Windows XP Resource Kit: Using Encrypting File System
Published: November 03, 2005

Microsoft Windows XP Encrypting File System (EFS) enables users to encrypt individual files, folders, or entire data drives. Because EFS provides strong encryption through industry-standard algorithms and public key cryptography, encrypted files are confidential even if an attacker bypasses system security. EFS users can share encrypted files with other users on file shares and in Web folders. Many EFS features can be configured through Group Policy settings or command-line tools, facilitating enterprise management.

For information on how to obtain the Windows XP Professional Resource Kit in its entirety, please see http://www.microsoft.com/mspress/books/6795.asp.

Overview

Security features such as logon authentication or file permissions protect network resources from unauthorized access. However, anyone with physical access to a computer such as a stolen laptop can install a new operating system on that computer and bypass the existing operating system's security. In this way, sensitive data can be exposed. Encrypting sensitive files by means of EFS adds

another layer of security. When files are encrypted, their data is protected even if an attacker has full access to the computer's data storage.

Only authorized users and designated data recovery agents can decrypt encrypted files. Other system accounts that have permissions for a file—even the Take Ownership permission—cannot open the file without authorization. Even the administrator account cannot open the file if that account is not designated as a data recovery agent. If an unauthorized user tries to open an encrypted file, access is denied.

Benefits of EFS

EFS allows users to store confidential information about a computer when people who have physical access to your computer could otherwise compromise that information, intentionally or unintentionally. EFS is especially useful for securing sensitive data on portable computers or on computers shared by several users. Both kinds of systems are susceptible to attack by techniques that circumvent the restrictions of access control lists (ACLs). In a shared system, an attacker can gain access by starting up a different operating system. An attacker can also steal a computer, remove the hard drives, place the drives in another system, and gain access to the stored files. Files encrypted by EFS, however, appear as unintelligible characters when the attacker does not have the decryption key.

Because EFS is tightly integrated with NTFS, file encryption and decryption are transparent. When users open a file, it is decrypted by EFS as data is read from disk. When they save the file, EFS encrypts the data as it is written to disk. Authorized users might not even realize that the files are encrypted because they can work with the files as they normally do.

In its default configuration, EFS enables users to start encrypting files from My Computer with no administrative effort. From the user's point of view, encrypting a file is simply a matter of setting a file attribute. The encryption attribute can also be set for a file folder. This means that any file created in or added to the folder is automatically encrypted.

How EFS Works

The following steps explain how EFS works.

1. EFS uses a public-private key pair and a per-file encryption key to encrypt and decrypt data. When a user encrypts a file, EFS generates a file encryption key (FEK) to encrypt the data. The FEK is encrypted with the user's public key, and the encrypted FEK is then stored with the file.
2. Files can be marked for encryption in a variety of ways. The user can set the encryption attribute for a file by using Advanced Properties for the file in My Computer, storing the file in a file folder set for encryption, or by using the Cipher.exe command-line utility. EFS can also be configured so that users can encrypt or decrypt a file from the shortcut menu accessed by right-clicking the file.
3. To decrypt files, the user opens the file, removes the encryption attribute, or decrypts the file by using the cipher command. EFS decrypts the FEK by using the user's private key, and then decrypts the data by using the FEK.

New for Windows XP Professional

EFS in Windows XP Professional includes the following new features:

- Additional users can be authorized to access encrypted files.
- Offline Files can be encrypted.
- Data Recovery Agents are recommended but optional.
- The triple-DES (3DES) encryption algorithm can be used to replace DESX.
- A password reset disk can be used to safely reset a user's password.
- Encrypted files can be stored in Web folders.

In addition, EFS in Windows XP Service Pack 1 includes the following new features:

- By default, the Advanced Encryption Standard (AES) algorithm is now used for encrypting files with EFS. (See article 329741, "EFS Files Appear

Corrupted When You Open Them," in the Microsoft Knowledge Base at http://support.microsoft.com for more information.)

- Changes in the RC2 cipher text algorithm make it more secure. (See article 841715, "You cannot decrypt data on Windows XP SP1 or later versions," in the Microsoft Knowledge Base at http://support.microsoft.com for more information.)

Finally, the Cipher.exe utility in Windows XP Service Pack 2 has been enhanced to include the /x switch supported by Windows Server™ 2003. This switch can be used to back up an EFS certificate and its associated keys to a file from the command line. (For more information, see article 827014, "New Functionality Is Available for Cipher.exe in Windows 2000 and Windows XP," in the Microsoft Knowledge Base at http://support.microsoft.com.)

Configuring EFS for Your Environment

EFS is enabled by default. Users can encrypt files if they have permission to modify the files. Because EFS relies on a public key to encrypt files, users need a public-private key pair and a public key certificate for encryption. Because EFS can use self-signed certificates, however, EFS does not require administrative effort before use.

Note The use of self-signed certificates for EFS is not recommended in a domain environment. Configuring certification authorities to deliver EFS certificates to users as part of your public key infrastructure simplifies the manageability of recovery agents.

If EFS is not appropriate in your environment or you have files that you do not want encrypted, you can disable EFS in various ways. There are also a number of ways in which you can configure EFS to meet the specific needs of your organization.

To use EFS, all users must have EFS certificates. If you do not currently have a public key infrastructure (PKI), you can use self-signed certificates. If you have certification authorities, however, you might want to configure them to provide EFS certificates.

You will also need to consider a disaster recovery plan if you use EFS on your system.

Components of EFS

EFS consists primarily of the following operating system components: the EFS service, the EFS driver, the EFS File System Run-Time Library (FSRTL), and an application programming interface (API). Like many other security services, EFS uses the Microsoft Cryptographic Application Programming Interface to obtain services from a cryptographic service provider such as the RSA Base Provider that is included with Windows XP Professional. Figure18-1 shows the architecture of EFS.

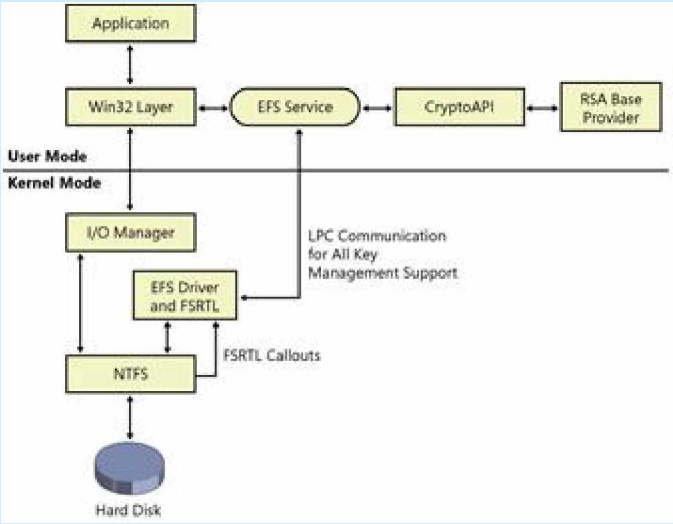

Figure 18-1 EFS architecture

EFS Service

The EFS service is part of the security subsystem. It uses the existing local procedure call (LPC) communication port between the Local Security Authority (LSA) and the kernel-mode security reference monitor to communicate with the EFS driver. In user mode, it interfaces with CryptoAPI to obtain file encryption keys and to generate data decryption fields (DDFs) and data recovery fields (DRFs). The EFS service also provides support for Win32 APIs.

The EFS service calls CryptoAPI to acquire the file encryption key (FEK) for a data file and then to encode the FEK, thus producing the DDF. The EFS service also returns the FEK, DRF, and DDF by way of the FSRTL to the EFS driver.

EFS Driver

EFS is tightly integrated with NTFS. The EFS driver is essentially a file system filter driver logically layered on top of NTFS. It communicates with the EFS service to request file encryption keys, DDFs, DRFs, and other key management services. It passes this information to the EFS FSRTL to perform various file system operations (open, read, write, and append) transparently.

CryptoAPI

CryptoAPI provides services that enable application developers to add cryptography to their Win32 applications. CryptoAPI consists of a set of functions that allow applications to encrypt or digitally sign data in a flexible manner, while providing protection for private key data. Applications can use the functions in CryptoAPI without knowing anything about the underlying implementation.

CryptoAPI provides the underlying security services for secure channels and code signing. CryptoAPI supports public key and symmetric-key operations such as key generation, key management and secure storage, key exchange, encryption, decryption, hashing, digital signatures, and verification of signatures. Developers can use certificates with these public key operations and perform the necessary encapsulations and encoding to apply certificates within their applications.

EFS uses CryptoAPI for all of its cryptographic operations.

Cryptographic Service Provider

By default, EFS uses the DESX algorithm, a variation of the U.S. government's Data Encryption Standard (DES) algorithm, for file encryption. The public-private key pairs for EFS users and recovery agent accounts are obtained from the Microsoft base cryptographic service provider (CSP), also called the RSA base provider. This CSP is included with Windows XP Professional and is approved for general export worldwide. The Microsoft-enhanced CSP can also be used for EFS.

3DES algorithm support

Windows XP Professional can be configured to use the triple-DES (3DES) algorithm instead of DESX. 3DES, which is compliant with Federal Information Processing Standards (FIPS 140-1 Level 1), offers significantly stronger encryption using a 128-bit or 168-bit key.

3DES is enabled through a Group Policy setting.

Note When 3DES is enabled, it is used as the encryption algorithm for IP Security as well as for EFS. For more information about configuring 3DES support, see "Enabling 3DES" later in this chapter.

When 3DES is enabled, all new encryptions are completed by using 3DES. Note that DESX and 3DES are always available for decryption, regardless of the encryption policy.

Note As of Service Pack 1 for Windows XP, the Advanced Encryption Standard (AES) algorithm is used by default for encrypting files with EFS. For more information, see article 329741, "EFS Files Appear Corrupted When You Open Them," in the Microsoft Knowledge Base at http://support.microsoft.com.

Data Protection API

The Data Protection API (DPAPI) is a set of function calls that provide data protection services to user and system processes. Applications either pass plaintext data to DPAPI and receive protected data back, or they pass the protected data to DPAPI and receive plaintext data back. For example, after a CSP generates keys for certificates, it calls CryptProtectData(), one of the primary functions of DPAPI, to protect those keys. When the keys are needed, DPAPI decrypts them.

EFS FSRTL

The EFS FSRTL is a module within the EFS driver that implements NTFS callouts to handle various file system operations such as reads, writes, and opens on encrypted files and directories, and operations to encrypt, decrypt, and recover file data when it is written to or read from disk. The EFS driver and FSRTL are implemented as a single component. However, they never

communicate directly. They use the NTFS file control callout mechanism to pass messages to each other. This ensures that NTFS participates in all file operations. The operations implemented by using the file control mechanisms include writing the EFS attribute data (DDF and DRF) as file attributes and communicating the FEK computed in the EFS service to FSRTL so that it can be set up in the open file context. This file context is then used for transparent encryption and decryption on writes of file data to disk and reads of file data from disk.

Win32 API

EFS provides an API set to expose its features. This API provides a programming interface for operations such as encrypting plaintext files, decrypting or recovering ciphertext files, and importing and exporting encrypted files (without decrypting them first). The API is remoted to support remote encryption, decryption, backup, and restore operations. The API is supported in a standard system DLL, Advapi32.dll.

Encrypting and Decrypting by Using EFS

Encryption and decryption are the primary tasks of EFS. Several encryption and decryption options are available to users. Users can encrypt and decrypt files using My Computer, the `cipher` command, or the shortcut menu accessed by right-clicking a file or folder.

EFS also allows users to encrypt offline files. Additionally, EFS provides several options for users to determine the encryption status of files and folders.

What Can Be Encrypted

Individual files and file folders (or sub-folders) on NTFS volumes can be encrypted. Although it is common to refer to file folders with the encryption attribute set as “encrypted,” the folder itself is not encrypted. When encryption is set for a folder, EFS automatically encrypts all new files created in the folder and all files copied or moved into the folder by using My Computer. Offline Files can also be encrypted.

Note When offline files are encrypted, the entire offline files database is encrypted rather than individual files. Individual files do not display the encryption attribute. The database is encrypted using the system's startup key.

System files and any files in the systemroot folder or its subfolders cannot be encrypted. No files or directories in a roaming user profile can be encrypted. A file cannot be both compressed and encrypted. Being compressed does not prevent encryption, but when the file is encrypted, it is uncompressed.

How Files Are Encrypted

EFS uses a combination of public key and symmetric key encryption to ensure that files are protected from all but the most computationally infeasible methods of attack. Public key encryption algorithms use asymmetric keys for encryption and decryption, which means that different keys are used to encrypt and decrypt the same data. Public key encryption involves the use of a private key (which is held only by its owner) and a public key (which is publicly available on the network). Information that is encrypted by using the public key can be decrypted only by using the corresponding private key. The two keys together are called a key pair or a key set.

Asymmetric cryptography, however, requires a significant amount of processing time for its mathematical operations. Public key operations are often used as part of initial key exchange or key protection operations. As soon as possible, cryptographic services change from public key to symmetric operations, in which the same key is used for both encryption and decryption. Compared with public key operations, symmetric encryption is commonly 100 to 1000 times faster.

EFS follows the industry standard cryptographic procedure of key encipherment. Data is encrypted using a symmetric file encryption key (FEK) for speed and then the FEK is secured asymmetrically for maximum security. When a user requests that a file be encrypted, EFS uses a uniquely generated FEK to encrypt a file and then encrypts the FEK by using the public key taken from the user's public key certificate. The encrypted FEK is stored in a file header. When a user requests decryption, EFS decrypts the FEK using the user's private key, and then uses the FEK to decrypt the file.

