments have many diverse options to protect sensitive data, but knowledge of the strengths and tradeoffs between options ensures that the best approach is deployed. Using FileVault 2, a properly provisioned and escrowed recovery key, and a working knowledge of deployment methodologies, an organization can ensure sensitive data is always protected from unauthorized access.

Appendix A: Architectural Overview

FileVault 2 is best defined as the systemwide management of user and administrator access to fully encrypted volumes—boot and data volumes. It's not a single process or component, but rather the management of multiple components into a single feature of OS X. That's why it's important to understand what each component is and how they relate as a single feature.

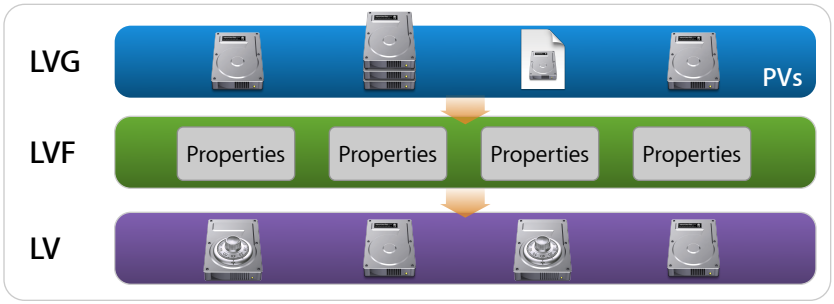
FileVault 2 consists of three essential components to provide its overall capabilities for large-scale deployments:

- CoreStorage
- Key management
- Recovery methods

CoreStorage

CoreStorage is Apple's architecture for an advanced logical volume manager (LVM). CoreStorage isn't the same as FileVault, but instead is the logical volume management that FileVault uses for the storage and retrieval of encrypted bits. Under the OS GUI, CoreStorage support for FileVault is also referred to as CoreStorage full disk encryption (CSFDE).

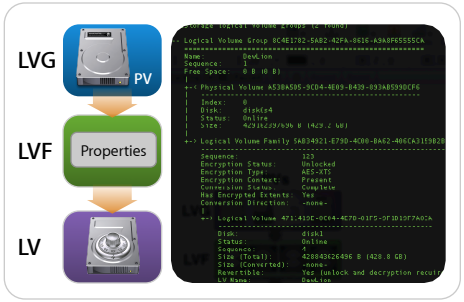
Detailed coverage of CoreStorage isn't in the scope of this paper, but it's important to have a high-level understanding of the architecture and relationship to FileVault.



The CoreStorage architectural diagram reflects multiple physical volumes (PVs) within each logical volume group (LVG), resulting in multiple logical volumes (LVs)—each selectively encrypted or not.

CoreStorage components:

- Physical volume (PV)
- Logical volume group (LVG)
- Logical volume family (LVF)
- Logical volume (LV)



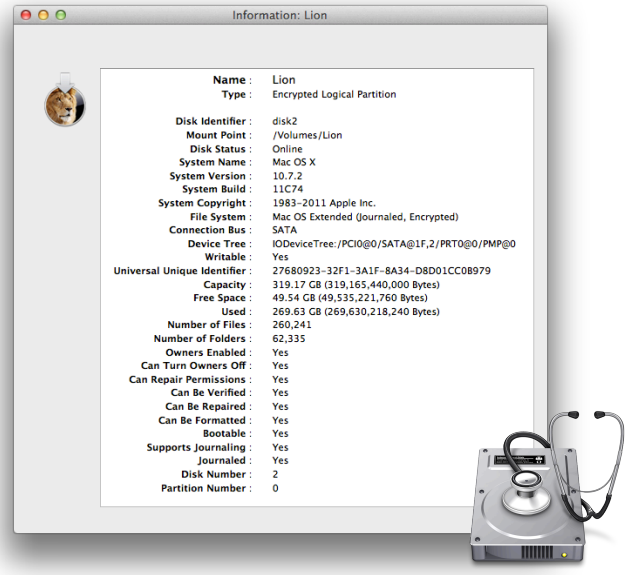
A physical volume (PV) is part of a logical volume group (LVG), containing a

logical volume family (LVF), describing the attributes of the resulting logical volume (LV).

The CoreStorage architecture calls for multiple LVGs, LVFs, and LVs, but for simplicity, let's focus on a single physical volume (PV) in a single logical volume group (LVG) resulting in a single encrypted logical volume (LV).

Each CoreStorage volume is individually referenced using its own universally unique identifier (UUID) formatted as a 128-bit value guaranteed to be unique over space and time. The standard format for UUIDs is represented in ASCII as a string punctuated by hyphens (for example, 486E9812-167F-4129-B1AA-E6A041C69EA6).

