ElcomSoft

Presentation on
Black Hat Europe 2003 Conference

Security Analysis of Microsoft Encrypting File System (EFS)

http://www.elcomsoft.com

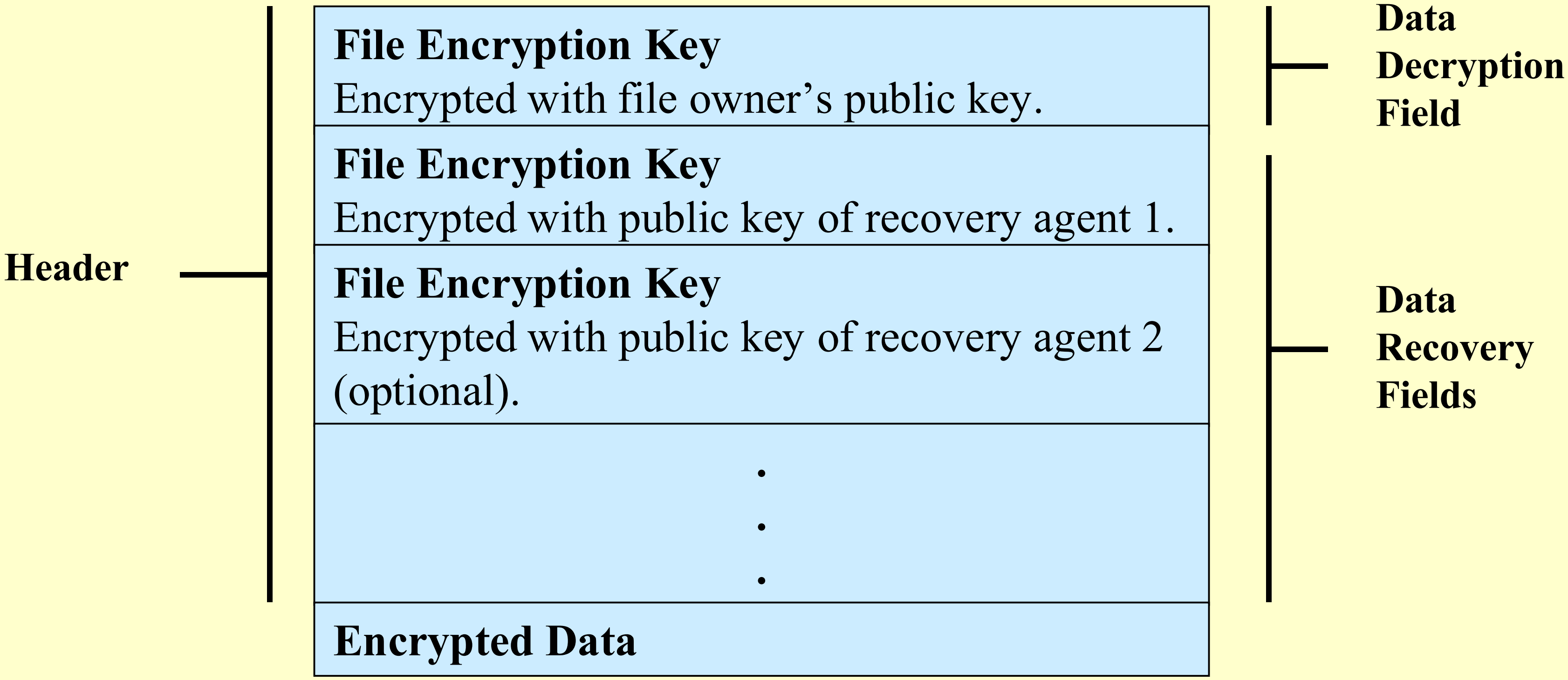
Microsoft Encrypting File System

Encrypting File System (EFS) is a new feature in Microsoft Windows 2000. EFS lets to protect sensitive data in files that are stored on disk using the NTFS file system. It uses symmetric key encryption in conjunction with public key technology to provide confidentiality for files. It runs as integrated system service, which makes EFS easy to manage, difficult to attack, and transparent to the file owner and to applications.

Even if the file can be stolen, over the network or physically, it cannot be decrypted without first logging on the network as the appropriate user. Since it cannot be read, it cannot be surreptitiously modified.

The file encryption key (FEK) — a symmetric bulk encryption key — is used to encrypt the file and is then itself encrypted by using the public key taken from the user's certificate, which is located in the user's profile. The encrypted FEK is stored with the encrypted file and is unique to it. To decrypt the FEK, EFS uses the encryptor's private key, which only the file encryptor has.

Structure of Encrypted File



Header

File Encryption Key
Encrypted with file owner's public key.

File Encryption Key
Encrypted with public key of recovery agent 1.

File Encryption Key
Encrypted with public key of recovery agent 2 (optional).

.
.
.

Encrypted Data

Data Decryption Field

Data Recovery Fields

ElcomSoft

http://www.elcomsoft.com

How EFS files are recovered

If the owner's private key is unavailable (for example, because it is damaged), a recovery agent account can open the file by using the private key for recovery, which is applied to the DRF to unlock the FEK. The mechanism for file recovery works essentially the same way as decrypting a file, by using the user's private key.