Structure of an Encrypted File

An encrypted file contains encrypted data and a header with fields to store copies of the encrypted FEK for authorized users and designated data recovery agents (DRA). For more information about DRAs, see “Data Recovery and Data Recovery Agents” later in this chapter.

The structure for an encrypted file is shown in Figure 18-2.

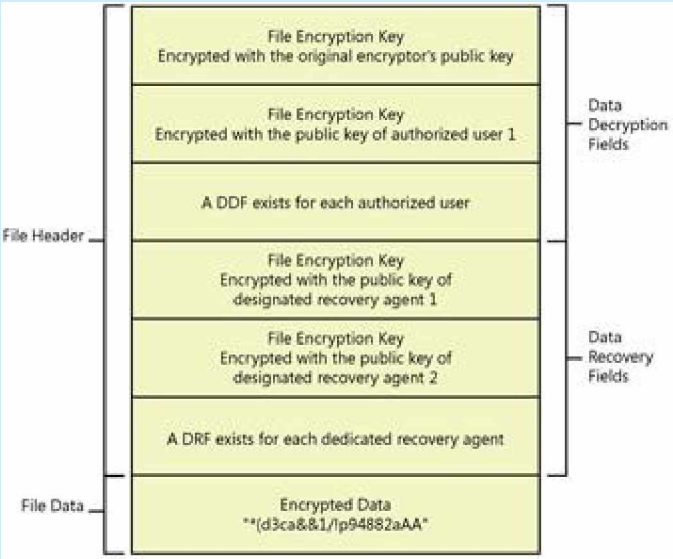

Figure 18-2 Structure of an encrypted data file

Data decryption field

An encrypted file contains a minimum of one stored FEK, the FEK encrypted by using the initial encryptor’s public key. The storage field for this encrypted FEK is called the *data decryption field* (DDF). Additionally, if an EFS-encrypted file is shared, a copy of the FEK is encrypted by using the newly authorized user’s public key, and the encrypted FEK is stored in another DDF.

For more information about sharing encrypted files, see “Authorizing Multiuser Access to Encrypted Files” later in this chapter.

Data Recovery Field

If a computer’s effective security policy designates one or more data recovery agent (DRA), copies of the FEK are encrypted for each DRA using each DRA’s

public key and stored in another file header field called the *data recovery field* (DRF).

The Encryption Process

When a user encrypts an existing file, the following process occurs:

1. The EFS service opens the file for exclusive access.
2. All data streams in the file are copied to a plaintext temporary file in the system's temporary directory.
3. An FEK is randomly generated and used to encrypt the file by using DESX or 3DES, depending on the effective security policy.
4. A DDF is created to contain the FEK encrypted by using the user's public key. EFS automatically obtains the user's public key from the user's X.509 version 3 file encryption certificate.
5. If a recovery agent has been designated through Group Policy, a DRF is created to contain the FEK encrypted by using RSA and the recovery agent's public key. For more information about using Group Policy to configure data recovery agents, see "Configuring Data Recovery Policy in a Stand-Alone Environment" later in this chapter.

   EFS automatically obtains the recovery agent's public key for file recovery from the recovery agent's X.509 version 3 certificate, which is stored in the EFS recovery policy. If there are multiple recovery agents, a copy of the FEK is encrypted by using each agent's public key, and a DRF is created to store each encrypted FEK.

   Note The file recovery property in the certificate is an example of an enhanced key usage (EKU) field. An EKU extension and extended property specify and limit the valid uses of a certificate. File Recovery is one of the EKU fields defined by Microsoft as part of the Microsoft public key infrastructure (PKI).
6. EFS writes the encrypted data, along with the DDF and the DRF, back to the file. Because symmetric encryption does not add additional data, file size increase is minimal after encryption. The metadata, consisting

primarily of encrypted FEKs, is usually less than one kilobyte. File size in bytes before and after encryption is normally reported to be the same.

7. The plaintext temporary file is deleted.

   Note Data from deleted files might not be erased when the file is deleted. The cipher /w command can be used to remove data from available unused disk space on the entire volume. For more information about the cipher command, see Windows XP Professional Help and Support Center, or use the cipher /? command at a command prompt. In addition, see article 814599, "How to Use Cipher.exe to Overwrite Deleted Data in Windows Server 2003," in the Microsoft Knowledge Base at http://support.microsoft.com.

When a user saves a file to a folder that has been configured for encryption, the process is similar except that no temporary file is created.

Figure 18-3 illustrates the process of obtaining an FEK to encrypt the data and obtaining public keys to encrypt the FEK, and it shows the structure of the encrypted file.

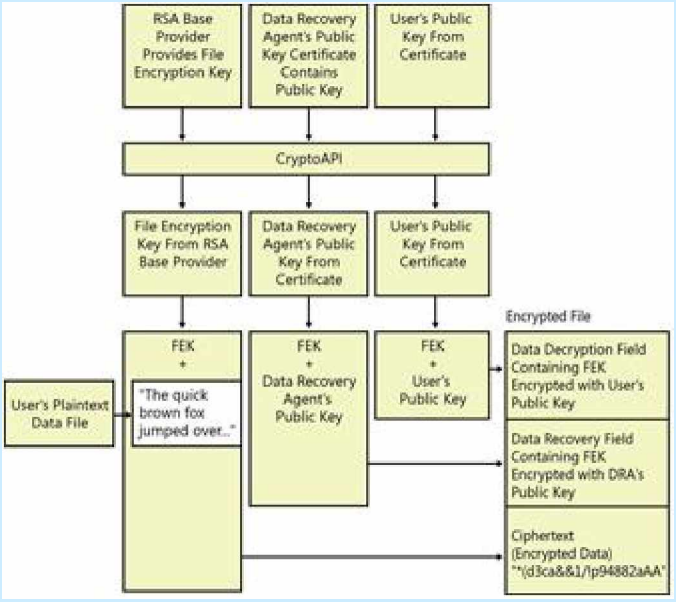

Figure 18-3 Encrypting a file with EFS

The Decryption Process

When an application accesses an encrypted file, decryption proceeds as follows:

1. NTFS recognizes that the file is encrypted and sends a request to the EFS driver.
2. The EFS driver retrieves the DDF and passes it to the EFS service.
3. The EFS service retrieves the user's private key from the user's profile and uses it to decrypt the DDF and obtain the FEK.
4. The EFS service passes the FEK back to the EFS driver.
5. The EFS driver uses the FEK to decrypt sections of the file as needed for the application.

   Note When an application opens a file, only those sections of the file that the application is using are decrypted because EFS uses cipher block chaining. The behavior is different if the user removes the encryption attribute from the file. In this case, the entire file is decrypted and rewritten as plaintext.
6. The EFS driver returns the decrypted data to NTFS, which then sends the data to the requesting application.

Figure18-4 illustrates the process of obtaining the user's private key from the user's profile, using it to decrypt the FEK, and using the FEK to decrypt the data for a user.

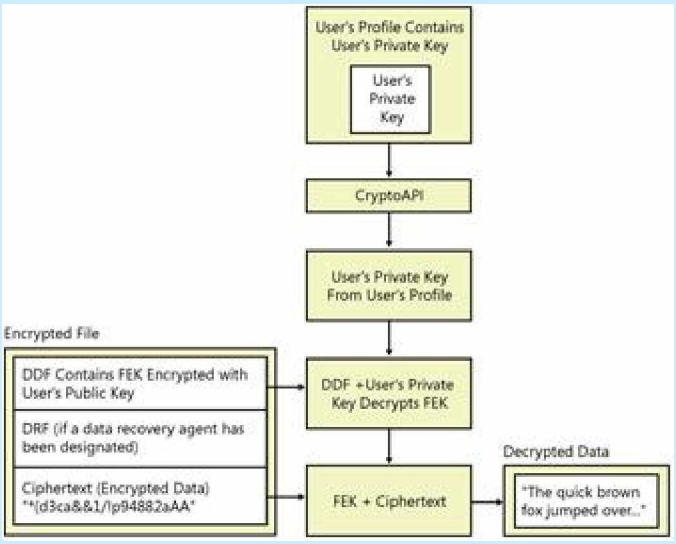

Figure 18-4 Decrypting a file for a user

Working with Encryption and Decryption

When encrypting files, it is best to turn on encryption for folders rather than to encrypt files individually. By using encrypted folders, you do not have to encrypt each file when you save it. This also ensures that any temporary or backup files that the application creates remain encrypted during and after editing, provided that the application does its editing in the same folder. When files or folders are encrypted or decrypted, the date and time stamps on the files and folders are updated to the current date and time.

Note Applications do not always use the same folder for temporary files or backup files. Microsoft Word, for example, uses the folder where the encrypted file is located for temporary and backup files but can be configured to use alternate folders. Also, if EFS is appropriate for a specific user, it is a good practice to encrypt the user's My Documents folder because many applications use this folder as the default location to save files. Remember, though, that no files or directories in roaming profiles can be encrypted.

Encrypting and Decrypting Files and Folders by Using My Computer

Applying encryption to a folder by using My Computer is simply a matter of assigning an attribute.

To apply encryption to a folder by Using My Computer

1. In My Computer, select the folder to encrypt.
2. Right-click the folder and click Properties.
3. On the General tab, click the Advanced button.
4. Select the Encrypt contents to secure data check box, and then click OK.

   When you click OK, if the folder contains unencrypted files or subfolders, another dialog box appears to ask you whether you want to apply the changes to just the folder or to the folder, its subfolders, and all files.

Table 18-1 shows the results of selecting the Apply changes to this folder only option.

Table 18-1 Results of Selecting the Apply Changes to This Folder Only Option

File Description	Encryption Status
Already stored in the folder and its subfolders	Unchanged. Files remain either encrypted or unencrypted.
Created in or copied to the folder by you later	File is encrypted and FEK is encrypted by using your public key.
Created in or copied to the folder by another user later	File is encrypted and FEK is encrypted by using the other user's public key.
Created in or copied to subfolders later	Unchanged.
Moved to the folder or subfolders later	Unchanged.

Table 18-2 shows the results of choosing the Apply changes to this folder, subfolders, and files option.

Table 18-2 Results of Selecting the Apply Changes to This Folder, Subfolders, and Files Option

File Description	Encryption Status
Already in the folder and its subfolders	If you have Write permission, file is encrypted and FEK is encrypted by using your public key; otherwise, files are unchanged.
Later created in or copied to the folder or subfolders by you	File is encrypted and FEK is encrypted by using your public key.
Later created in or copied to the folder or subfolders by another user	File is encrypted and FEK is encrypted by using the other user's public key.
Later moved to the folder or subfolders	Moving unencrypted files into an encrypted folder will automatically encrypt those files in the new folder.

With either choice, the folder's list of files remains in plaintext and you can enumerate files as usual.

Caution EFS lets you encrypt files you do not own, provided that you have Write Attributes, Create Files/Write Data, and List Folder/Read Data permissions for the files. If you select Apply changes to this folder, subfolders, and files in folders where other users also store files, no one but you will be able to decrypt the files. If this occurs, you can recover the files by reversing the process. Select the folder, and clear the Encrypt contents to secure data check box. On a shared computer, it is better to encrypt folders such as My Documents for each user. If users have roaming profiles, the My Documents folder cannot be encrypted because no files in a roaming profile can be encrypted. In this case, it is better to create individual data folders outside the user profile for each user.

You can turn on EFS for an individual file by using My Computer in the same way that you apply encryption to a folder. However, when you encrypt a single file, the following warning appears: "You have chosen to encrypt a file that is not in an encrypted folder. The file can become decrypted when it is modified." If you select the Always encrypt only the file check box, the warning no longer appears and EFS encrypts only the file that you select.

To decrypt a file or folder by Using My Computer

1. Right-click the file or folder, and then click Properties.
2. Click Advanced, and then clear the Encrypt contents to secure data check box.

   This causes EFS to decrypt the selected folder and mark it as unencrypted. When you apply your choice, you also have the option to decrypt all files and subfolders in the folder.

   Note In Windows XP and later, anyone who has permissions to change the attributes of a folder can clear the Encrypt contents to secure data check box in the folder properties. For more information, see article 821737, “A User Who Has Permissions to Change the Folder Attributes Can Now Change the Folder Encryption Attribute,” in the Microsoft Knowledge Base at http://support.microsoft.com.

Encrypting and Decrypting Files and Folders by Using the Cipher Command

You can encrypt and decrypt folders or files by using the Cipher.exe command-line utility. Table 18-3 lists some of the parameters for the tasks that you can perform by using the cipher command. For a full list of parameters available, type cipher /? at a command prompt.

Note You can use wildcard characters and multiple parameters with the cipher command. A space is required between multiple parameters.

Table 18-3 Tasks and Parameters for the Cipher Command

Task	Parameter(s)
Display the encryption status of files and folders.	Use the cipher command with no parameters or with the name of a specific file or folder.
Set the encryption attribute for folders in the current directory.	/e
Encrypt files in the current directory.	/e /a
Remove the encryption attribute from folders in the current	/d

directory.	
Decrypt files in the current directory.	/d /a
Display all the options available with cipher.	/?

In the following example, a folder called “project docs” is encrypted and decrypted. Cipher “project docs” displays the status as “U” or unencrypted. Cipher /e “project docs” encrypts the folder. Cipher /d decrypts the folder.

X:\>cipher "project docs" Listing X:\ New files added to this directory will not be encrypted. U Project docs X:\>cipher /e "project docs" Encrypting directories in X:\ Project docs [OK] 1 directorie(s) within 1 directorie(s) were encrypted. X:\>cipher "project docs" Listing X:\ New files added to this directory will not be encrypted. E Project docs X:\>cipher /d "project docs" Decrypting directories in X:\ Project docs [OK] 1 directorie(s) within 1 directorie(s) were decrypted. X:\>cipher "project docs" Listing X:\ New files added to this directory will not be encrypted. U Project docs

Enabling EFS Options on the Shortcut Menu

Some organizations might choose to enable EFS on the shortcut menu. Encrypt and Decrypt are then available when a user right-clicks a file or folder in My Computer.