CoreStorage objects can be viewed and manipulated using either the Disk Utility application or command line interface (CLI) tool called diskutil.




```
Usage:  diskutil [quiet] coreStorage|CS <verb> <options>
       where <verb> is as follows:

list              (Show status of CoreStorage volumes)
info[rmation]     (Get CoreStorage information by UUID or disk)
convert                 (Convert a volume into a CoreStorage volume)
revert            (Revert a CoreStorage volume to its native type)
create            (Create a new CoreStorage logical volume group)
delete            (Delete a CoreStorage logical volume group)
createVolume      (Create a new CoreStorage logical volume)
unlockVolume      (Attach/mount a locked CoreStorage logical volume)
changeVolumePassphrase(Change a CoreStorage logical volume's passphrase)

diskutil CoreStorage <verb> with no options will provide help on that verb
```

To gather information about existing CoreStorage volumes, execute the following diskutil command in a Terminal window:

```
$ diskutil cs list
+-- Logical Volume Group 486E9812-167F-4129-B1AA-E6A041C69EA6
    =========================================================
    Name:         Lion
    Sequence:     1
    Free Space:   0 B (0 B)
    |
    +-< Physical Volume ABF9FDE7-8481-43FB-A9BB-1E316C182DC1
    |   ----------------------------------------------------
    |   Index:    0
    |   Disk:     disk0s2
    |   Status:   Online
    |   Size:     319484211200 B (319.5 GB)
    |
    +-> Logical Volume Family 02064501-2BE1-442E-B317-C21379CE5DD2
        ----------------------------------------------------------
        Sequence:                 13
        Encryption Status:        Unlocked
        Encryption Type:          AES-XTS
        Encryption Context:       Present
        Conversion Status:        Complete
        Has Encrypted Extents:    Yes
        Conversion Direction:     -none-
        |
        +-> Logical Volume F9EC4CFD-9440-4A43-A3F9-83F670153DFC
            ---------------------------------------------------
            Disk:                 disk2
            Status:               Online
            Sequence:             4
            Size (Total):         319165440000 B (319.2 GB)
            Size (Converted):     -none-
            Revertible:           Yes (unlock and decryption required)
            LV Name:              Lion
            Volume Name:          Lion
            Content Hint:         Apple_HFS
```

FileVault encryption utilizes the AES-XTS-128 encryption algorithm with a 256-bit key following NIST guidance in NIST SP 800-38E.

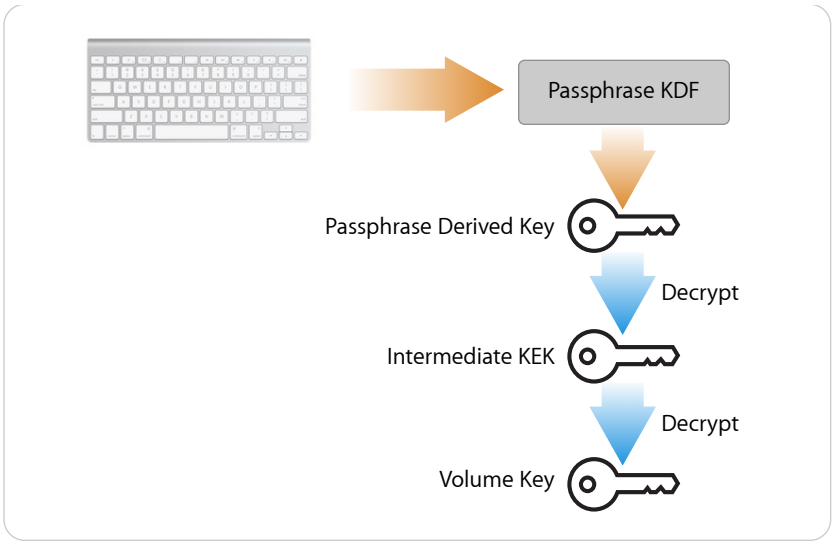
The status or existence of encryption on any volume can be obtained by executing the following command with the appropriate UUID:

```
$ diskutil cs info 02064501-2BE1-442E-B317-C21379CE5DD2
      CoreStorage Properties:
   Role:                     Logical Volume Family (LVF)
   UUID:                     02064501-2BE1-442E-B317-C21379CE5DD2
   Parent LVG UUID:          486E9812-167F-4129-B1AA-E6A041C69EA6
   LVF Encryption Status:    Unlocked
   LVF Encryption Type:      AES-XTS
```

This sample volume is fully encrypted using AES-XTS and currently mounted and unlocked.

