Global Information Assurance Certification Paper

Windows Encryption – EFS or PGP?
Anthony D. Ombrellaro, Jr.
April 24, 2003
Version 1.4b

Abstract

In today's workplace companies are becoming increasingly concerned with the need to protect data from unauthorized access. Besides the need to protect data stored on their internal network, companies are requiring data stored on laptops and home computers be protected as well. With the continual increase in the number of telecommuters and laptops within originations, the need to protect the information contained on these computers becomes paramount.

Left unchecked companies expose themselves to a number of potential risks associated with the loss of "critical" information. These risks include the possible loss of revenue, fines, and the potential loss of business.

One of the most effective ways to protect this information is by the use of encryption. Until the release of NTFS 5, encryption on Microsoft operating systems was only achievable through third party encryption packages. Microsoft, recognizing the need for a standard encryption package within their operating systems released NTFS 5 with encryption built in. While Microsoft's encryption system offers the user the ability to encrypt information on their system it does present certain limitations that other encryption packages may be better suited to handle.

This document will examine the advantages and disadvantages to both Microsoft's encryption system EFS and a third party encryption package PGP, to determine which encryption package should be used to encrypt data within the Microsoft environment. While a number of third party encryption tools are available to encrypt data, only PGP will be compared in this study.

EFS

Microsoft's encryption technology EFS (Encrypting File System) was released as part of NTFS 5, and upgrade of NTFS, which was part of the Windows 2000™ release. With the release of Windows 2000™ and the subsequent releases of Windows XP™ and Windows 2003™, NTFS 5 has become the standard to encrypt data within the Microsoft environment.

Within the NTFS 5 file system, attributes have been added to allow for the encrypting of both files and folders. When encryption is enabled on a folder, all files contained in that folder will also be encrypted. It is important to note that EFS is not a file system itself, but a subsystem of NTFS 5.

How It Works

Like with most encrypting packages, EFS requires the use of keys to both encrypt and decrypt data. With EFS both public and symmetric key encryption is used to encrypt and decrypt files. By using these two methods, EFS provides a level of security that protects against all but the most sophisticated methods of attack.

Public key encryption (Asymmetric) utilizes a key pair to encrypt and decrypt data. This key pair consists of a public key, used to encrypt the data, and a private key used to decrypt the data. With symmetric key encryption only one key is used to both encrypt and decrypt the data.

One drawback with public key encryption is the amount of processing time required to encrypt the data. Because symmetric key encryption uses the same key to both encrypt and decrypt, it's usually 100 to 1,000 times faster then the public key method.

The first time a file is encrypted, the users public key pair that was issued to the user by the certification authorities (CA) will be assigned to the file. If no CA is available, EFS will issue its own certificate to be used in this process.

When a file is marked for encryption, each file is given it's own file encryption key (FEK), which is a unique random number generated by the operating system. This FEK is a shared secret key that is used to encrypt the file. The FEK used is then encrypted by the users public key and attached to the header of the file.

Using this method requires that two encryptions take place on each file, one for the actual data and the other for the FEK's. This method may seem redundant to some, but it actually lends itself to bulk encryptions making this extremely fast when dealing with large amounts of data to be encrypted. Currently, EFS uses only 128-bit DES to perform the encryption. At some point Microsoft may add other methods to actually perform the encryption.

To decrypt a file, EFS uses the users private key to decrypt the FEK, which is then used to decrypt the actual contents of the file. Because the users private key is needed to decrypt the FEK, encrypted files cannot be shared with anyone.

Certificates

As illustrated above, EFS relies on certificates to both encrypt and decrypt data. In the event a user losses their private key or a user leaves the company, access to any file that user had previously encrypted will be lost forever. To help guard against these types of loss, Microsoft suggests that each user export their private key for safekeeping, and that a recovery policy be implemented before

EFS encryption is enabled.

For standard Windows 2000 Certificate Services certificates, private key exporting is enabled only for EFS user and recovery agent certificates. When the private key is exported, the key will be stored in a password protected encrypted file. The password used will be one designated by the user performing the export. Whenever access to any exported key is required, the user will be required to enter the password used during the export, before the system will grant access to these keys.