To enable EFS options on the shortcut menu by editing the registry

1. In the Run dialog box, type regedit.exe.
2. Navigate to the subkey HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \Windows\CurrentVersion\Explorer\Advanced.
3. On the Edit menu, point to New, and then click DWORD Value.
4. Enter EncryptionContextMenu for the value name, and 1 for the value data.

This change takes effect the next time My Computer is opened. When the user right-clicks a file or folder on an NTFS volume, the option to encrypt or decrypt appears on the shortcut menu.

Caution Do not edit the registry unless you have no alternative. The Registry Editor bypasses standard safeguards, allowing settings that can damage your system or even require you to reinstall Windows. If you must edit the registry, back it up first.

Encrypting Offline Files

Microsoft Windows 2000 introduced client-side caching functionality, now called Offline Files, which is an IntelliMirror management technology that allows network users to access files on network shares even when the client computer is disconnected from the network. When disconnected from the network, mobile users can still browse, read, and edit files by using the same UNC path that is used on the network because the files have been cached on the client computer. When the user later connects to the server, the system reconciles the changes with the server. The Windows XP Professional client can use EFS to encrypt offline files and folders. This feature is especially attractive for traveling professionals who need to work offline periodically and maintain data security.

To encrypt offline files

1. In My Computer, on the Tools menu, click Folder Options.
2. On the Offline Files tab, select Enable Offline Files and Encrypt offline files to secure data, and click OK.

Offline files will now be encrypted when cached locally, even if they were not encrypted in the network folder.

Determining Encryption Status by Using My Computer

Because encryption is an attribute of a file or folder, it is possible to determine whether a file or folder is already encrypted by examining its attributes. You can open the Advanced Properties sheet for the file or folder and see that the Encrypt contents to secure data check box is selected. You can also add the Attributes column to the Details view. When you do this, any file with an E

attribute is encrypted and any folder with an E attribute has the encryption attribute set.

Both encrypted and compressed files can be displayed with alternate colors in My Computer. Encrypted files are green, and compressed files are blue. Files and folders can either be encrypted or compressed, but they cannot be both encrypted and compressed.

To display encrypted files in an alternate color

1. In My Computer, on the Tools menu, click Folder Options.
2. On the View tab, select the Show encrypted or compressed NTFS files in color check box, and click OK. All encrypted file and folder names are displayed in green.

Determining Encryption Status Using the Cipher Command

Use the cipher command with no parameters or with a file or folder name to display encryption status. In the following example, the encryption status of files and folders in the X:\Project docs directory is displayed. Cipher is executed without any parameters. An “E” in the left column means that the file or folder is encrypted, and a “U” means that it is unencrypted.

```
X:\Project docs>cipher  Listing X:\Project docs\ New files added to this directory will be encrypted. E cipher E plan2.txt E plan3.txt E plan4.txt E plan5.txt E plan6.txt E plan7.txt E plan8.txt E secretplan.txt
```

Remote EFS Operations on File Shares and Web Folders

Users can encrypt and decrypt files that are stored on network file shares or on Web Distributed Authoring and Versioning (WebDAV) Web folders. Web folders have many advantages compared to file shares, and Microsoft recommends the use of Web folders whenever possible for remote storage of encrypted files. Web folders require less administrative effort and are more secure than file shares. Web folders can also securely store and deliver encrypted files over the Internet by using standard HTTP file transfers. Using file shares for remote EFS operations requires a Windows 2000 or later domain environment because EFS

must impersonate the user by using Kerberos delegation to encrypt or decrypt files for the user.

The primary difference between remote EFS operations on files stored on file shares and files stored on Web folders is where the operations occur. When files are stored on file shares, all EFS operations occur on the computer on which the files are stored. For example, if a user connects to a network file share and chooses to open a file that he or she previously encrypted, the file is decrypted on the computer on which the file is stored and then transmitted in plaintext over the network to the user's computer. When files are stored on Web folders, all EFS operations occur on the user's local computer. For example, if a user connects to a Web folder and chooses to open a file that he or she previously encrypted, the file remains encrypted during transmission to the user's computer and is decrypted by EFS on the user's computer. This difference in where EFS operations occur also explains why file shares require more administrative configuration than Web folders.

Remote EFS Operations in a File Share Environment

Remote EFS operations on files stored on network file shares are possible in Windows 2000 or later domain environments only. Domain users can remotely encrypt or decrypt files, but this capability is not enabled by default. The following are requirements for successful remote EFS operations in a file share environment:

1. The files to be encrypted must be available to the user through a network share. Normal share-level security applies.
2. The user must have Write or Modify permissions to encrypt or decrypt a file.
3. The user must have either a local profile on the computer where EFS operations will occur or a roaming profile. If the user does not have a local profile on the remote computer or a roaming profile, EFS creates a local profile for the user on the remote computer.

   If the remote computer is a server in a cluster, the user must have a roaming profile.

4. To encrypt a file, the user must have a valid EFS certificate. If EFS cannot locate a pre-existing certificate, EFS contacts a trusted enterprise certification authority for a certificate. If no trusted enterprise certification authorities are known, a self-signed certificate is created and used. The certificate and keys are stored in the user's profile on the remote computer or in the user's roaming profile if available.

   Note To verify a certificate's authenticity, a certification authority signs the certificates that it issues with its private key. EFS creates and uses a self-signed certificate if no file encryption certificate is available from a certification authority. A self-signed certificate indicates that the issuer and subject in the certificate are identical, and that no certification authority has signed the certificate.

5. To decrypt a file, the user's profile must contain the private key associated with the public key used to encrypt the file encryption key (FEK).

6. EFS must impersonate the user to obtain access to the necessary public or private key. This requires the following:

   1. The computer must be a domain member in a domain that uses Kerberos authentication because impersonation relies on Kerberos authentication and delegation.

   2. The computer must be trusted for delegation.

   3. The user must be logged on with a domain account that can be delegated.

      Note Use the Active Directory Users and Computers snap-in to configure delegation options for both users and computers. To trust a computer for delegation, open the computer's Properties sheet and select Trusted for delegation. To allow a user account to be delegated, open the user's Properties sheet. On the Account tab, under Account Options, clear the The account is sensitive and cannot be delegated check box. Do not select The account is trusted for delegation. This property is not used with EFS.

Remote decryption is a potential security risk because files are decrypted prior to transmission and are transmitted unencrypted. EFS decrypts the file on the computer that stores the encrypted file, and the data is then transmitted over the network in plaintext. Organizations need to consider whether this level of risk is acceptable. You can greatly reduce or eliminate this risk by enabling IP Security to use Encapsulating Security Payload (ESP)—which will encrypt transmitted data, enabling another network layer security protocol—or by using Web folders. For more information about configuring IP Security, see "Internet Protocol Security" in the *TCP/IP Core Networking Guide* of the *Microsoft Windows 2000 Server Resource Kit.*

Remote Encryption on File Shares

Encrypting files and then encrypting the user's FEK by using the user's public key requires access to the user's EFS certificate. If the user is encrypting a file stored on the computer at which he or she is logged on, EFS uses the existing EFS certificate, obtains a certificate from an enterprise certification authority (CA), or creates a self-signed certificate.

If the user is encrypting a file on a remote computer, EFS must first impersonate the user by using Kerberos delegation. If this impersonation is successful, EFS determines whether the user has a roaming profile and/or a local profile. If the user has a roaming profile, EFS loads it. If not, EFS loads the local profile, if one is available. If no profile can be located, EFS creates one for the user.

EFS looks for a valid EFS certificate and its associated private key in the user's profile. If no certificate exists, or if the profile contains a certificate but not the associated private key, EFS attempts to locate a trusted enterprise CA and obtain an EFS certificate for the user. As part of this attempt, EFS generates a public-private key pair. If a trusted enterprise CA can be contacted, EFS requests a certificate. If a certificate cannot be obtained from a CA, EFS self-signs a certificate.

After EFS has a user profile and a valid certificate, an FEK is created and used to encrypt the file's data. The public key is then used to encrypt the FEK, and the encrypted FEK is stored in the file header.

If EFS creates a public-private key pair, both keys are stored in the user's profile on the remote computer.

Remote Decryption on File Shares

To decrypt a file, EFS must first obtain the user's private key. More specifically, EFS needs the private key associated with the public key used to encrypt the file's FEK. EFS uses the private key to decrypt the FEK and then uses the FEK to decrypt the file. The private key is stored in the user's profile. Before decrypting the file, EFS must:

1. Locate the user's profile. The user is not currently logged on at the remote computer where the decrypting is taking place, so EFS impersonates the user. The user might have a local profile or a roaming profile.
2. Find the appropriate private key. When a profile is located, EFS checks any private keys contained in the profile for a match with the public key that encrypted the FEK.

   Note Public-private key pairs share a "thumbprint" value, a unique hash value stored with each key.
3. Decrypt the FEK. If the user profile contains the correct private key, EFS uses it to decrypt the FEK.

When EFS decrypts a file on the computer at which the user is logged on, the user is already accessing his or her user profile. When the user attempts to decrypt a remote file, however, EFS needs to impersonate the user to get access to the user's profile, in which the user's private keys are stored. This requires the computer to be trusted for delegation.

In a remote decryption scenario, then, EFS determines whether the computer has been trusted for delegation. If not, the decryption process fails.

The user account must also not be designated as a sensitive account that cannot be delegated. Domain administrator accounts, for example, are flagged as nonforwardable (identity cannot be delegated). Any account that cannot be delegated cannot use EFS remotely.

If the computer is trusted for delegation and the user account that EFS needs to impersonate can be delegated, EFS can next locate the user's profile. The process is similar to that used after a new key pair has been generated and the private key needs storage. EFS looks for a local profile and a roaming profile, and it uses the roaming profile if it exists. If the user does not have a local or a roaming profile, the decryption process fails. EFS cannot create a user profile in this situation because it needs the existing private key (and thus the profile in which it is stored) to decrypt the FEK.

If a user profile is located, EFS looks for a private key to match the public key used to encrypt the FEK. If found, the private key is used to decrypt the FEK and file decryption can begin. If the private key is not found, the decryption process fails.

When the FEK is decrypted and used to decrypt the file, the data is ready to be transmitted in plaintext across the network.

Note that an attacker can use network monitoring software to access the plaintext data as it is transmitted over the network. You can prevent this by using IP Security with ESP and encryption to secure data as it is transmitted, or by storing encrypted files on Web folders.

Local and Remote File Operations in a File Share Environment

Encrypted files and folders can be renamed, copied, or moved. Renaming an encrypted file or folder either locally or remotely does not cause decryption. However, moving or copying a file or folder can result in decryption. The effects of moving or copying encrypted files and folders vary according to whether the files or folders are moved or copied locally or remotely.

For more information about renaming, copying, or moving encrypted files and folders, see Windows XP Professional Help and Support Center.

Local file operations and encrypted files

Encrypted files or folders retain their encryption after being either copied or moved, either by using My Computer or by using command-line tools, to local volumes, provided that the target volume uses the version of NTFS used in

Windows 2000 or later. Otherwise, encrypted files are stored as plaintext and encrypted folders lose the encryption attribute.

Note Most floppy disk drives are FAT volumes, so encryption is lost when files are copied to disk unless the files are backed up by using the Backup tool before they are copied. Encrypted files that are copied or moved to servers or workstations running Microsoft Windows NT 4.0 also lose their encryption.

Local file operations and plaintext files

When plaintext files are copied or moved in My Computer to an encrypted folder on a local NTFS volume, they are encrypted. Plaintext files are also encrypted if they are copied to an encrypted folder on a local volume by using the copy or xcopy commands.

The move command produces different results if the plaintext file is moved to an encrypted folder on the same volume or on a different local volume. If the move command is used to move a plaintext file to an encrypted folder on the same volume, the file remains in plaintext. This is because the move command simply renames the file, unless the file is moved to a different volume. If the file is moved to an encrypted folder on a different local volume, it is encrypted.

Renaming plaintext files that are in encrypted folders produces different results at the command line and in My Computer. If a plaintext file in an encrypted folder is renamed in My Computer, the resulting file is encrypted. If the file is renamed by using the ren command, it remains in plaintext.

Remote file operations

When encrypted files or folders are copied or moved to or from a network file share on a remote computer, the files are decrypted locally, transmitted in plaintext, and then re-encrypted on the target volume if possible. Encrypted files transmitted to or from Web folders remain encrypted during transmission. With remote file operations, EFS must impersonate the encryptor, so the encryptor's account must be configured to enable delegation, and the remote computer must be trusted for delegation. EFS makes use of either an existing EFS certificate for the encryptor if one is available, a certificate obtained from a trusted enterprise CA, or a self-signed certificate.

Encrypted files or folders retain their encryption after being either copied or moved from a local computer to a remote computer, provided that the remote computer is trusted for delegation and the target volume uses the version of NTFS used in Windows 2000 or later. A moved or copied file is re-encrypted on the target volume, so it has a new file encryption key (FEK), and the FEK is encrypted by using the user's public key if it is available or by using a new public key EFS generates if the profile is unavailable. In the latter case, a new user profile is generated for the user on the remote computer to store the new private key.

If all the conditions necessary for remote encryption are not met, the transfer might fail. If the computer is not trusted for delegation, EFS cannot impersonate the user. When this occurs, the user receives an "Access is denied" message. If EFS can successfully impersonate the user, but the target volume does not use the version of NTFS used in Windows 2000 or later, the transfer will succeed, but the file or folder loses its encrypted status.