The execution of diskutil cs info on an LVF UUID also provides a way to locate and identify the parent LVG UUID. When executing this same command with the LV UUID, the full chain of UUIDs is listed from the LV UUID, to the parent LVF UUID, to the parent LVG UUID, along with the status conversion to an encrypted volume—noted here as "Complete":

```
$ diskutil cs info F9EC4CFD-9440-4A43-A3F9-83F670153DFC
      CoreStorage Properties:
   Role:                     Logical Volume (LV)
   UUID:                     F9EC4CFD-9440-4A43-A3F9-83F670153DFC
   Parent LVF UUID:          02064501-2BE1-442E-B317-C21379CE5DD2
   Parent LVG UUID:          486E9812-167F-4129-B1AA-E6A041C69EA6
   Device Identifier:        disk2
   LV Status:                Online
   Conversion Status:        Complete
   Content Hint:             Apple_HFS
   LV Name:                  Lion
   Volume Name:              Lion
   LV Size:                  319165440000 B
```



Key Management

This section looks at the key management under FileVault 2 by the OS and doesn't refer to the institutional management of recovery keys. Details on institutional recovery key provisioning and management are covered later in this appendix.

For any encrypted volume using CoreStorage, there's a string of wrapped encryption keys to protect the data and provide for authenticated access, resetting of user passwords, and support recovery access. The emphasis here is on the use of FileVault for the boot volume, but CoreStorage also provides for disk passwords on data volumes where a single unique password may also be useful for deployments.

Key management involves the management and use of both encryption keys and recovery keys. There are three levels of encryption keys and two variations of recovery keys.

Understanding the purpose and flexibility of this three-level key wrapping architecture begins with an understanding of the three keys involved:

- Volume encryption key (VEK)
- Key encryption key (KEK)
- Derived encryption key (DEK)

Volume Encryption Key

At the lowest level, CoreStorage operates on 512-byte logical blocks using a symmetric volume encryption key (VEK). All cryptographic operations on a logical volume are unique to that volume because each VEK is randomly generated for each independent logical volume. This unique VEK, combined with a per-block tweak, provides greater strength in encrypting all data blocks. Because access to the data on a volume depends on its VEK, it must remain constant for the life of the volume. Ultimately, the only key that CoreStorage really needs is this VEK.

The encryption performed at this block level is AES-XTS-128, which uses a 256-bit VEK. XTS is a new AES block cipher mode recommended by the National Institute of Standards and Technology (NIST) for confidentiality on storage devices. The acronym XTS stands for the XEX Tweakable Block Cipher with Ciphertext Stealing. According to NIST, "In the absence of authentication or access control, XTS-AES provides more protection than the other approved confidentiality-only modes against unauthorized manipulation of the encrypted data." For more detail on XTS-AES, go to http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf.

A three-level key indirection approach mitigates the need for an institution to rekey a volume as may have been required on other platforms in the past. If an organization wants to rekey a volume at this low level, it requires a full decryption and re-encryption of the entire volume. This concept is generally referred to as "un-vaulting and re-vaulting" if FileVault is used on that volume.

Key Encryption Key

During encrypted volume initialization, a random symmetric intermediate key encryption key (KEK) is generated, encrypts (wraps) the VEK, and is stored in the CoreStorage metadata. This intermediate key allows for indirection to support the requirements mentioned earlier in the design approach. This indirection allows derived encryption keys (DEK) to be changed independently of each other and the VEK. It also allows the VEK to be changed independently of the DEKs. Once the VEK is wrapped with the KEK, the resulting encrypted blob is stored inside the volume's CoreStorage metadata for actual retrieval and decryption of data.

Derived Encryption Key

At the highest level, a derived encryption key (DEK) must be available to begin the chain of unlocking the remaining two keys and resulting in the decryption and access to the encrypted volume. Any DEK can be independently changed without affecting the KEK itself or even the VEK. This distinction becomes very powerful in enabling various ways to access the same encrypted volume without exposing any keying material (or in the case of an Institutional Recovery, not even needing to know a specific user's credentials).

A given CoreStorage volume must support multiple cryptographic users—each with their own DEK. A cryptographic user may be, but doesn't need to be, the same as an OS authenticated user. A cryptographic user is simply all the parameters needed for a given key to unlock the volume. Entry of any of the valid cryptographic user credentials stored for a given volume can unlock that volume.

There are two methods for deriving this top-level key:

- Passphrase-based DEK
- X.509 Identity-based DEK

Passphrase-based DEK

When a passphrase is entered, it's converted to a key with the RSA Password-Based Key Derivation Function (PBKDF2) using SHA256-HMAC for the internal pseudo-random function specified as PKCS#5 v2.0 and also as RFC2898. This resulting key is used as the DEK.

There are two times when a passphrase is used in generating a DEK:

- Login password–based DEK
- Disk password–based DEK

A major goal of the system is to be resistant to offline passphrase guessing attacks against the encrypted volume. If the user's passphrase was only trivially hashed, an attacker could mount precomputed or brute-force attacks against it, potentially with specialized hardware. To make this attack more difficult, a common cryptographic technique is used that iterates a hash function over the passphrase many times, mixing the result in a way that's hard to parallelize.