A private key for recovery cannot decrypt the DDF. If there are multiple recovery agent accounts, each private key for recovery decrypts only its own 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.

File decryption process

The following steps are needed to be accomplished to decrypt a file encrypted by Windows 2000:

1. Get the **System Key** from Registry, floppy disk or password.
2. Decrypt user's **password hash** stored in SAM Registry record.
3. Decrypt user's **Master Key**.
4. Decrypt user's **Private Key**.
5. Decrypt **File Encryption Key** of the file.
6. Decrypt **file data**.

System Key

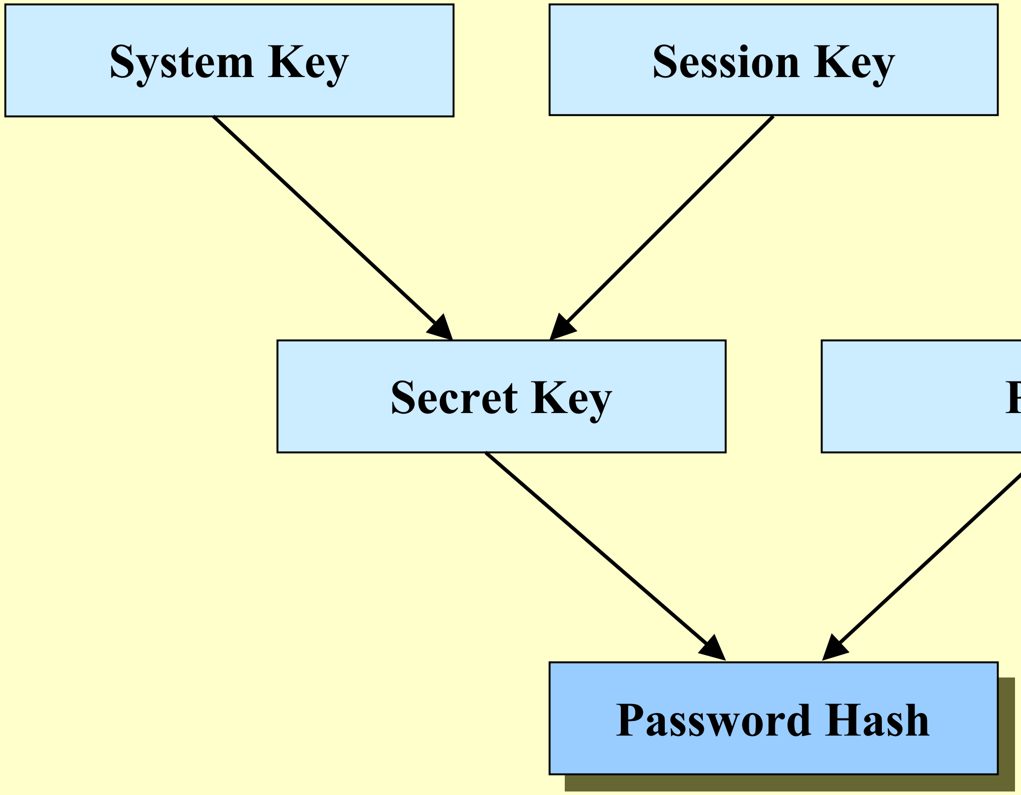
A **System Key** is used to protect the **SAM** (System Account Manager) record in the Registry. There are three ways to store a **System Key**:

1. System Key is stored in the Registry in obfuscated form. It's loading automatically when Windows starts.

2. System Key is stored on a floppy disk. This disk is needed to be inserted when Windows starts to unlock the SAM.

3. System Key is derived from a password entered by user on Windows startup. In this case System Key is not stored somewhere.