File operations to non-EFS capable volumes

Encrypted files might need to be decrypted before the files can be copied or moved to volumes that are not EFS-capable. For any file operation in My Computer, if the destination volume is not capable of re-encrypting the file, the following message is displayed to the user: "The file *filename* cannot be copied or moved without losing its encryption. You can choose to ignore this error and continue, or cancel." If the user chooses to ignore the message, the file is copied or moved, but it is stored in plaintext on the destination volume.

Attempts to copy encrypted files by using either the copy or xcopy commands to non-EFS capable volumes fail. In both cases, an error message informs the user that the operation failed because the file could not be encrypted. If used with appropriate parameters, however, the copy and xcopy commands can be used to copy encrypted files to non-EFS capable volumes.

Note The copy and xcopy command-line parameters that enable copying of encrypted files to non-EFS capable volumes are available in Windows XP Professional, but they are not available in earlier versions of Windows.

Table 18-4 lists the parameters for the tasks that you can perform by using the copy and xcopy commands.

Table 18-4 Tasks and Parameters for the Copy and Xcopy Commands

Task	Parameter
Copy encrypted files to non-EFS capable volumes by using the copy command.	copy /d
Copy encrypted files to non-EFS capable volumes by using the xcopy command.	xcopy /g

Remote EFS Operations in a Web Folder Environment

When users open encrypted files stored on Web folders, the files remain encrypted during the file transfer and EFS decrypts them locally. Both uploads to and downloads from Web folders are raw data transfers, so even if an attacker could access the data during the transmission of an encrypted file, the captured data would be encrypted and unusable.

EFS with Web folders eliminates the need for specialized software to securely share encrypted files between users, businesses, or organizations. Files can be stored on common intranet or Internet file servers for easy access while strong security is maintained by EFS.

The WebDAV redirector is a mini-redirector that supports the WebDAV protocol, an extension to the HTTP 1.1 standard, for remote document sharing over HTTP. The WebDAV redirector supports the use of existing applications, and it allows file sharing across the Internet (for example, using firewalls, routers) to HTTP servers. Internet Information Services (IIS) version 5.0 (Windows 2000) supports Web folders.

Users access Web folders in the same way that they access file shares. Users can map a network drive to a Web folder by using the net use command or by using My Computer. Upon connecting to the Web folder, the user can choose to copy, encrypt, or decrypt files exactly as they would by using files on file shares.

Note Web folders are not included in browse lists. To connect to a Web folder, the user must specify the full path (for example, \\ServerName\WebShareName).

Creating a Web Folder

You can create a Web folder on a server that is running Internet Information Services 5.0 or later. WebDAV is an Internet standard protocol, and WebDAV implementations are possible with other third-party Internet servers.

To create a Web folder

1. Right-click a folder on the server, and then select Properties.
2. On the Web Sharing tab, click the Share this folder button.
3. On the Edit Alias screen, enter an alias (equivalent of a share name) and appropriate permissions.

For more information about configuring WebDAV folders, see Internet Information Services 5.0 Help.

Remote Encryption of Files on Web Folders

When a user chooses to encrypt a file on a Web folder, the file is automatically copied from the Web folder to the user's computer, encrypted on the user's computer, and then returned to the Web folder. The advantage to this is that the computer hosting the Web folder does not need to be trusted for delegation and does not require roaming or remote user profiles. No other administrative tasks beyond creating the Web folder and assigning user permissions are required. The disadvantage is that the file must be transmitted from the Web folder to the local computer in order to be encrypted. Organizations need to consider whether the bandwidth requirements for Web folders outweigh the administrative effort necessary to maintain file shares for encrypted file storage. It must also be considered that Web folders are not recommended for files over 60 MB in size.

Note Bandwidth requirements are reduced if the user encrypts the file locally and then stores the file on the Web folder. The encrypted file is also transmitted in ciphertext when it is transmitted to a Web folder.

Remote Decryption of Files on Web Folders

When a user chooses to decrypt a file on a Web folder, the file is automatically copied from the Web folder to the user's computer in ciphertext. EFS then decrypts the file on the user's computer. If the user opens the encrypted file for

use in an application, the file is never decrypted anywhere except on the user's computer. If the user chooses to decrypt a file on the Web folder rather than on the local computer, the file is transmitted in plaintext and stored in plaintext on the Web folder after it is decrypted on the user's computer. The computer hosting the Web folder does not require any configuration except for the creation of the Web folder and the assigning of user permissions in order for remote decryption to function.

File Copy from a Web Folder

Encrypted files are copied from Web folders in the same way that plaintext files are copied from file shares. The copy and xcopy commands do not require any special parameters. The file is transmitted in ciphertext and remains encrypted on the local computer if possible. The encryption status for files copied from Web folders is the same as that for files copied locally. For more information about encryption status for files copied locally and about the copy and xcopy commands, see "Local and Remote File Operations in a File Share Environment" earlier in this chapter.

Delivering EFS Certificates to Users

All EFS users must have valid certificates for use with EFS. Each designated data recovery agent (DRA) must have a valid file recovery certificate. These certificates can be created and self-signed by EFS if no certification authorities (CA) are available. Users can request certificates from an enterprise CA.

How EFS Uses Certificates

When a user sets the encrypted attribute for a file or folder, EFS attempts to locate the user's certificate in the personal certificate store. If the user does not have a certificate that has been authorized for use with EFS, EFS requests a certificate from an available enterprise certification authority (CA). If an enterprise CA is not available, EFS automatically generates its own self-signed certificate for the user.

In either case, a public-private key pair for use with the certificate is generated by CryptoAPI. The public key is stored in the user's certificate, the certificate is

stored in the user's personal certificate store, and the private key is securely stored in the user's profile.

Each time a user accesses an encrypted file or creates a new encrypted file, EFS needs to access the user's EFS certificate to obtain the public key and/or to access the user's private key. In all cases, the private key is used to decrypt a file encryption key (FEK) and the public key is used to encrypt an FEK.

If the EFS user certificate expires, EFS ensures that the certificate is renewed if possible or, if not, that a new public-private key pair and a new public key certificate are issued for the user the next time an EFS operation is performed for that user.

Note Certificates that EFS generates are self-signed rather than signed by a CA. Therefore, the certification path is the same as for root CA certificates, which are also self-signed. EFS certificates that are self-signed are identified by Windows XP Professional as "not trusted" because the certifying authority does not have a certificate in the Trusted Root Certification Authorities store. Nevertheless, self-signed EFS certificates are valid for use by EFS.

Determining Whether an EFS Certificate Exists

If you upgraded from Windows 2000 Professional, a certificate might already exist for you. If your computer is not joined to a domain and you have encrypted a file, EFS has generated a self-signed certificate for you. You can determine whether you have an existing certificate using the Certificates snap-in.

To view your certificates by using the Certificates snap-in

1. Open the Certificates snap-in configured for My user account.
2. Expand the Personal folder, and then right-click Certificates.

Figure18-5 shows the EFS certificate for the account alice.

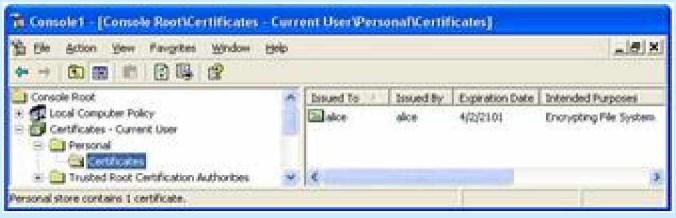

Figure 18-5 Certificates snap-in and EFS certificate

If the issuer (Issued By) is the same as the subject (Issued To), the certificate is self-signed. If the certificate is issued by a certification authority, its name is listed in the Issued By column.

Obtaining an EFS Certificate in a Stand-Alone Environment

In a stand-alone environment, EFS automatically generates EFS certificates. Users obtain a certificate by encrypting a file. Each user who logs on at the computer can encrypt files, and EFS generates a unique certificate and key pair for each user. Unless a user shares the encrypted files for others to access, no user can access another user's files.

You can also request certificates by using the Advanced Certificate page Web form.

Using Enterprise Certification Authorities to Issue Certificates

Domain environments can be configured so that EFS works just as it does in a stand-alone environment, creating self-signed certificates for users. Enterprise CAs can also be configured in the domain to create certificates for users. Certificates that are issued by enterprise certification authorities (CA) are based on certificate templates. Certificate templates are stored in Active Directory and define the attributes of certificate types to be issued to users and computers. The following three version 1 certificate templates support EFS user operations by default:

- User
- Administrator
- Basic EFS

Administrator and user certificates have a number of uses in addition to EFS. A basic EFS certificate can be used for EFS operations only.

Enterprise CAs use Access Control Lists (ACLs) for certificate templates in Active Directory to determine whether to approve certificate requests. If a user has the Enroll permission for a certificate template, the CA will issue a certificate of the type defined by the template to the user.

By default, members of the Domain Admins and Domain Users security groups have Enroll permission for basic EFS certificates and user certificates. By default, members of the Domain Admins and Enterprise Admins security groups have Enroll permission for administrator certificates.

Users can obtain an EFS certificate from an Enterprise CA by using one of these methods:

- On-demand enrollment using an Enterprise CA and properly configured certificate templates
- Manual enrollment by the end-user

Using an Enterprise CA ensures that users can easily and seamlessly obtain EFS certificates. This is also the lowest-cost option for certificate deployment.

Certificates can be requested from a CA by using the Certificates snap-in.

To request a certificate from a CA

1. Open the Certificates snap-in, expand Personal folder, right-click Certificates, and then select All Tasks and Request New Certificate. The Request New Certificate Wizard starts.
2. On the Certificate Types page, select Basic EFS, and click Next.
3. Enter a Friendly name and a Description, and click Next.
4. Click Finish to close the Request New Certificate Wizard.

Open the Certificates folder to see the new certificate. In Figure18-6, the administrator has a self-signed recovery certificate that EFS generated for the default EFS recovery policy. You can tell that the File Recovery certificate is self-signed because Issued To is the same as Issued By. Below it is the EFS certificate that was just requested. The EFS certificate was issued by a certification authority named CA1.

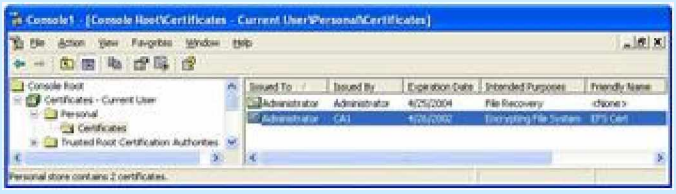

Figure 18-6 Certificates snap-in

Renewing Certificates and Keys

EFS automatically renews certificates if possible. Self-signed certificates are valid for 100 years, so renewal is not an issue if you use these certificates. Certificates issued by certification authorities are typically valid for only a few years. You can view a certificate's expiration date in the Certificates snap-in. If EFS is unable to automatically renew an expired certificate, you are unable to encrypt more files by using the existing certificate. You can still decrypt files, however, because EFS stores existing private keys. If EFS cannot renew the certificate, it requests a new certificate from a trusted enterprise CA if one is known and available. Otherwise, EFS creates a new self-signed certificate.

If for any reason you want to generate a new self-signed certificate for a user, you can use the cipher command.

To generate a new certificate by using the cipher command

- At the command line, type:

```
cipher /k
```

The output is the identifier that links the new private and public keys. You can view this identifier in the EFS certificate in the Certificates snap-in.

The following is an example of output from the cipher command.

Note Some parts of the following code snippet have been displayed in multiple lines only for better readability. These should be entered in a single line.

```
X:\>cipher /k  Your new file encryption certificate thumbnail information on the PC named WK-01 is:   AEE8 15AD 68EE B640 9CD7 C25F 4E98 4CBC C2D9 7378
```

Note If you generate a new certificate and key by using cipher /k, new files are encrypted using the new public key. Existing encrypted files are decrypted by using the old private key, which continues to be stored in the user's profile.

Replacing Self-Signed Certificates with CA-Issued Certificates

Self-signed certificates enable users to use EFS in the absence of a public key infrastructure or Active Directory. When a CA is deployed, however, PKI administrators need to migrate users from their existing self-signed certificates to CA-issued certificates. Otherwise, EFS continues to use the self-signed certificate, even if a CA has issued a new certificate.

Users can request new CA-issued file encryption certificates for EFS to replace their existing self-signed file encryption certificates by using the cipher command.

To request a CA-issued file encryption certificate

- At the command line, type:

```
cipher /k
```

This causes Windows XP Professional to archive the existing self-signed certificate and request a new one from a trusted CA. Any files that have been encrypted by using the earlier public key can still be decrypted, and when they are subsequently saved, they can be encrypted by using the new public key.

Warning Archived certificates must not be deleted. When a certificate is deleted, the corresponding private key is also deleted. Without the private key, files previously encrypted cannot be decrypted. Users can use the cipher /u command to update encrypted files on local drives using the new keys.

The cipher utility can be called in a logon script to automatically and invisibly migrate users. This utility only works locally. It cannot request new certificates for files that have been encrypted on remote servers.

Note If the cipher /k command is used to replace a self-signed certificate but no trusted enterprise CA can be located, the existing self-signed certificate is replaced with a new self-signed certificate as described in “Renewing Certificates and Keys” earlier in this chapter.

Authorizing Multiuser Access to Encrypted Files

Users can share encrypted files with other local, domain, and trusted domain users. Authorizing user access to encrypted files is a separate process from

sharing files for network access by using share-level security and access control lists. Because there is no method to issue a certificate for a group, only individual user accounts can be authorized for access to an encrypted file. Groups cannot be authorized for access.