Login Password–based DEK

An OS authenticated user is any user who's been enabled to unlock the encrypted volume. The users' login password is their individual directory service account password. Because FileVault must support unlocking by multiple OS authenticated users, each login password is converted to its own unique DEK, which individually wraps the KEK and is stored in the user's bundle within the volume's CoreStorage metadata, as noted for later retrieval.

Support for multiple OS authenticated users means that any FileVault-enabled user can unlock the whole volume, as would be expected. However, if the system environment warrants additional cryptographic separation and containment of user files, consider using Legacy FileVault (FileVault 1) simultaneously. Coverage of Legacy FileVault on an OS X system using FileVault FDE is out of scope of this paper. In brief, it's the

continued use of FileVault 1, container-based encryption of the user's home directory, on top of FileVault 2 providing full disk encryption.

Disk Password–based DEK

Consider the use of a disk password a special use case where a single passphrase is used to generate a single DEK to wrap and unlock the corresponding KEK or volume. A volume protected in this manner could be thought of as a FileVault-enabled volume with a single, non-OS authenticated user. A disk password prohibits separate users from unlocking the volume with their own password and prevents the use of an IRK for access to that volume.

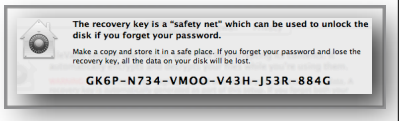
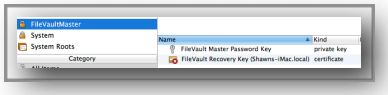
X.509 Identity-based DEK

An institution needs a method to unlock access to encrypted volumes without using a FileVault-enabled user's password. This is where the power of public key infrastructure (PKI) comes in. Rather than trying to keep a central IT passphrase in sync across multiple devices, an institution can use a more secure method of a provisioned X.509-based identity (FVMI). Rather than using PBKDF2 to convert a passphrase into a DEK, or using the PRK, FileVault uses the asymmetric public/private key pair of FVMI. The public key is used to wrap the KEK and the private key is used to unwrap the KEK.

Recovery Methods

Recovery is critical for both the individual and the organization. Limitations or restrictions on resources used for recovery along with isolated systems without access over the network, in some cases, may determine which deployment strategy is best for all involved. If all FileVault-enabled users on a particular system forget their passwords, credentials aren't available, and there's no recovery key available, the encrypted volume can't be unlocked and the data is unaccessible. Data may be lost permanently, so proper recovery planning is essential.

Two recovery methods to consider can span more than one deployment strategy, but for simplicity, this paper covers the most common mappings between deployment strategies and recovery methods:

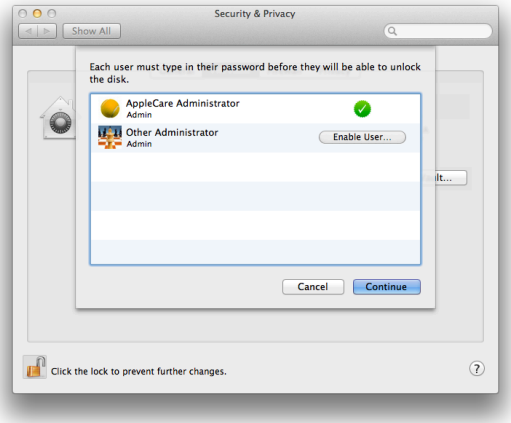
	**Personal Recovery** Generation and use of a random symmetric key value for a personal safety net
	**Institutional Recovery** Use of an X.509-based asymmetric key pair for corporate safety net and private key escrow

Personal Recovery

Personal recovery provides a safety net with the use of a Personal Recovery Key (PRK). Users start by going directly to the FileVault pane in Security & Privacy preferences.

Note: Enabling FileVault requires privileges for acquiring the right (`system.preferences.security`), which by default is granted to members of the local administrator group.

After clicking Turn On FileVault, if the Mac has multiple user accounts, the administrator is asked to identify the user accounts that are allowed to unlock the encrypted drive (to start the computer or recover from sleep or hibernation). By default, the current user logged in is automatically marked as enabled with a green checkmark.



If all user accounts are enabled for FileVault, the "Enable User" button no longer appears in the preference pane. All subsequently created accounts are automatically enabled for FileVault.

Users not enabled for FileVault unlock can log in to that Mac only after a FileVault 2–enabled user has unlocked the drive. Once unlocked, the drive remains unlocked and available to all users until the computer is shut down or goes into hibernation.

The administrator needs to enter the password, or have users enter their passwords, for each account that needs the ability to unlock the volume.