The recovery policy provides the ability to decrypt files by someone other then the owner. When a recovery policy is configured, it can be configured to identify the user's who can recover encrypted data in addition to the user who originally encrypted the file. These users will be designated as recovery agents and will require a certificate be created for each recovery agent specified. These recovery agent certificates will be identical to the user certificates assigned for EFS, except the "Intended Purpose" will be set to "File Recovery" instead of "EFS". The administrator is designated as the default recovery agent.

Like with the users FEK, an FEK is created for each recovery agent and appended to the file. This method assures that any file encrypted can be decrypted by any of the authorized recovery agents as well as the user who actually encrypted the file.

When working with recovery agents, Microsoft suggest the following:

- Each Recovery agent should backup their certificate using the export command.
- Delete the recovery agent from the certificates manager after it has been backed up.
- Import the recovery agent only when needed and delete it immediately after use.
- Don't delete the last recovery agent; otherwise users will no longer be able to encrypt files.
- Make sure that all recovery agents are in place before encrypting any file.

Feature Set

With EFS, Microsoft has done a good job of integrating encryption into their Window's platform and has made encryption / decryptions transparent to the user. EFS allows users to work with encryption with no additional effort.

When using EFS, users should be aware of the following EFS policies:

- Only files and folders on NTFS5 volumes can be encrypted.
- Encrypted files will automatically be decrypted if they are copied or

moved to a volume that is not an NTFS5 volume.
- Compressed files and folders cannot be encrypted.
- Moving unencrypted files into an encrypted folder will automatically encrypt the files.
- Files marked with the "system" attribute or files located in the %systemroot% cannot be encrypted.
- Only the user who has encrypted the file can open the file
- Encrypted files cannot be shared
- Encrypting a folder or file does not protect against deletion. Anyone with delete permission can delete encrypted folders or files.
- Temporary files are automatically encrypted as long as they are created in an encrypted folder
- Recovery agent should be used so that the file can be decrypted if the user loses their encryption certificate.
- 128-bit DESX encryption used. (40-bit encryption used in the international versions).

While the concept of EFS encryption provides the user with a seamless encryption option, it does present some potential limitations that users should be aware of before they attempt to use it as a solution to all their encrypting needs.

Some of the problems presented by EFS encryption are:

- Single-user access. Files are only accessible to the user who encrypts them
- Heavy computer load on the server to handle encryption and decryption
- Reliance on user passwords to encrypt / decrypt files. If a users account is compromised, the "hacker" now has access to that uses encrypted files.
- 3DES encryption is not supported.
- Only the file contents are encrypted and not the file system structures. This allows everyone to see file names, sized, dates, etc. which in some cases may be a security breach.
- When access encrypted files over the network, the data is transmitted in clear text.

PGP

PGP (Pretty Good Privacy) is an encryption program created by Phil Zimmermann, a special director of Computer Professionals for Social Responsibility (CPSR) from 1997 – 2000, in the 1980's to promote awareness of the privacy issues associated with today's digital society.

PGP encryption allows users to encrypt email message and data files so that

they can be shared with other users in a secure fashion. Whereas EFS only allows users to encrypt data files, PGP plugins are available for a number of commercially available email systems.

Freeware versions of PGP are available for non-commercial use along with a commercial version that is available from PGP Corporation. In the past commercial versions of PGP were only available in the US and Canada, but PGP has since been granted an export license which has allowed it to be used in other countries.

Third party encryption packages like PGP allow users to work with one encryption product on multiple platforms. Unlike EFS, PGP can encrypt files on any file system and is not limited to systems running Windows 2000, XP or Windows 2003 with NTFS 5.

How It Works

PGP is a hybrid cryptosystem that combines some of the best features of both conventional and public key cryptography. Like with EFS, users are required to have a public key pair consisting of a private and public key before data can be encrypted.

When PGP encrypts data, it first compresses the data before starting the encryption process. Once the data is compressed, PGP will then encrypt the data using a session key to create the final encrypted file (ciphertext).

By compressing the data first, PGP reduces not only the disk space required to hold the encrypted file, but will reduce transmission time when transferring the file. In addition, compression provides an additional level of security helping to reduce the risk of possible hacks. File compression will only take place on files that meet a certain length requirement. Files that do not meet this length requirement will not be compressed before they are encrypted.