How Users Are Authorized for Access to Encrypted Files

After a file has been encrypted, additional users can be authorized to access encrypted files by using the Advanced Attributes dialog box in the file's properties. When the original encryptor or another authorized user shares the encrypted file with another user, EFS uses the following process:

1. An authorized user opens the Advanced Attributes dialog box for the encrypted file and clicks the Details button.
2. In the Encryption Details dialog box, the user clicks the Add button to open the Select User dialog box. EFS certificates stored in the user's profile in the Other People and Trusted People certificate stores are automatically displayed. To locate user certificates that are stored in Active Directory, the user can click the Find User button and then click the Find Now button to locate the selected user(s). A dialog box will display any users that hold valid EFS certificates in the Active Directory based on the search criteria.

   EFS certificates can also be imported to the user's profile. If the EFS certificate is self-signed, it is added to Trusted People. If the EFS certificate was issued by a CA, it is added to Other People. Certificates added to either container appear in the Select User dialog box.

   Note If you import someone's certificate, you have the option to manually select its location. If you import a self-signed certificate, be sure to add it to Trusted People. Certificates in Trusted People are the only certificates that are not chain validated because the certificates are already trusted. Self-signed certificates will always fail chain validation because no CA was involved in certifying them, so placing them anywhere other than in Trusted People makes them unusable.

3. Before a user can be authorized to access an encrypted file and be added to the file, EFS needs to determine whether the certificate can be trusted. When a user's certificate is selected, EFS attempts to validate the certificate chain. If the chain validation fails, CryptoAPI also checks to see whether the certificate is in Trusted People store. If it is not, the certificate cannot be used.

   Imported self-signed certificates are automatically placed in Trusted People, and there is no certificate chain to validate, so the user can be added to the encrypted file.

   If the certificate was signed by a CA, EFS attempts to build a certification chain and validate the certificate. If the chain ends with an untrusted root CA, EFS will not use the certificate and the user is not added to the file. If the user's EFS certificate is signed by a CA and includes information about certificate revocation list (CRL) distribution points, EFS attempts to connect with the distribution points to check for certificate revocation. If a CRL distribution point cannot be reached, the certificate will not be used, even if the root is trusted. Finally, if the EFS certificate is signed by a CA, the CA is trusted, and the CA does not use CRL distribution points, EFS accepts the certificate and adds the user.

4. For the next part of the process, assume that Alice already has access to the encrypted file. She is either the original encryptor, or she has been authorized for access. Alice is authorizing Bob to access the file.

5. EFS obtains Alice's private key from her user profile to decrypt the file encryption key (FEK) contained in the file's data decryption field.

6. EFS obtains Bob's public key from his certificate and uses it to encrypt the FEK.

7. Bob's encrypted FEK is stored in a new data decryption field with the file.

   Note At no time in this process is the file itself decrypted, so it is not at risk directly. However, the FEK is briefly decrypted. Because EFS performs this operation in nonpaged memory, the decrypted FEK is never paged and so is never exposed.

Figure 18-7 shows what happens when Alice shares an encrypted file with Bob.

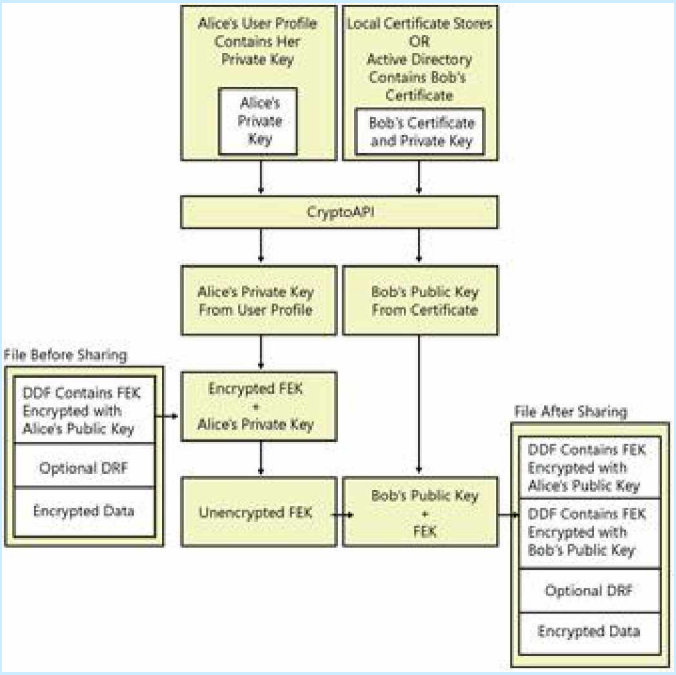

Figure 18-7 Sharing an encrypted file

Considerations for Sharing Encrypted Files

It is important that users electing to share encrypted files keep the following points in mind:

1. Shared EFS files are not file shares. If authorized users need to access shared EFS files over the network, a file share or Web folder is required. Alternatively, users could establish remote sessions with computers that store encrypted files by using Terminal Services.
2. Any user who is authorized to decrypt a file can authorize other users to access the file. Granting access is not limited to the file owner. Caution users to share files only with trusted accounts, because those accounts can authorize other accounts. Removing the Write permission from a user or group of users can prevent this problem, but it also prevents the user or group from modifying the file.
3. EFS sharing requires that the users who will be authorized to access the encrypted file have EFS certificates. These certificates can be located in roaming profiles or in the user profiles on the computer on which the file

to be shared is stored, or they can be stored in and retrieved from Active Directory.

4. EFS sharing of an encrypted file often means that the file will be accessed across the network. It is best if Web folders are used for encrypted file storage whenever possible. For more information about using Web folders to share encrypted files, see “Remote EFS Operations on File Shares and Web Folders” earlier in this chapter.
5. If a user chooses to remotely access an encrypted file stored on a file share and to authorize other users to access the file, the authorization process and requirements are the same as on the local computer. Additionally, EFS must impersonate the user to perform this operation, and all the requirements for remote EFS operations on files stored on file shares apply. For more information about the requirements for EFS operations, see “Remote EFS Operations on File Shares and Web Folders” earlier in this chapter.
6. If a user chooses to remotely access an encrypted file stored on a Web folder and to authorize other users to access the file, the file is automatically transmitted to the local computer in ciphertext. The authorization process takes place on the local computer with the same requirements as for encrypted files stored locally.

Sharing Encrypted Files

You can authorize individual users to access encrypted files.

To share an encrypted file with other users

1. In My Computer, right-click the encrypted file, and then click Properties.
2. On the General tab, select Advanced.
3. In the Advanced Attributes dialog box, under Compress or Encrypt Attributes, select Details.

Note If you select an encrypted folder instead of an encrypted file, the Details button appears dimmed. You can add users to individual encrypted files but not to folders.

4. In the Encryption Details dialog box, click Add.
5. Add a user from the local computer or from Active Directory.

To add a user from the local computer

- In the Select Users dialog box, click the user's certificate, and then click OK.

To add a user from Active Directory

1. Click Find User. In the Find Users, Contacts, and Groups dialog box, click Browse to search for users.
2. In the Browse for Container dialog box, click the folder or domain in which you want to begin your search. You can perform your search in the entire directory or start searching from a folder or domain within the directory.
3. To narrow the search, click Advanced and then click Field to search for users by using conditions and values.
4. Select the user, and then click OK.

   Caution Any authorized user of an encrypted file can authorize other users' access, so it is important to authorize access for trusted accounts only.

For more information on how to share encrypted files, see article 308991, "How to Share Access to an Encrypted File in Windows XP," in the Microsoft Knowledge Base at http://support.microsoft.com.

Taking Recovery Precautions

Encrypting a file always includes a risk that it cannot be read again. The owner of the private key, without which a file cannot be decrypted, might leave the organization without decrypting all of his or her files. Worse yet, he or she might intentionally or accidentally encrypt critical shared files so that no one else can

use them. A user's profile might be damaged or deleted, meaning that the user no longer has the private key needed to decrypt the file's FEK. Because losing data is often disastrous, there are four methods of recovering when encrypted files cannot be decrypted. You can use data recovery agents, export and import of EFS recovery keys, and Backup for file backup and restoration.

Data Recovery and Data Recovery Agents

You can use Local Security Policy on stand-alone computers to designate one or more users, typically Administrator accounts, as data recovery agents. These data recovery agents (DRAs) are issued recovery certificates with public and private keys that are used for EFS data recovery operations.

The default design for the EFS recovery policy is different in Windows XP Professional than it was in Windows 2000 Professional. In Windows XP Professional, stand-alone computers do not have a default DRA, but Microsoft strongly recommends that all environments have at least one designated DRA.

In a Windows 2000 environment, if an administrator attempts to configure an EFS recovery policy with no recovery agent certificates, EFS is automatically disabled. In a Windows XP Professional environment, the same action allows users to encrypt files without a DRA. In a mixed environment, an empty EFS recovery policy turns off EFS on Windows 2000 computers, but eliminates the requirement for a DRA only on Windows XP Professional computers.

When a domain user logs on at a domain computer that is within the scope of the EFS recovery policy, all DRA certificates are cached in the computer's certificate store. This means that EFS on every domain computer can easily access and use the DRA's public key (or multiple public keys if multiple DRAs are designated). On computers where an EFS recovery policy is in effect, every encrypted file contains at least one data recovery field, in which the file's FEK is encrypted by using the DRA's public key and stored. By using the associated private key, any designated DRA can decrypt any encrypted file within the scope of the EFS recovery policy.

Warning The private key for a DRA must be located on the computer where recovery operations are to be conducted.

Because each DRA has a distinct private key that can decrypt a file's FEK, data recovery discloses only the encrypted data, not the private keys of any user other than the DRA. A private key for recovery cannot decrypt the data decryption field (DDF). If there are multiple recovery agent accounts, each private key for recovery decrypts only its own data recovery field (DRF) and no other. Thus, there is no danger that an unauthorized recovery agent account can access information from the file that enables access to other files.

Note It is best not to encrypt files when you are logged on with a DRA account. The effectiveness of EFS recovery is compromised if a file's creator is both the user and the recovery agent.

For more information about encrypted file recovery, see Windows XP Professional Help and Support Center.

For more information about how to change EFS recovery policy for the local computer, see Windows XP Professional Help and Support Center and "Configuring Data Recovery Policy in a Stand-Alone Environment" later in this chapter.

Data Recovery Implementation Considerations

When designating DRAs for your files, it is important to keep the following considerations in mind:

1. Files are more secure, but less recoverable, if only one person can decrypt them. If you choose not to designate any DRAs, be sure that user profiles are backed up regularly. The user's private key, without which the file cannot be decrypted, is stored in the user profile. Additionally, because users can have multiple profiles on multiple computers or might have a roaming profile, be sure that all profiles are being backed up. This is especially true for notebook computers. If the user logs on with a local account and encrypts a file, the private key for that transaction is contained in the user profile for the local account, not the user's domain account profile.

   The most effective way for users to ensure access to encrypted files is to export their EFS certificates and private keys. For more information about

exporting EFS certificates and keys, see “Exporting and Importing EFS and DRA Certificates and Private Keys” later in this chapter.

2. Files are less secure, but more recoverable, if more than one person can decrypt them. If you want the ability to recover encrypted files, designate one or more DRAs by using the EFS recovery policy.
3. For more flexible EFS recovery management, consider issuing EFS recovery certificates to designated recovery agent accounts in addition to the default Administrator account. Also, legal or corporate policy might require that the recovery agent account be different from the domain Administrator account. You can also configure EFS recovery policy for portable computers to use the same recovery agent certificates, whether the computers are connected to domains or are operated as stand-alone computers.
4. EFS is normally used to encrypt user data files. Application files (for example, .exe, .dll, .ini files) are rarely encrypted. If computers are configured to use System Restore as part of recovery policy, application files are saved at restore points. These files can then be restored to return the system to a previous state. If application files that System Restore is monitoring are encrypted, it is important to note the following expected results of restoration:
    - If you decrypt a previously encrypted monitored file or folder and then restore to a point before the file or folder was decrypted, the restored file or folder remains decrypted. If you undo the restore, the file or folder remains decrypted.
    - If you encrypt a monitored file and then restore to a point before the monitored file was encrypted, the restored file is unencrypted. If you undo the restore, the file remains unencrypted.
    - If you modify a monitored file that is encrypted for multiple users and then restore to a point before the modification occurred, the file will be accessible only to the first user who modified the file after the restore point was created. If you undo the restore, only the user who ran the restore will be able to access the file. The

filter will back up the file during restore in the context of the user who is running the restore operation.

- If you delete a monitored encrypted file and then restore to a point before the deletion, the deleted file is restored with its encryption attributes intact. If you undo the restore, the file is again deleted.
- If you encrypt a directory and then restore to a point before it was encrypted, the directory remains encrypted. Monitored files created in this directory after the restore point are encrypted, but they will be deleted by a restore. Monitored files moved into this directory retain the encryption status from the directory in which they were created. After the restore, monitored files will be moved back to their original directory and retain the encryption status from that directory. If you undo the restore, the directory remains encrypted and files are returned.
- If you modify an encrypted directory (for example, change its name) and then restore to a point before the modification, the modification is lost, but the directory remains encrypted. If you undo the restore, the modification returns and the directory remains encrypted.
- If you delete a directory that is encrypted and then restore to a point before it was deleted, the directory retains its encryption status. If you undo the restore, the directory is again deleted.

If you encrypt application files, consider these recommendations:

- The best practice is to place these files in a partition/drive that is excluded from System Restore protection. This reduces the risk of restoring files to a pre-encrypted state.
- If you choose to encrypt application files and use System Restore on the partition/drive storing the files, turn off System Restore (losing all previous restore points), complete the encryption settings, and then turn System Restore back on. This ensures that the files cannot be reverted to a pre-encrypted state.