After enabling users for disk unlock, the PRK is shown.

If "Do not store the recovery key with Apple" is selected, then the 24-digit alphanumeric recovery key must be captured and retained out of band. This means the individual should set up the system to write the key value down and store it securely off the computer. Apple has no way to assist in recovery and all data would be lost if the individual or organization doesn't retain the recovery key randomly generated at enable time.

This method has no financial cost or required change to infrastructure. When proper care is taken for protection of the PRK, this can be a quick and easily deployed safety net for end users. However, if end users don't properly protect the recovery key—for example, if they write it down on a sticky note attached to their office desk—the recovery key could easily be found and an intruder could gain access.

How the key is protected and retrieved if users select "Store the recovery key with Apple" is discussed next.

Personal Recovery Key Protection

To protect the PRK and allow for a repeatable method to retrieve and unwrap this key, a symmetric PRK wrapping key is needed. This PRK wrapping key is derived by hashing the responses to all three selected security questions. Because this key is derived by hashing the responses, the exact responses must be provided to retrieve the PRK. If any of the responses differ even slightly, a valid symmetric key can't be derived. Because the answers to the questions are known only by the individual enabling FileVault, Apple has no way to assist in recovery—all data would be lost.

Personal Recovery Key Retrieval from Apple

When users forget their account password, or an IT staff member must access the protected volume without having an alternate enabled account, the PRK is needed and can be retrieved from Apple.

Entering the wrong login password three times prompts a message below the password field that states, "If you forgot your password, you can reset it using your recovery key." Users must click the triangle next to the message to reveal the Recovery Key field (which replaces the Password field) and AppleCare contact information, along with the computer's serial number and record number. The record number is generated when the PRK is originally sent to Apple.

To retrieve the PRK from Apple, contact AppleCare and provide the required information, which includes the serial number and record number. Then, the AppleCare representative can provide the PRK. Upon successful retrieval and entry of the recovery key, users are prompted to change their login password. When resetting the login password, users are also prompted to create a new login keychain. The previous login keychain, by default, was set to the same password as the account. When

resetting the account, there's no knowledge or storage of the previous password by OS X and the previous login keychain can't be unlocked. Only non–password protected content in the keychain is transferable to a new keychain.

After changing the login password, it's also recommended that users change their FileVault recovery key and upload the new one to Apple.

Changing a Recovery Key

From the FileVault pane in Security & Privacy preferences, click Turn Off FileVault to disable FileVault. Once turned off, FileVault begins to decrypt the drive. After decryption is complete, the Turn On FileVault button can be clicked. Doing so allows administrators to enable unlock-capable users, shows a new recovery key, and provides the option of sending this new key to Apple. The old key sent to Apple won't unlock the newly encrypted disk. If retrieval of the recovery key from Apple is necessary, only the latest one will be retrieved based on the serial number and record number displayed in the login window.

Institutional Recovery

Institutional Recovery provides corporate IT departments with a safety net and enables private key escrow with the use of an X.509 identity and its asymmetric public/private key pair.

Institutional Recovery Key

An IRK refers to a corporate provisioned X.509-based identity known as the FileVault Master Identity (FVMI). An FVMI consists of an X.509 certificate containing the public key along with the corresponding private key. FileVault wraps (encrypts) the KEK with the public key found in the certificate of the identity and unwraps (decrypts) the KEK with the private key.

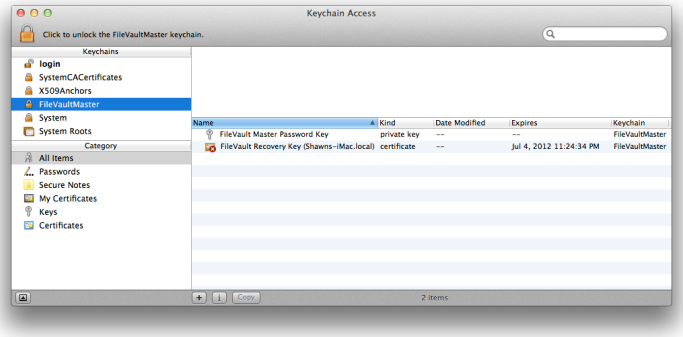
FileVault Master Identity

The FVMI is an integral component in the IT access to FileVault-encrypted volumes. This identity provides the system administrator or IT staff with an alternative way to access the user's encrypted volume without knowing any authorized user's password.

Only one FVMI is used per host. An institution can choose its granularity for FVMI provisioning. A single FVMI could be provisioned for the whole company if simplicity is desired, but that approach means any compromise of the identity could expose the remaining systems to a malicious entity. Provisioning a different FVMI for every machine mitigates that risk, but brings with it increased complexity in managing the association between FVMIs and individual machines. There's no clear best practice for the level of granularity an institution should take other than determining the highest level of granularity that can be securely managed.