The session key is a one-time-only secret key that is used to encrypt the file. This session key is a random number generated from the users mouse movements and keystrokes, thus producing its uniqueness. After the file has been encrypted with the session key, PGP will then encrypt the session key with the recipient's public key. When the encrypted file is transmitted, both the ciphertext and the encrypted session key will be transmitted.

PGP decryption works in reverse. To decrypt a file, PGP will first decrypt the session key with the recipient's private key to recover the original session key. Once the session key has been decrypted, PGP will then use it to decrypt the ciphertext producing the original unencrypted file.

Like EFS, PGP allows users to encrypt individual files, but unlike EFS, PGP also

supports email encrypting and allows users to encrypt the entire contents of a disk. However the problem that PGP users have that is not presents in EFS is that users must first decrypt a file before they can work on it. Once the user has finished working with the unencrypted file, the user is forced to re-encrypt it once they are done working on it. EFS does the decryption and re-encryption automatically.

KEYS

Like EFS, PGP uses keys to both encrypt and decrypt data, but unlike EFS PGP allows users besides the person encrypting the data to decrypt it. This allows encrypted files to be shared between multiple users.

Like with any cryptograph system, keys are the key to PGP encryption. When a user installs PGP a key pair (public and private key) will be generated for that user. This key pair is critical to the encryption process. To share encrypted data with other users, users are required to share their public keys with the users they will be sharing the data with. To share public keys, PGP recommends that one of the following procedures be used:

- Make their public key available on a public certificate server
- Send it as part of an email to the intended user(s)
- Copy it to a text file for distribution.

To encrypt a file the sender will use the recipient's public key to encrypt the file. By using the recipient's public key in this process, the final encrypted file can be shared with that user. Once the recipient receives the file they will then use their private key to decrypt the file.

When using keys to encrypt data, the size of the key is important in determining how secure the data becomes, after the data has been encrypted. The bigger the key length, the more secure the data becomes. While using large keys will increase the security associated with a file, larger keys will also increase the amount of time needed to both encrypt and decrypt a file.

Because key sizes are directly related to how secure the data is and how long it take to encrypt and decrypt the data, choosing the right key length is critical to the overall performance of the encryption process. Unlike EFS who only support 128-bit DES, PGP supports up to 4096 bit encryption. Generally with PGP 1024-bit keys are used to provide an acceptable level of security and minimize the time needed to process the encryption / decryption process.

PGP keys are stored in two files on the users hard drive, one for the private keys and one for the public keys. When keys are stored in these files, called keyrings, they are stored in encrypted form preventing them from being easily compromised.

The private keyring contains all the users private keys and the public keyring contains the public keys of everyone the sender communicates with, in encrypted form. When a user transfers encrypted data to a new recipient, the new recipient's public key will be added to the sender's public keyring. If a user loses their private keyring they will be unable to decrypt any information encrypted with the keys stored in this keyring.

To help protect against someone using someone else's private key, PGP requires that a paraphrase be created and assigned to the private key when the key pair is created. The user uses this paraphrase whenever they need to access their private key to decrypt data sent to them. Remember, to decrypt a message a user must use their private key, and before a user can use their private key they must unlock it with the paraphrase associated with that key pair.

When a user creates a public key pair, PGP prompts the user for the paraphrase to be associated with this key pair. This paraphrase should be more then 8 characters long and a combination of uppercase and lowercase characters and numbers. The more complex the paraphrase is, the less likelihood the paraphrase can be cracked by hackers. However users should choose a paraphrase that is easy for them to remember since they will need to input the paraphrase whenever they wish to decrypt a message. PGP allows users to change a paraphrase at any time. By allowing users to change the paraphrase associated with a key pair, users do not have to generate new keys pairs if a paraphrase is compromised.

To increase the level of security associated with key pairs, PGP began supporting Split Key pairs with their release of version 6.0. Split Key Pairs allows both public and private keys to be split among multiple users.

Splitting these keys among different users provides another level of security when dealing with encryption. While this feature is not recommended since it requires that a private key be split among multiple users, some companies are utilizing this feature when dealing with Corporate Signing Keys. When split keys are used to encrypt / decrypt data, more then one or two users are required to supply a piece of the key to be able to reconstruct the key. If not enough users are available the key is unusable.