To configure System Restore

1. On the System Tools menu, click System Restore.
2. Select System Restore Settings, and configure the options as appropriate for your environment.

   Note You must be an administrator to configure System Restore.

Data Recovery Agent Decryption Process

Decrypting a file for a data recovery agent (DRA) is identical to the process described earlier for a user except that the copy of the FEK encrypted by using the DRA's public key is taken from the file's DRF, decrypted, and used to decrypt the file.

Figure 18-8 illustrates the process of obtaining the DRA's private key from his or her user profile, using it to decrypt the FEK, and using the FEK to decrypt the data.

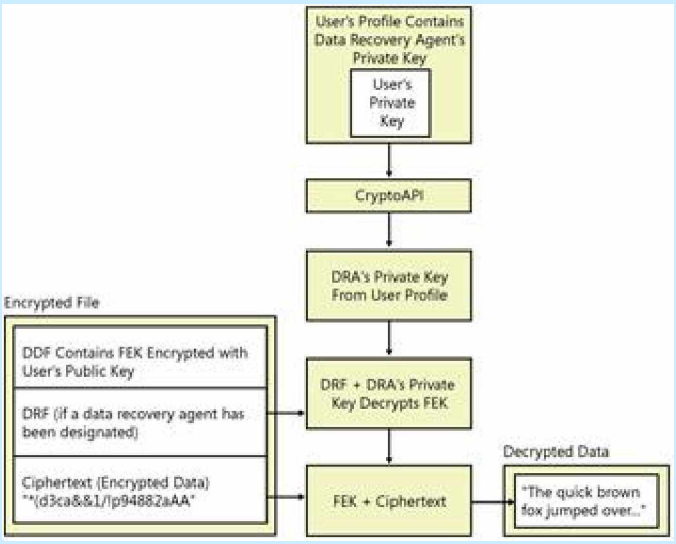

Figure 18-8 Decrypting a file for a data recovery agent

Configuring Data Recovery Policy in a Stand-Alone Environment

In a stand-alone environment, an administrator for the computer needs to generate a recovery agent certificate and designate a DRA.

Designating a Data Recovery Agent in a Stand-Alone Environment

For stand-alone computers, Windows XP Professional does not create a default recovery agent. A DRA can be added by using Group Policy on the local

computer, but the intended DRA must first have a recovery certificate. Because the computer is stand-alone, EFS creates a self-signed certificate for the DRA. This requires the cipher.exe command-line utility.

To generate a recovery agent certificate

1. Log on as an administrator.
2. At a command prompt, type:

```
cipher /r:filename
```

This generates importable .pfx and .cer files with the file names you specify.

To designate a DRA

1. Log on as the intended DRA.
2. Open the Certificates snap-in, and import the .cer file containing the recovery agent certificate.

   – or –

   Import the .pfx file containing the recovery key.

   Note For more information about importing certificates, see “Exporting and Importing EFS and DRA Certificates and Private Keys” later in this chapter.

The local administrator can now open the Group Policy MMC snap-in and add the DRA.

To add a DRA

1. In the Local Computer node of Group Policy, expand Computer Configuration.
2. Expand Windows Settings, Security Settings, Public Key Policies, and Encrypting File System.
3. Right-click Encrypting File System, and select Add Data Recovery Agent.
4. Follow the steps of the Add Recovery Agent Wizard.

The local administrator can also use the Group Policy MMC snap-in to remove a DRA.

To remove a DRA

1. In the Local Computer node of Group Policy, expand Computer Configuration.
2. Expand Windows Settings, Security Settings, Public Key Policies, and Encrypting File System.
3. Select the DRA to remove and delete the certificate.

Exporting and Importing EFS and DRA Certificates and Private Keys

As added protection in the event that a user's certificate or private key is damaged or lost, certificates and private keys can be exported. The certificate is stored in a file with a .cer extension, and the certificate and private key are stored in a password-protected file with a .pfx extension. These archive files can be stored on a floppy disk or other medium, and they can be imported easily from these locations.

Exporting these keys does not automatically remove them from the system; however, it is possible to remove the private key after it has been exported. Public keys are always available. They do not need to be removed. Circumstances might warrant removal of the private key, however.

If an EFS user wants to safeguard against loss of a private key through, for example, a damaged or deleted user profile, it can be exported and protected with a strong password. Keys can be exported to a floppy disk, and the disk can be stored off-site or locked in a vault, depending on the level of security needed.

If one or more DRAs have been designated by Group Policy, it is important that those keys be exported. In addition:

- Use strong passwords to protect exported private keys.
- After export, delete exported private keys from the system.
- Store exported DRA keys on a floppy disk or other media in a physically secure location.

In a domain environment, when DRAs are designated by Group Policy, any DRA can recover (decrypt) any domain user's encrypted files. Each encrypted file can have one or more data recovery fields, in which a copy of the file's FEK is encrypted by using a DRA's public key and stored. The DRA's private key, however, is not needed unless an EFS user is unable to decrypt a file. If a file needs to be recovered, a DRA can either import his or her private key on the computer where the encrypted file is stored, or the user can use the Backup utility included with Windows XP Professional to back up the encrypted file and make it available for the DRA to restore on a computer used for data recovery.

Caution Every DRA has a personal EFS certificate and private key. Normally, when the DRA recovers a file, the DRA's personal private key is used. However, recovery certificates and keys are not bound to a specific user. Anyone who has access to a DRA's private key can import that key and use it to decrypt any files for which the DRA is a recovery agent. Therefore, it is extremely important to protect exported private keys by using strong passwords and physically secure storage.

Exporting Certificates and Keys

You can use the Certificate Export Wizard to export a certificate and private key to a removable medium. The same process is used to export any certificate.

To export a certificate

1. Open the Certificates snap-in, and then expand the Personal folder.
2. Double-click Certificates, and then right-click the certificate you want to export.
3. Select All Tasks, and then select Export.
4. Select Yes, export the private key.

   The export format is PCKS #12. You have the option of deleting the private key or leaving it on the computer. If you are backing up an EFS certificate and keys, you might want to leave the private key on the computer so that you can decrypt your files without importing the private key. If you are backing up a file recovery certificate and keys, it is best if you delete the

private key after the export. The private key is needed only for disaster recovery, and files are more secure if the private key is removed.

5. Select an option, and then click Next.
6. Enter a password to protect your exported private key. It is best if you use a strong password.
7. Click Next.
8. Enter a file name for the exported certificate and private key.
9. Click Next, review the final information, and then click Finish.

   The export file has a .pfx extension.

Importing Certificates

The same process is used to import either an EFS certificate or a File Recovery certificate.

To import a certificate

1. Open the Certificates snap-in, and then expand the Personal folder.
2. Right-click the Certificates folder, click All Tasks, and click Import.

   This starts the Certificate Import Wizard. You can also start this wizard by double-clicking a certificate file.
3. In the dialog box, enter the name of your certificate file.
4. Enter the password that protects the private key. This password was created during export. You have the option of protecting the imported private key by using the same password. If you enable this option, you will be prompted to enter this password when you attempt to open an encrypted file.

   Note Enabling the option to protect the imported private key adds another layer of protection. However, if the password needed to open the file is forgotten, you will not be able to open the files. A designated data recovery agent would have to import the file recovery private key and open the file.

You also have the option of marking the key as exportable. If you are importing from a backup copy of your keys and plan to keep the backup copy, it is best not to select this option. This prevents an attacker from exporting your EFS private key.

5. Designate a location for the certificate. The default location is the personal certificate store.

Backing Up and Restoring Encrypted Files or Folders

In Windows XP Professional, encrypted files and folders remain encrypted if you back them up by using Backup in Administrative Tools. You can also use the ntbackup command, the backup APIs, or other backup products designed for use with Windows XP Professional. Backup files remain encrypted when transferred across the network or when copied or moved onto any storage medium, including non-NTFS volumes. If backup files are restored to volumes formatted by using the version of NTFS used in Windows 2000 or later, they remain encrypted. Along with providing excellent disaster recovery, backups can also be used to securely move files between computers, sites, and so on.

Opening restored, encrypted files is no different from decrypting and opening any encrypted files. However, if files are restored from backup onto a new computer, in a new forest, or at any location at which the user's profile (and thus the private key needed to decrypt the files) is not available, the user can import an EFS certificate and private key. After importing the certificate and private key, the user can decrypt the files.

Recovering Encrypted Files

Any data recovery agent can recover an encrypted file when a user's private key fails to decrypt the file.

To recover an encrypted file

1. Log on to a computer that has access to the user's profile; for example, a computer that belongs to a designated recovery agent or has a recovery key available on removable media such as a floppy disk. You might also log on at the user's computer, or the user might have a roaming profile.

2. Locate the encrypted file. For example, the user might have made a backup of the file by using Backup or sent the file to a WebDAV Web folder.
3. Decrypt the file by using either the cipher command or My Computer. This will make the file available to the user.

   For more information about decrypting files, see "Working with Encryption and Decryption" earlier in this chapter.

Strengthening Key and File Security

When an encrypted file is saved, Windows XP Professional automatically provides four levels of encryption and a fifth, the startup key, can be configured. These levels of encryption protect the encryption keys that are used to protect the file:

1. EFS provides the FEK, which encrypts the data in the file.
2. EFS uses the public key in the user's EFS certificate to encrypt the FEK. The public key and certificate are stored by default in the computer's certificate store. The corresponding private key, used to decrypt the FEK, is stored in an encrypted form in the user's profile for the corresponding user or the data recovery agent account in the RSA folder.
3. The Data Protection API generates the user's master key that is used to encrypt the user's private keys.
4. The Data Protection API generates a symmetric password encryption key, derived from a hash of the file creator's credentials, which encrypts the user's master key.
5. A startup key (also called the syskey) can be used to protect all master keys as well as a variety of other secrets that are stored on computers. At system startup, the startup key is used to encrypt all the private keys on the computer, including private keys that are used for EFS. Startup keys are automatically generated and used for computers in a domain but must be manually configured on stand-alone computers. Startup key security can be increased by storing the key on removable media or by requiring a system startup password.

Windows XP Professional provides several configuration options that increase security. To prevent loss of access to your master keys on stand-alone computers, you can create a password reset disk (PRD). To provide stronger encryption for files, you can enable the 3DES encryption algorithm. In addition, you can implement options to prevent your computer from entering hibernation mode and to delete paging files at system shutdown so that data is not needlessly exposed.

Certificate and Public Key Storage

Windows XP Professional stores a user's public key certificates in the user's personal certificate store. Certificates are stored in plaintext because they are public information, and they are digitally signed by certification authorities to protect against tampering.

User certificates are located in the *RootDirectory*\Documents and Settings\username \Application Data\Microsoft\SystemCertificates\My\Certificates folder for each user profile. These certificates are written to the user's personal store in the system registry each time the user logs on to the computer. For roaming profiles, users' certificates are located on the domain controller and follow users when they log on to different computers in the domain.

Private Key Storage

Private keys for the Microsoft RSA-based cryptographic service providers (CSPs), including the Base CSP and the Enhanced CSP, are located in the user profile under *RootDirectory*\Documents and Settings*username*\Application Data\Microsoft \Crypto\RSA. In the case of a roaming user profile, private keys reside in the RSA folder on the domain controller and are downloaded to the user's computer until the user logs off or the computer is restarted.

Because private keys must be protected, all files in the RSA folder are automatically encrypted by using a random symmetric key called the user's master key. The user's master key is 64 bytes in length and is generated by a strong random number generator. 3DES keys are derived from the master key and are used to protect private keys. The master key is generated automatically

and is periodically renewed. It encrypts each file in the RSA folder automatically as the file is created.

For more information about CryptoAPI, see the Microsoft Platform SDK link on the Web Resources page at http://www.microsoft.com/windows/reskits/webresources.

For more information about Data Protection API, see the Technet link on the Web resources page at http://www.microsoft.com/windows/reskits/webresources.

Caution The RSA folder must never be renamed or moved. This is the only place the CSPs look for private keys. If you need additional protection for this folder, the administrator can provide additional file system security for users' computers or use roaming profiles.

Master Key Storage and Security

The Data Protection API automatically encrypts the user's master key or keys. Master keys are stored in the user profile under *RootDirectory*\Documents and Settings*username*\Application Data\Microsoft\Protect. For a domain user who has a roaming profile, the master key is located in the user's profile and is downloaded to the user's profile on the local computer until the computer is restarted.

While the user is logged on, when a master key is not being used for a cryptographic operation, it is encrypted and stored on disk. Before master keys are stored, they are 3DES-encrypted using a key derived from the user's password. When a user changes his or her logon password, master keys are automatically unencrypted and re-encrypted using the new password.

Master Key Loss and Data Recovery

If a logon password is forgotten or if an administrator resets a user password, the user's master keys become inaccessible. Because the decryption key is derived from the user's password, the system is unable to decrypt the master keys. Without the master keys, EFS-encrypted files are also inaccessible to the user and can be recovered only by a data recovery agent, if one has been configured, or through the use of a password reset disk (PRD), if one has been created. For more

information, see article 290260, “EFS, Credentials, and Private Keys from Certificates Are Unavailable After a Password Is Reset,” in the Microsoft Knowledge Base at http://support.microsoft.com.

Note The password reset disk feature is available for local accounts only. Domain account passwords cannot be backed up by using a PRD.

For example, Alice uses EFS on her stand-alone computer to encrypt her private files. She forgets her password but remembers the password for the built-in administrator account. She logs on as the administrator and resets the password for her account named Alice. When she resets the password, she is able to log on as Alice, but files that she encrypted as Alice are inaccessible because the master keys are inaccessible.