The FVMI consists of three components that together represent an X.509 digital identity:

- Self-signed X.509 root certificate
- Public key (embedded in the certificate)
- Private key



By default, the FVMI is protected inside an OS X–generated and maintained keychain named FileVaultMaster, located at `/Library/Keychains/FileVaultMaster.keychain`.

This particular keychain is one of the OS X–managed keychains that's not required to be listed in the keychain list (viewable via the Keychain Access utility). Users or administrators can, however, manually add this keychain to their keychain list if they're interested in viewing its contents. Simply double-click the keychain file or choose File > Add Keychain, navigate to the previously noted path, and click Open to add the keychain to the active list.

Viewing the visual contents in the keychain doesn't require any knowledge of the keychain protection credential. However, any manipulation of the keychain contents (such as deletion, exporting of the private key, or replacement of the identity) requires knowledge of the passphrase protecting the FileVaultMaster.keychain—originally entered by the user or administrator. Because use of the protected FVMI can provide access to a fully encrypted volume, it's critical to protect that identity from unauthorized access and use.

Self-Signed X.509 Root Certificate

The random X.509 identity is generated when an administrator uses either of two built-in services: "Set/Change Master Password" in Users & Groups preferences, or the CLI command `security`. The generated identity certificate is self-signed simply because FileVault 2 is designed to work for any user, regardless of environment. This means that anyone, even those without a full public key infrastructure (PKI), can take advantage of the power of FileVault 2. This architecture also enables an organization to use a designated identity from its own enterprise certificate authority (CA).

Regardless of the approach (provisioned directly on OS X or from an enterprise CA), OS X doesn't need to validate the trust path for the FileVault identity certificate to a trusted anchor. In the case of an OS X–generated identity, the certificate is a self-signed root CA certificate. For an enterprise CA–issued certificate, it could be a trusted root CA certificate, intermediate certificate, or leaf certificate. The existence of the identity with its properly provisioned public/private key pair with key size of 1K, 2K, or 4K, and the noted key usage is required, but validation of the trust path or revocation is not required.

The certificate has a few attributes that require specific values, and others can be set to any values:

Subject Name		
Common Name	FileVault Recovery Key	Required
Description	MySystem.local	Optional

Issuer Name		
Common Name	FileVault Recovery Key	Required
Description	MySystem.local	Optional
Signature Algorithm	SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)	Required

The Common Name is set to FileVault Recovery Key. This is required and must be exact for proper recognition and use by FileVault. Be extra careful that this attribute is properly set if the FVMI is provisioned from an organization's own CA.

Description is set to MySystem.local by default. It's generated from the computer name set in System Preferences > Sharing when the Master Password is set on the FVMK. No specific value is required for the Description field, but some organizations may want to set this attribute to the host's fully qualified domain name (FQDN), or user's name, for purposes of alternate identification within the certificate if the identity is issued from the organization's own CA.

This OS X–generated certificate is a self-signed root certificate. Both the subject name and issuer name are set to FileVault Recovery Key, which is the special case CA for use by FileVault on OS X. No certificate revocation checking is done on this certificate, so there's no need to be concerned with the certificate expiration date, nor does this certificate need to be explicitly set as trusted—it's inherently trusted by this process.

Public Key

The public key is the first half of the asymmetric (public/private) key pair, which is used to encrypt/wrap the KEK. As the name implies, it's publicly

accessible without the need for any protective measures. The public key is embedded in the X.509 certificate and doesn't need to be extracted or stored as a separate object.

Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Key Usage	Encrypt, Verify, Wrap
Key Size	1024, 2048, or 4096; default = 1024

The actual public key of the FileVault Recovery Key identity must be set to Encrypt, Verify, Wrap. If an enterprise chooses to replace the system-generated identity, the corresponding public key information would need to have at least the same extended key usage defined.

Extension—Key Usage (2.5.29.15)	
Critical	No
Usage	Digital Signature, Key Encipherment, Data Encipherment, Key Cert Sign

As noted about the public key of the FileVault Recovery Key identity, the extended key usage must be defined for all four functions noted: Digital Signature, Key Encipherment, Data Encipherment, Key Cert Sign.

Extension—Basic Constraints (2.5.29.19)	
Critical	Yes
Certificate Authority	Yes

Because this is a self-signed root certificate, it must be marked as critical and as a certificate authority.

Private Key

The private key is the second, critical half of the asymmetric (public/private) key pair, which is used to decrypt (unwrap) the key encryption key (KEK). It's to remain private and must be protected at all times or the FileVault 2 protection could be compromised. The ability to safely recover the encrypted volume and proper management of this FVMI are critical in keeping the system safe from unauthorized access.