Another benefits associated with the type of public key cryptography used by PGP is that it provides a method for employing digital signatures. By using digital signatures, recipients of encrypted data can be assured of the origin of the data. With public key digital signatures, not only is the data's authenticity assured, but its integrity as well. In addition, digital signatures provide non-repudiation, meaning that a sender cannot claim that they did not actually send the data. Using digital signatures prevents against the likelihood of someone spoofing a users identity and sending data as if they were that person.

Key Security

As with EFS, PGP users should backup both their public and private keys. Since PGP keys are stored in keyrings on the users hard drive, users can backup these files to any media they wish. Both the public keyring (pubring.pkr) and the private keyring (secring.skr) are found in the "PGP Keyrings" subdirectory located under the main PGP directory located on the users hard drive.

One problem with keys is that users can forget their paraphrase or even lose their private key. When this happens it can lead to the loss of data. To help prevent against this with its release of PGP 8.0, PGP has incorporated Key Reconstruction to assist with the recovery of user keys. With Key Reconstruction, users are required to supply five questions and answers to be used to recover a key. Once the user has entered this information it is packaged and sent either to a PGP Keyserver or to a standard LDAP directory.

Once Key Reconstruction has been setup it can be used to reconstruct a users paraphrase if they forget it. If the user supplies at least 3 correct answers the entire key is reconstructed with a new paraphrase supplied by the user. Using this feature an entire key can be lost and still reconstructed.

Conclusion

Both EFS and PGP work as advertised and provide a user with different options to encrypt data. While EFS is only suited for the NTFS 5 environment, the fact that it is a subsystem of NTFS 5 makes it easy for any user to work with encryption.

The problem with EFS is that it's a single user encryption packaging, meaning that encrypted data cannot be shared. If there is a need to share encrypted data among different users, then EFS cannot be used. Plus the fact that EFS only provides 128-bit DESX makes it relatively week in comparison to other products.

PGP on the other hand is not restricted to NTFS 5, but is a third party product meaning that it is not as easy for the end user to use. In addition companies may need to purchase the corporate version of PGP causing them to incur additional costs that are not present when using EFS.

In analyzing the feature set of both it is impossible to definitively state one product is better then the other and should be used as the "Windows encryption standard". The choice in determining which package should be used should be left up to the individual company. My recommendation would be to use EFS if; file sharing is not required, only NTFS 5 is to be used, and there are budget concerns with purchasing addition software packages. Otherwise if you cannot live with any of these issues, then you should consider purchasing a third party

encryption package like PGP.

Some companies may choose to use both EFS and PGP to support all their encryption needs. While this would provide and "best of both worlds" encryption solution, it will also complicate the encryption process. Now instead of managing one encryption package, administrators will have two encryption packages to manager and users will have to keep track of which encryption method was used on each file.

Bibliography

The Encrypting File System: How Secure is it?
By: Howard Wright
URL: http://www.sans.org/rr/win2000/EFS_sec.php (2 Nov. 2001)

Windows EFS Pros and Cons
URL: http://www.networkcomputing.com/1121/1121ws1side1.html?ls-NCJS_1121bt (30 Oct 2000)

Hands On Microsoft Windows XP
Global Knowledge, MC6825C-001 (June 2002)

Windows 2000 Security
Global Knowledge, M6662C-001 (January 2001)

How EFS works
URL: http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/distrib/dsck_efs_wqpt.asp

Using Win2000's Foolproof Encryption
By: Howard Marks
URL: http://www.networkcomputing.com/1121/1121ws1.html?ls-NCJS_1121bt

Encrypting Files and Drives in Windows 95, 98, NT and 2000
By: Kurt Seifried
URL: http://www.seifried.org/security/cryptography/crypto-book/chapter-09.html

PGP Tutorial for Beginners to PGP
URL: http://www.pitt.edu/~poole/PGP.htm

How PGP Works
URL: http://www.pgpi.org/doc/pgpintro/

PGP Basics (by Shelly Brisbin)
Macworld July 2000

URL: http://www.macworld.com/2000/07/features/pgpbasics/

Inside PGP Key Reconstruction
A PGP Corporation White Paper (revision 4)
By: Will Price (August, 2002)
URL: http://www.pgp.com/products/whitepapers/PGPKeyRecon.pdf