Alice can recover her encrypted data in two ways. First, if a data recovery agent was configured for Alice’s computer before she encrypted the files, the data recovery agent (DRA) can recover the files. Second, even without a DRA, if Alice created a PRD before she lost or reset her password, she can use the PRD to safely change her password and recover her data.

Password Reset Disk Creation

PRDs protect against loss of access to master keys and encrypted files on stand-alone computers. Users can create PRDs for their own user accounts. When a user creates a PRD, the system creates a public-private key pair and a self-signed certificate. The user’s password is encrypted by using the public key and stored locally in the registry at HKEY_LOCAL_MACHINE \SECURITY\Recovery\<*user SID*>.

The private key is exported to a removable media and deleted from the local computer.

To create a password reset disk for your user account

1. In Control Panel, click User Accounts.
2. Click your user account.
3. Under Related Tasks, select Prevent a forgotten password. This starts the Forgotten Password Wizard.

4. On the Welcome screen, click Next.
5. Select the removable drive on which you would like to store your password key, and then click Next.
6. Enter your current password, and click Next.

If you have created a PRD and you forget your password or enter the wrong password, you will be prompted with the following message: “Did you forget your password? You can use your password reset disk.” You can then use the Password Reset Wizard to reset your password. The system will do the following:

1. Use the private key stored on the PRD to decrypt the stored copy of your old password.
2. Create a decryption key based on an SHA-1 hash of your old password.
3. Decrypt your master keys by using the decryption key.
4. Prompt you for a new password.
5. Encrypt master keys by using your new password.

To use the PRD to reset your password

1. At the logon screen, click Use your password reset disk to start the Password Reset Wizard and then click Next.
2. Insert the removable media that contains your password key, and then click Next.
3. Enter your new password in the Reset the User Account Password dialog box, and then click Next.

Enabling the Startup Key

The Syskey Wizard enables startup key protection. If enabled, the startup key protects the following sensitive information:

- Master keys that are used to protect private keys.
- Protection keys for user account passwords stored in Active Directory.

- Protection keys for passwords stored in the registry in the local Security Accounts Manager (SAM) registry key.
- Protection keys for Local Security Authority (LSA) secrets.
- The protection key for the administrator account password that is used for system recovery startup in safe mode.

You must be a member of the local Administrators group to use the syskey command. Using this utility, an administrator can configure the system to do one of the following:

1. Use a computer-generated random key as the startup key, and store it on the local system by using a complex obfuscation algorithm that scatters the startup key throughout the registry. This option enables computer restarts without the need to enter the startup key.
2. Use a computer-generated random key, but store it on a floppy disk. The floppy disk must be inserted into a drive during system startup for the startup sequence to complete. This option is more secure than the first, but it effectively rules out restarting the computer remotely.

   Warning If the startup key password is forgotten or the floppy disk that contains the startup key is lost, it might not be possible to start the system. If this occurs, the only way to recover the system is to use a repair disk to restore the registry to a state prior to when startup key protection was enabled. Any changes made after that time would be lost so it is important to store the startup key safely. If it is on a floppy disk, make backup copies and store them in different locations.
3. Use a password chosen by the administrator to derive the startup key. The administrator is prompted for the password during the initial startup sequence.

   Note After startup key protection is enabled, it cannot be disabled, but it can be configured to operate at different security levels.

To enable startup key protection

1. At the command line, type syskey.

2. Click Encryption Enabled, and then click *OK*.

   – or –

   Click Update if encryption was previously enabled.

3. Select an option for the key.

   The default option is a system-generated password that is stored locally. If you use the password-derived startup key option, syskey does not enforce a minimum password length. However, passwords longer than 12 characters are best. The maximum length is 128 characters.

4. Click OK to restart the computer.

   When the system restarts, you might be prompted to enter the startup key, depending on the key option you selected. The first use of the startup key is detected and a new random password encryption key is generated. The password encryption key is protected by using the startup key, and then all account password information is strongly encrypted.

After the startup key has been enabled, the following process occurs at system startups:

- The startup key is retrieved from the locally stored key, the password entry, or insertion of a floppy disk, depending on the option you selected.
- The startup key is used to decrypt the master protection key.
- The master protection key is used to derive the per-user account password encryption key, which is then used to decrypt the password information in Active Directory or the local SAM registry key.

The syskey command can be used again later to change the startup key storage option or to change the password. Changing the startup key requires knowledge of, or possession of, the current startup key.

To change the startup key option or password

1. At the command line, type syskey.
2. In the first dialog box, click Update.

3. In the next dialog box, select a key option or change the password and then click OK.
4. Restart the computer.

Enabling 3DES

You can strengthen security by replacing the default DESX algorithm with 3DES. In a stand-alone environment, enabling 3DES is recommended. In a domain environment, the appropriate encryption algorithm is typically determined by corporate policy.

Note As of Service Pack 1 for Windows XP, the Advanced Encryption Standard (AES) algorithm is now used by default for encrypting files with EFS. For more information, see article 329741, "EFS Files Appear Corrupted When You Open Them," in the Microsoft Knowledge Base at http://support.microsoft.com. This change in the default operation of EFS might affect consideration of whether you want to implement 3DES encryption.

3DES can be enabled by using the system cryptography Group Policy setting. If this setting is configured for 3DES, IP Security and EFS both use 3DES for encryption. Not all IP Security implementations are capable of encrypting by using 3DES, however. Windows 2000 without the High Encryption Pack, for example, cannot use 3DES. It is also possible to configure EFS to use 3DES without affecting encryption elsewhere. This requires modification of a registry setting.

To enable system-wide 3DES by using Group Policy

1. In Computer Configuration, expand Windows Settings, Security Settings, Local Policies, and then expand Security Options.
2. Double-click System cryptography: Use FIPS compliant algorithms for encryption.
3. Select Enabled, and then click OK.

To enable 3DES for EFS only, a registry entry must be added.

To enable 3DES for EFS only

1. In the Run dialog box, type regedit.exe.
2. Navigate to the subkey HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \Windows NT\CurrentVersion\EFS.
3. On the Edit menu, point to New and then click DWORD Value.
4. Enter AlgorithmID for the value name and 0x6603 for the value data to enable 3DES.
5. Restart the computer.
6. To disable 3DES and enable DESX, simply delete the AlgorithmID setting and restart the computer.

   Caution Do not edit the registry unless you have no alternative. The registry editor bypasses standard safeguards, allowing settings that can damage your system or even require you to reinstall Windows. If you must edit the registry, back it up first.
7. When 3DES is enabled, files encrypted by using both the DESX and 3DES algorithms can be decrypted. However, all new files are encrypted by using the 3DES algorithm.

   Caution If a user needs to access an encrypted file from both Windows 2000 and Windows XP Professional, do not enable the 3DES algorithm. The Windows 2000 operating system does not support the 3DES algorithm by default. However, if the High Encryption Pack has been installed on Windows 2000, it can use 3DES.

Increasing Security for Open Encrypted Files

File data is decrypted before it is sent to an application. This means that the file encryption key (FEK) is also decrypted. Although the FEK is not exposed, file data might be.

Because the EFS File System Run-Time Library (FSRTL) is located in the Windows operating system kernel and uses the nonpaged pool to store the FEK,

FEKs cannot be leaked to paging files. However, because the contents of paging files are not encrypted, the plaintext contents of encrypted files might temporarily be copied to paging files when open for application use. If the plaintext contents of encrypted files are copied to a paging file, the plaintext remains in the paging file until the contents are replaced by new data. Plaintext contents can remain in paging files for a considerable amount of time even after applications close the encrypted files.

A paging file is a system file, so it cannot be encrypted. (By default, the name of the paging file is Pagefile.sys.) The file system security for paging files prevents any user from gaining access to and reading these files, and these security settings cannot be changed. However, someone other than the authorized user might start the computer under a different operating system to read a paging file.

To prevent others from reading the contents of paging files that might contain plaintext of encrypted files, you can do either of the following:

- Disable hibernation mode on your computer.
- Configure security settings to clear the paging files every time the computer shuts down.

Disabling Hibernation Mode

When a computer hibernates, the contents of system memory and any open files are written to a storage file on the hard drive, and the system is powered off. This saves energy and allows the computer to be restarted with the same applications and files that were open when the system hibernated. However, hibernation can be a security risk because files are decrypted for use in applications. If an encrypted file is opened and then the system is hibernated, the contents of the open encrypted file will be in the hibernation storage file as plaintext. An attacker could potentially access the storage file used during hibernation. For this reason, EFS users might want to disable hibernation so that encrypted files are not placed at risk. If you choose to use hibernation mode, be sure to close any open encrypted files before letting the system hibernate.

To disable hibernation

1. In Control Panel, double-click Performance and Maintenance, and then click Power Options.
2. On the Hibernate tab, clear the Enable hibernate check box.
3. Click Apply.

Clearing the Paging File at Shutdown

When a file is encrypted or decrypted, plaintext data can be paged. This can be a security problem if an attacker boots the system by using another operating system and opens the paging file. The paging file can be cleared at shutdown by means of Group Policy.

To clear the paging file at shutdown

1. In the Group Policy snap-in, select a Group Policy object to edit.
2. Expand Computer Configuration and Windows Settings, Security Settings, Local Policies, and then expand Security Options.
3. Double-click Shutdown: Clear virtual memory pagefile.
4. Click Enabled, and then click OK.

Disabling EFS

EFS is enabled by default but can be disabled for individual files or individual file folders, or they can be disabled entirely for a computer or domain. Disabling EFS for a stand-alone computer requires adding an entry to the registry.

Disabling EFS for an Individual File

Although files can be made unencryptable by setting the system attribute or placing the file in the %systemroot% folder or any of its subfolders, these options are undesirable in many cases. For example, system files are also normally hidden from view, and a user might want a file that is unencryptable to be visible to other users.

Note Even with Write permission, users cannot encrypt files or folders in the %systemroot% folder, or files or folders that have the system attribute set. Many

of these files are needed for the system to start up and decryption keys are not available during the startup process to decrypt them.

Denying Write permissions for a file also makes it unencryptable by the users or groups within the scope of the denial. Simply attributing the file as read-only, however, does not prevent encryption. A user who has Write permissions can encrypt read-only files.

In most cases, the best solution is to disable EFS for a folder rather than an individual file.

Disabling EFS for a File Folder

To disable encryption within a folder, create a file called Desktop.ini that contains:

```
[Encryption] Disable=1
```

Save the file in the directory in which you want to disable EFS. If a user attempts to encrypt the folder or any files in the folder, a message tells the user that “An error occurred applying attributes to the file: *filename*. The directory has been disabled for encryption.”

Note The Desktop.ini file affects only the current folder and the files in it. If you create a subfolder, both the subfolder and any files in it can be encrypted. Also, encrypted files can be copied or moved, without losing their encryption, into the directory that contains the Desktop.ini file.

Disabling EFS for a Stand-Alone Computer

A registry entry must be added to disable EFS for a stand-alone computer.

To disable EFS on a stand-alone computer by editing the registry

1. In the Run dialog box, type regedit.exe.
2. Navigate to the subkey HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft \Windows NT\CurrentVersion\EFS.
3. On the Edit menu, point to New, and then click DWORD Value.

4. Enter EfsConfiguration for the value name and 1 for the value data to disable EFS. (A value of 0 enables EFS.)
5. Restart the computer.
6. If EFS is disabled and a user tries to encrypt a file or folder, a message tells the user that “An error occurred applying attributes to the file: filename. The directory has been disabled for encryption.”

   Note Do not edit the registry unless you have no alternative. The Registry Editor bypasses standard safeguards, allowing settings that can damage your system, or even require you to reinstall Windows. If you must edit the registry, back it up first.

Tips for Implementing EFS

Encryption is a sensitive operation. It is important that encrypted data not become decrypted inadvertently. To this end, observe the following best practices:

- The most important practice is to have users export their EFS certificates and private keys to removable media, and store the media securely. It is best if users do not remove the private keys from their computers because they will not be able to decrypt any files without importing the private key. The most common EFS-related problem that users encounter is the inability to decrypt files after losing access to the EFS private key in the user profile. Users who switch between operating systems can also benefit from exporting their EFS certificates and private keys. If both operating systems can use EFS, users can import their certificates and keys. This enables them to access their encrypted files when they are logged on to either operating system.
- Export the private keys for recovery accounts, store them in a safe place on secure media, and remove the keys from computers. This prevents someone from using the recovery account on the computer to read files that are encrypted by others. This is especially important for stand-alone computers where the recovery account is the local Administrator or

another local account. For example, a portable computer that contains encrypted files might be lost or stolen, but if the private key for recovery is not on the computer, no one can log on as the recovery account and use it to recover files.

- The private keys associated with recovery certificates are extremely sensitive. Export each such key into a .pfx file, protected under a strong password, to removable media, and then physically secure the media.
- Do not use the recovery agent account for any other purpose.
- Do not destroy recovery certificates and private keys when recovery agent policy changes. Keep them in archives until you are sure that all files that are protected by them have been updated with new recovery agent information.
- Encrypt the My Documents folder (*RootDirectory**UserProfile*\My Documents). This ensures that personal folders where most Microsoft Office documents are saved are encrypted by default. Remember, however, that no files or directories in roaming profiles can be encrypted.
- Encrypt folders rather than individual files. Applications work on files in various ways; for example, some applications create temporary files in the same folder during editing. These temporary files might or might not be encrypted, and some applications substitute them for the original when the edit is saved. Encrypting at the folder level ensures that files do not get decrypted transparently in this way.
- Never rename or move the RSA folder. This is the only place EFS looks for private keys.
- In a domain, change the default recovery agent account (the Administrator of the first domain controller installed for the domain) as soon as possible, and set a password for each recovery agent account. This adds an extra layer of protection in case the Administrator account is compromised, and it provides easy tracking of use of the recovery account.
- Designate two or more recovery agent accounts per organizational unit (a subgroup of computers, or even a single computer, within a domain),

depending on the size of the organizational unit. Designate one computer for each designated recovery agent account, and give permission to appropriate administrators to use the recovery agent accounts.