Proper use and protection of this X.509-based identity can easily be misunderstood by individuals not entirely familiar with OS X controls and FileVault architecture. A strong understanding of the process that OS X uses to provide for the creation, use, and management of the FVMI is crucial.

Master Password

The Master Password simply refers to the password set on the FileVault Master Keychain (FileVaultMaster.keychain). An OS X keychain password is used to encrypt (wrap) the private data stored inside the keychain. The Master Password is required to unlock and decrypt (unwrap) the private key component of the FVMI. The corresponding certificate is public information and doesn't require protection.

This Master Password has no direct link to the contents of the FileVault Master Keychain or to the actual protection of the KEK, but rather simply to unlocking access to the private key if it's stored in the keychain. The password can be set to any valid password desired by the organization. It's frequently set to be unique to a single machine/user even if the deployed and protected FVMI is identical for all deployed systems.

Rotation of the Master Password

Resetting the FileVaultMaster.keychain password periodically might be of interest to an organization if it chooses to retain the complete FVMI within the keychain. In this case, the periodic Master Password rotation would help to mitigate the risk of compromising the identity. If an FVMI had been compromised and an unauthorized individual had access to both the certificate and private key together, the individual could unlock access to any protected volume without authorized user credentials if enabled for FileVault and originally associated with that FVMI.

Firmware

Standby Mode

All computers have firmware of some type—EFI, BIOS—to help in the discovery of hardware components and ultimately to properly bootstrap the computer using the desired OS instance. In the case of Apple hardware and the use of EFI, Apple stores relevant information within EFI to aid in the functionality of OS X. For example, the FileVault key is stored in EFI to transparently come out of standby mode.

Note: Standby causes kernel power management to automatically hibernate a machine after it has slept for a specified time period. This saves power while asleep. This setting defaults to ON for supported hardware. The setting standby is visible in pmset -g if the feature is supported on this machine. Standby only works if hibernation is turned on to hibernate mode 3 or 25.

Organizations especially sensitive to a high-attack environment, or potentially exposed to full device access when the device is in standby mode, should mitigate this risk by destroying the FileVault key in firmware. Doing so doesn't destroy the use of FileVault, but simply requires the user to enter the password in order for the system to come out of standby mode.

The destruction of the FileVault key when going to standby mode can be accomplished by setting a specific power management environment variable using the `pmset` command. Performing the following command on the targeted system interactively, or during the execution of a script for automation or deployments, sets the key for destruction:

```
# pmset destroyfvkeyonstandby 1
```

Setting the same variable to 0 causes a retention of the FileVault key when going into standby mode.

The full description of the `pmset` command `destroyfvkeyonstandby`, along with other available variable, can be seen by executing the following command:

```
# man pmset
```

The description for `destroyfvkeyonstandby` reads as follows:

```
destroyfvkeyonstandby - Destroy FileVault Key
when going to standby mode. By default
FileVault keys are retained even when system
goes to standby. If the keys are destroyed,
user will be prompted to enter the password
while coming out of standby mode.(value: 1 -
Destroy, 0 - Retain)
```

To determine the current state of `destroyfvkeyonstandby`, or any other environment variable, simply execute the following command:

```
# pmset -g
```

The result of this command is similar to the following:

```
bash-3.2# pmset -g
System-wide power settings:
 DestroyFVKeyOnStandby          0
Active Profiles:
Battery Power           -1
AC Power                -1*
Currently in use:
 standbydelay         4200
 standby              0
 womp                 1
 halfdim              1
 hibernatefile        /var/vm/sleepimage
 gpuswitch            2
 sms                  1
 networkoversleep     0
 disksleep            10
 sleep                0
 hibernatemode        3
 ttyskeepawake        1
 displaysleep         60
 acwake               0
 lidwake              1
```

Firmware Password

OS X systems support the use of a password for locking down the current firmware settings and preventing unintended modifications of firmware on a specific system. This is not for modifications to what's installed on the hard drive itself. This firmware password is most frequently used to prevent tightly controlled users from booting from an alternative system volume, or preventing use of other "catch keys" to alter the flow of the boot process (such as booting into single-user mode). This password also prevents unauthorized users from booting into the hidden Recovery partition if not allowed at an organization. A firmware password can also prevent Direct

Memory Access (DMA) via interfaces such as FireWire. Target Disk Mode requires DMA, so a firmware password also prevents its use on the system.

An authorized individual who knows the firmware password can hold down the Option key on the keyboard at boot time, enter the password when prompted, and select an alternate boot volume using the system's "Boot Picker" interface.

This action allows the individual to perform various modifications to the original boot volume (such as running a disk repair tool or even unlocking access to the encrypted volume for forensic purposes).