- Implement a recovery agent archive program to ensure that EFS files can be recovered by means of obsolete recovery keys. Recovery certificates and private keys must be exported and stored in a controlled and secure manner. Store archives in a controlled-access vault and have both a master archive and a backup archive. The master archive is on-site; the backup archive is in a secure off-site location.
- Configure Syskey for stand-alone computers that are not members of a domain to provide startup key protection for the EFS users' private keys.
- Encourage users in stand-alone environments to create password reset disks in case they forget their passwords. This enables recovery of the master key for stand-alone computers.
- Use WebDAV folders to store encrypted files that must be transmitted across the network, especially public networks. This ensures that the data is encrypted during transmission.

Troubleshooting EFS

In some situations, EFS might not operate as expected. Some EFS issues can be resolved very simply, while others require complex high-level administrative solutions. Understanding how EFS works is the basis for resolving EFS issues. This section presents some common situations that might arise with EFS and the most likely causes for these problems.

For more information about troubleshooting EFS, see Windows XP Professional Help and Support Center.

Unable to Encrypt Files

If you find that you are unable to encrypt files or folders, one of the following might be the cause:

- The file is not an NTFS volume.
- You do not have Write access to the file.

- If you are having trouble encrypting a remote file, check to see that your user profile is available for EFS to use on that computer (which typically means having a roaming user profile), make sure the remote computer is trusted for delegation, and make sure your account is configured to enable delegation. Sensitive accounts are not enabled for delegation by default, so users like Enterprise Administrator might not be able to encrypt or decrypt files remotely.

  Note Sometimes users think that a file is not encrypted because they can open it and read the file. You can verify whether a file is encrypted by checking the file's attributes.

For more information about formatting volumes as NTFS, see Windows XP Professional Help and Support Center.

For more information about the encryption process, requirements, and procedures, see "Encrypting and Decrypting by Using EFS" earlier in this chapter.

For more information about remote EFS operations, see "Remote EFS Operations in a File Share Environment" earlier in this chapter.

Unable to Decrypt Remote Files

The following are the major causes of and solutions for remote decryption failure (usually indicated by an "Access is denied" message):

- The computer on which the encrypted file is stored is not trusted for delegation. Every computer that stores encrypted files for remote access must be trusted for delegation. To check a computer's delegation status, open the computer's properties sheet in the Active Directory Users and Computers snap-in.
- The user account that EFS needs to impersonate cannot be delegated. To check a user's delegation status, open the user's Properties sheet in the Active Directory Users and Computers snap-in.
- The user's profile is not available. Using roaming user profiles is the solution for this problem.

- One of the user's profiles is available, but it does not contain the correct private key. Using roaming user profiles is the solution for this problem.

For more information about the decryption process, requirements, and procedures, see "Encrypting and Decrypting by Using EFS" earlier in this chapter.

For more information about remote EFS operations, see "Remote EFS Operations in a File Share Environment" earlier in this chapter.

Unable to Open Encrypted Files

If you are unable to open an encrypted file, you might not have the correct EFS certificate and private key for the file. If the file is old, the public key and private key set might no longer be available. Although expired certificates and private keys are archived, users can delete archived certificates and private keys, or they can be damaged. If this has occurred, you can recover the file.

This problem can also occur when a computer that previously operated in stand-alone mode is now a member of a domain. The file might have been encrypted by a local self-signed certificate issued by the computer, but the CA designated at the domain level is now the issuing authority.

To access a file that was encrypted while the computer was in stand-alone mode, log off, and then log back on to the local computer instead of the domain.

The same conditions apply for encrypting and decrypting remote files. The user's profile must be available for EFS to use, the computer must be trusted for delegation, and the user's account must be enabled for delegation.

For information about how to recover files, see "Recovering Encrypted Files" earlier in this chapter.

For more information about encryption and decryption processes, requirements, and procedures, see "Encrypting and Decrypting by Using EFS" earlier in this chapter.

For more information about remote EFS operations, see "Remote EFS Operations in a File Share Environment" earlier in this chapter.

Determining Whether the Certificate Used to Encrypt a File Is Still Available for Decryption

The public key certificate is not used for decryption because it does not contain the private key. However, if a file cannot be decrypted, you can determine who is authorized to access the file and which certificate and public key were used to encrypt the file. This information is listed in the Encryption Details dialog box under a file's advanced properties. When you determine whether the user has been authorized to access the file, you can determine what certificate was used to encrypt the file. Then you can determine whether the certificate is still available. If the certificate is not available, the private key will not be available and the user will not be able to decrypt the file. If the certificate is available, the user might have exported the private key and deleted it from the computer.

To view authorized users and certificate thumbprint information

1. Right-click the encrypted file and then click Properties.
2. On the General tab, in the Attributes section, click Advanced.
3. In the Advanced Attributes dialog box, click Details to open Encryption Details. Users who are authorized to access the file are listed under Users Who Can Transparently Access This File. The thumbprint for the certificate used to encrypt the file is listed with the name of the authorized user.

   Note that any data recovery agents are listed as well.

To compare the certificate thumbprint associated with the encrypted file with other certificate thumbprints

1. Open the Certificates snap-in, and locate the user's certificates.
2. Double-click a certificate, and then click the Details tab. The certificate thumbprint is one of the listed details.
3. Compare the certificate thumbprint associated with the encrypted file with the thumbprints of each of the user's EFS certificates.

Determining Whether the Access Problem Is Related to the Availability of the Necessary Private Key

If the user who wants to access the file is not listed in Users Who Can Transparently Access This File, the user has not been authorized to access the file or has been deleted from the authorized user list. You can then determine whether the user is supposed to be authorized to access the file.

If the user is already authorized to access the file, you can compare the certificate thumbprint listed next to the users' name with thumbprints of the user's certificates in the Certificates snap-in.

- If the thumbprint listed in Users Who Can Transparently Access This File matches the thumbprint for the user's EFS certificate, an access problem probably results from either exporting the private key and deleting it from the system or from other access rights issues. Determine whether the user has exported and saved the private key. If so, the private key can be imported and access to the file is restored.
- If the thumbprint listed in Users Who Can Transparently Access This File does not match any thumbprints for the user's certificates, it is likely that the certificate has been deleted. This means that the private key has also been deleted. Determine whether the user exported the certificate and keys before deleting the certificate. If so, the certificate and keys can be imported to restore access to the file.

If the EFS certificate used to encrypt the file is available, determine whether the user has the necessary read and write permissions for the file. If the user does have read and write permission, it is likely that the private key was exported and deleted from the computer.

If the necessary private key is not available and cannot be imported, you can use the Encryption Details dialog box to determine whether any other users or any data recovery agents are authorized to access the file. If so, you can have any of these authorized users decrypt the file.

If the necessary private key is not available and cannot be imported, and there are no other authorized users or any data recovery agents, the file cannot be

recovered. Determine whether the user's profile is available for restore from backup.

Encrypted File Is Unencrypted When Copied or Moved

Encrypted files are decrypted when they are copied or moved to non-EFS capable volumes. If the user copies or moves files by using My Computer, the system provides a warning to the user if the destination volume is not EFS capable. If the user copies or moves files by using the copy /d command or the xcopy /g command, the files will be decrypted on the target volume with no warning that this has occurred.

For more information about changes to encryption status when files are copied or moved, see "Remote EFS Operations in a File Share Environment" earlier in this chapter.

Virus Check Program Cannot Check All Files

When your virus check program tries to check all the files on your hard disk, you get an "Access is denied" error message. Your virus check program can only read files that have been encrypted by you. If other users have encrypted files on your hard disk, access to these files is denied to the virus check program. To perform a virus check for files that have been encrypted by other users, the other users must log on and run the virus check program.

Common Error Messages

While performing EFS tasks, users might encounter error messages. The following are the most common messages that can occur and the possible causes for them.

Access is denied

You might receive an "Access is denied" message in one of the following situations:

- You do not have basic permissions to access the file (for example, the file is read-only).
- You are trying to encrypt a system file or folder. System files and folders cannot be encrypted.

- You are trying to share an encrypted file with someone, but you are not authorized.
- You are trying to decrypt a file that you did not encrypt and for which you have not been authorized.
- You are trying to decrypt a file that you encrypted or that you are authorized to access, but your private key (and probably your user profile) is not available.

The directory has been disabled for encryption

This message appears if a user tries to encrypt a folder (or files in the folder) in which a Desktop.ini file has been placed with encryption disabled.

The server is not trusted for remote encryption operation

This message might appear if the remote server that stores a file or folder you are attempting to encrypt or decrypt is not trusted for delegation.

The disk partition does not support file encryption

EFS cannot be enabled on a non-NTFS disk partition. To use EFS, the disk partition must be NTFS.

No valid key set

This message typically occurs in remote encryption or decryption operations. It means that EFS could not locate the correct keys for the operation. This is most likely to occur in remote decryption scenarios. EFS needs to locate your user profile and the private key associated with the public key used to encrypt the file's FEK. This error message might occur if the profile cannot be found, or if the profile can be found but does not contain the correct private key.

Additional Resources

These resources contain additional information and tools related to this chapter.

Related Information

- "Cryptography for Network and Information Security" in the *Distributed Systems Guide* of the *Microsoft Windows 2000 Server Resource Kit* for more information about cryptography

- “Windows 2000 Certificate Services and Public Key Infrastructure” in the *Distributed Systems Guide* of the *Microsoft Windows 2000 Server Resource Kit* for information about the public key infrastructure in Windows
- *Microsoft Internet Information Services 5.0 Resource Guide*, 2000, Redmond, WA: Microsoft Press, for more information about WebDAV
- The Microsoft Platform SDK link on the Web Resources page at http://www.microsoft.com/windows/reskits/webresources, for more information about CryptoAPI
- The Technet link on the Web resources page at http://www.microsoft.com/windows/reskits/webresources for more information about Data Protection API
- Article 223316, “Best practices for the Encrypting File System,” in the Microsoft Knowledge Base at http://support.microsoft.com

EFS, Credentials, and Private Keys from Certificates Are Unavailable After a Password Is Reset

Article ID 290260

SYMPTOMS

After you reset the password of an account on a Windows XP-based computer that is joined to a workgroup, you may lose access to the user's:

- Web page credentials.
- File share credentials.
- EFS-encrypted files.
- Certificates with private keys (SIGNED/ENCRYPTed e-mail).

CAUSE

This issue can occur if the password was forcefully reset by an administrator or owner, instead of being changed by the user.

RESOLUTION

NOTE: For any of the following resolutions to work, the user's original account must still exist, and the user's profile must be present and unchanged since the user last had access to the data.

To recover all of the data, you must have one of the following:

- The original password. This is the password with which the user last logged on successfully and was able to access their credentials and files.
- Password Recovery Disk (PRD). This password recovery disk must have been created while the user had access to the files.

To Completely Recover By Using the Original Password

1. Log on to the computer as the user with the current password.
2. Click Start, and then click Control Panel.
3. In Control Panel, click User Accounts.
4. Click your user name.
5. Click Change my password.

6. Follow the instructions to change the password back to your original password.
7. Restart your computer.

To Completely Recover By Using the Password Recovery Disk

1. If you are logged on, log off of the computer.
2. Attempt to log on as the user, and deliberately type an incorrect password.
3. Click use your password reset disk.
4. Follow the instructions in the wizard.
5. Log on, and note that you have access to your files.

Recovering Access to Encrypted EFS Data

If you have encrypted some of your files by using the Encrypting File System (EFS), you have additional options to recover access to those encrypted files. The following provisions apply only to EFS encrypted files, and will not recover access to saved credentials or certificates.

If you have previously exported the user's EFS private key from the user's account, you may import the key back into the account and recover access to the encrypted files.

If you did not export the private key and you have defined a Data Recovery Agent (DRA) prior to encrypting the files, you may regain access to EFS files as the Data Recovery Agent. For additional information about how to recover data in this case, click the article number below to view the article in the Microsoft Knowledge Base:

255742 Methods for Recovering Encrypted Data Files
If you do not have the required items or information specified for the preceding recovery solutions, the data is permanently encrypted, and cannot be recovered.

STATUS

This behavior is by design.

MORE INFORMATION

The behavior that is described in this article is a security measure taken to protect the security of the user's private information. A malicious administrator that can reset a user's password and thereby gain access to the user's account cannot access encrypted files or authentication materials without the user's knowledge or permissions.

Before being allowed to reset a password, an administrator or owner of the computer is prompted with the following messages:

- Resetting this password might cause irreversible loss of information for this user account. For security reasons, Windows protects certain information by making it impossible to access if the users password is reset.

The data loss will occur the next time the user logs off.

You should use this command only if a user has forgotten his or her password and does not have a password reset disk. If this user has created a password reset disk, then he or she should use that disk to set the password.

If the user knows the password and wants to change it, he or she should log in, then use the User Accounts in Control Panel to change the password.

- You are Resetting the password for *user name*. If you do this, *user name* will lose all EFS-encrypted files, personal certificate, and stored passwords for Web sites or network resources.

  To avoid losing data in the future, ask user2 to make a password reset floppy disk.

To avoid data loss because of a password reset in the future, create a password recovery disk to reset the password and have users change their own password while logged in.

To create a password recovery disk:

1. Click Start, and then click Control Panel.
2. Click User Accounts.
3. Click your user name.
4. Click Prevent a forgotten password, and then follow the instructions in the wizard.
5. Store the disk in a safe location.

NOTE: The Prevent a forgotten password button and the password recovery disk functionality are not available on computers that are joined to a domain.