The firmware password stays with the host computer, so moving an encrypted drive from one system to another doesn't bring the firmware password with it. However, if an individual attempts to mount a volume from another computer using Target Disk Mode, the firmware password needs to be entered before mounting the volume from the target system.

Boot Camp

Boot Camp is Apple technology that supports the use of native execution of Microsoft Windows on supported Apple hardware. With every new Mac, users can install and run Windows at native speeds directly from a disk partition, using a built-in utility called Boot Camp. Setup is simple and safe for Mac files because it's on a completely separate partition from the OS X boot partition. With Windows installed using Boot Camp, users can boot up their Mac using either OS X or Windows.

Microsoft Windows and third-party FDE solutions for Windows don't have the capability to interpret and utilize a volume managed under CoreStorage. This prevents the use of Boot Camp and Windows on an OS X system using FileVault 2 FDE technology.

Instead, any organization running multiple operating systems at the same time (such as OS X and Windows on any Apple hardware) can use virtualization and install Windows using VMware or Parallels software.

For more information on Boot Camp and running Windows on Apple hardware, go to the OS X compatibility web page and Boot Camp support page:

- http://www.apple.com/macosx/what-is/compatibility.html
- http://www.apple.com/support/bootcamp/

Two-Factor Authentication

As discussed earlier, when using FileVault 2 (FDE), the initial authentication takes place as part of the EFI pre-boot authentication process. At this very early stage of the boot phase, none of the OS-reliant services are able to load because they're dependent on the OS running. This means that alternative authentication mechanisms other than password-based authentication aren't supported at this time.

Any support for additional two-factor authentication mechanisms, such as smart cards or one-time passwords (OTP), requires further development of those services in the highly restricted space and execution of EFI. If an organization needs to use smart cards for authenticating and unlocking access to encrypted storage, use of container-based Legacy FileVault should be examined more closely.

More information about Legacy FileVault and its support for smart cards can be found by searching http://www.apple.com/support.

Appendix B: FileVault 2 Process Flow

User Access Flow

1. User powers on device
2. EFI
   - 2.1. Loads authorized FDE users' information from boot volume
   - 2.2. EFI-based login displays authorized users by icon and name
   - 2.3. Password authentication required when user is selected
3. User
   - 3.1. Enters account password
4. FileVault
   - 4.1. Password converted to a key using PBKDF2
     - 4.1.1. PBKDF2—RSA Password-Based Key Derivation Function
   - 4.2. Password validated by attempting to unlock KEK keys found in user bundles retrieved from CoreStorage metadata
   - 4.3. Success means a successful unwrapping of a KEK key
5. OS kernel is loaded
6. Storage of boot parameters in AppleKeyStore for later kernel retrieval
   - 6.1. Unwrapped VEK using the obtained KEK
   - 6.2. Token identifying authenticated user
   - 6.3. User's password
7. Transfer control to kernel
8. Unlock volume
   - 8.1. Apple_CoreStorage partition and AppleKeyStore references
   - 8.2. VEK used to unlock Apple_CoreStorage partition (boot volume)
9. Operating System boot
   - 9.1. System discovers root volume as usual
   - 9.2. Login
     - 9.2.1. Password forwarding attempted with password user entered at pre-boot time
       - 9.2.1.1. Password forwarding enabled via an authentication mechanism entry `builtin:forward-login,privileged` referenced in Authorization Database (`/etc/authorization`) under the right: `system.login.console`
     - 9.2.2. Directory Service authenticates user/password
     - 9.2.3. User granted access and presented with personal desktop

Appendix C: Additional Resources

Related Knowledge Base Articles

A variety of Knowledge Base articles are available from the Apple Support website on the setup and use of FileVault 2. Visit support.apple.com to learn more about topics such as:

- OS X Lion: About FileVault 2
- OS X Lion: Using FileVault 2 and Lion Recovery
- OS X Recovery Disk Assistant v1.0
- MacBook Air: Recovering a lost EFI firmware password
- Mac OS X: How to start up in single-user or verbose mode

Related Web Page

- OS X Recovery restores your Mac with a few clicks

Security Configuration Guides

Apple provides additional best practices for enhancing Mac security with detailed guidance that results from years of collaboration with globally respected security organizations such as the National Security Agency (NSA), National Institute of Standards and Technology (NIST), and Defense Information Systems Agency (DISA). For the most current guidance on Apple products, visit www.apple.com/support/security/guides.

Training and Certification

Apple Authorized Training Centers offer a wide variety of IT training and certification opportunities for professionals interested in planning, maintaining, and integrating OS X, OS X Server, and other Apple solutions into their enterprise environments. Instructor-led courses offer demonstrations and lectures mixed with hands-on, real-world labs and exercises to deliver the most comprehensive training available for IT professionals. To learn more about Apple training and certification, visit http://training.apple.com/certification.

8/17/12