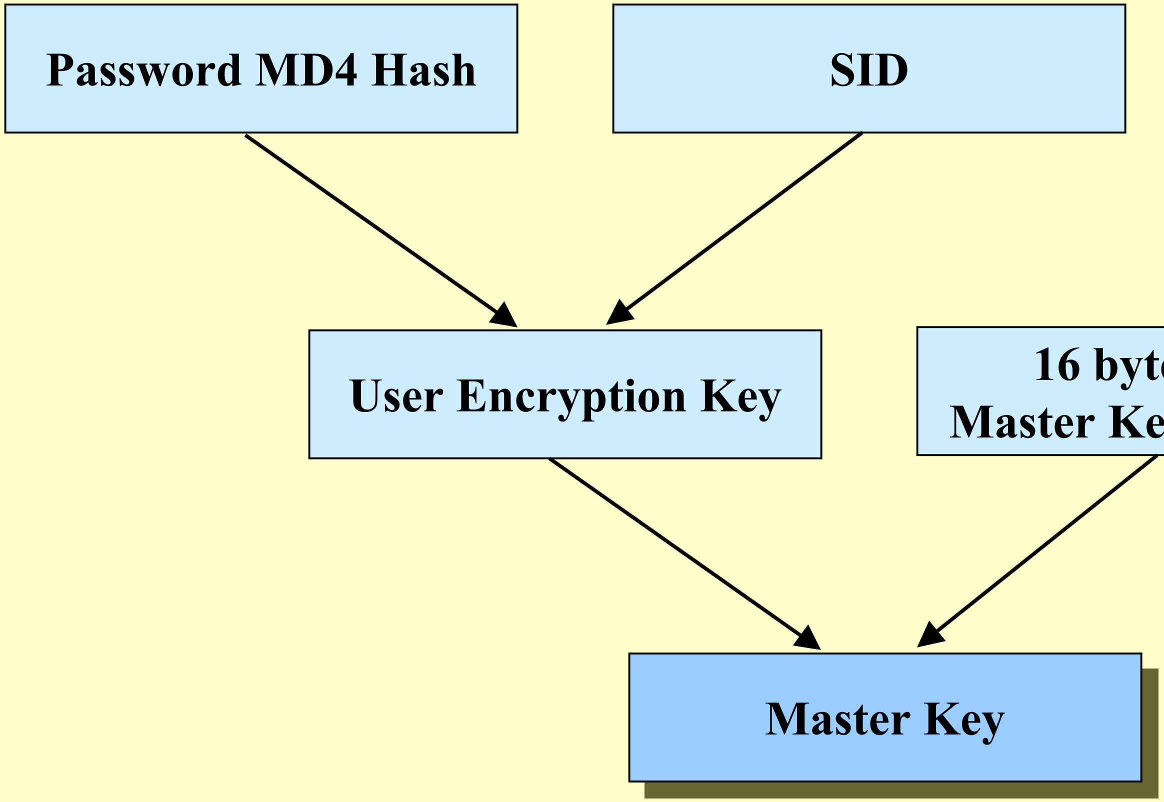
User password hash decryption process



ElcomSoft

http://www.elcomsoft.com

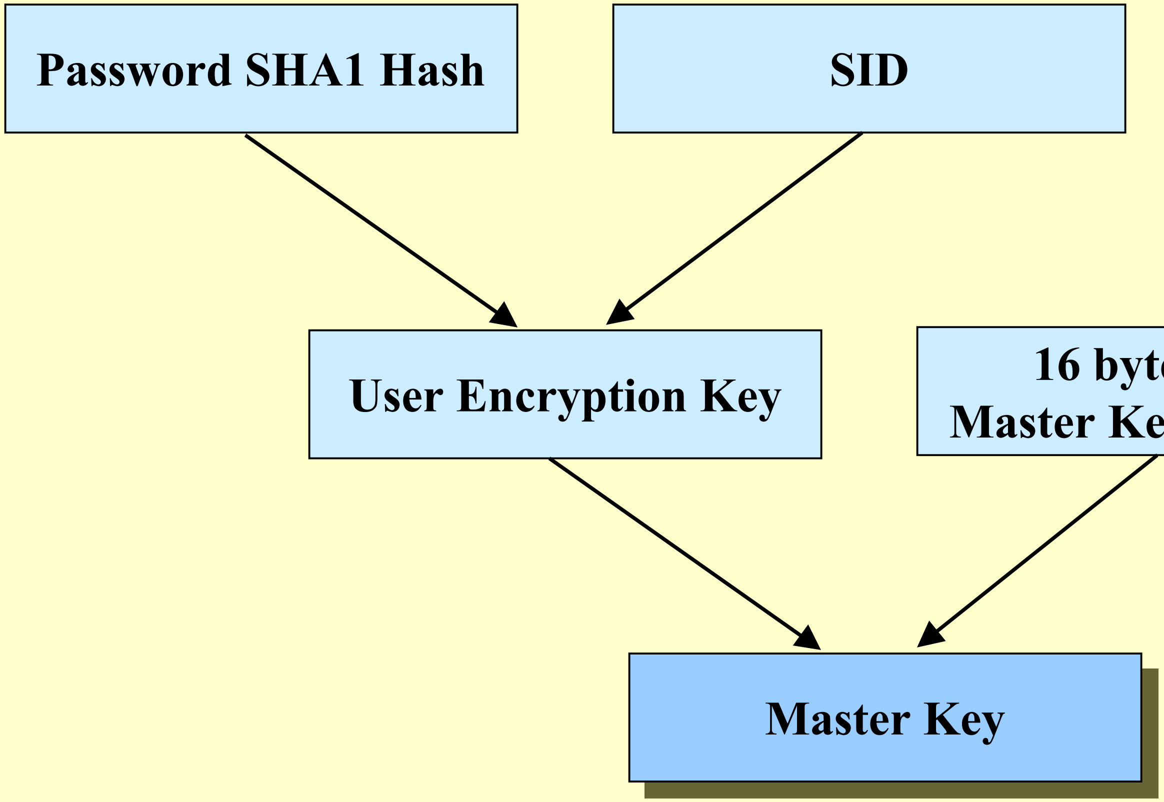
Master Key decryption process (Windows 2000)



ElcomSoft

http://www.elcomsoft.com

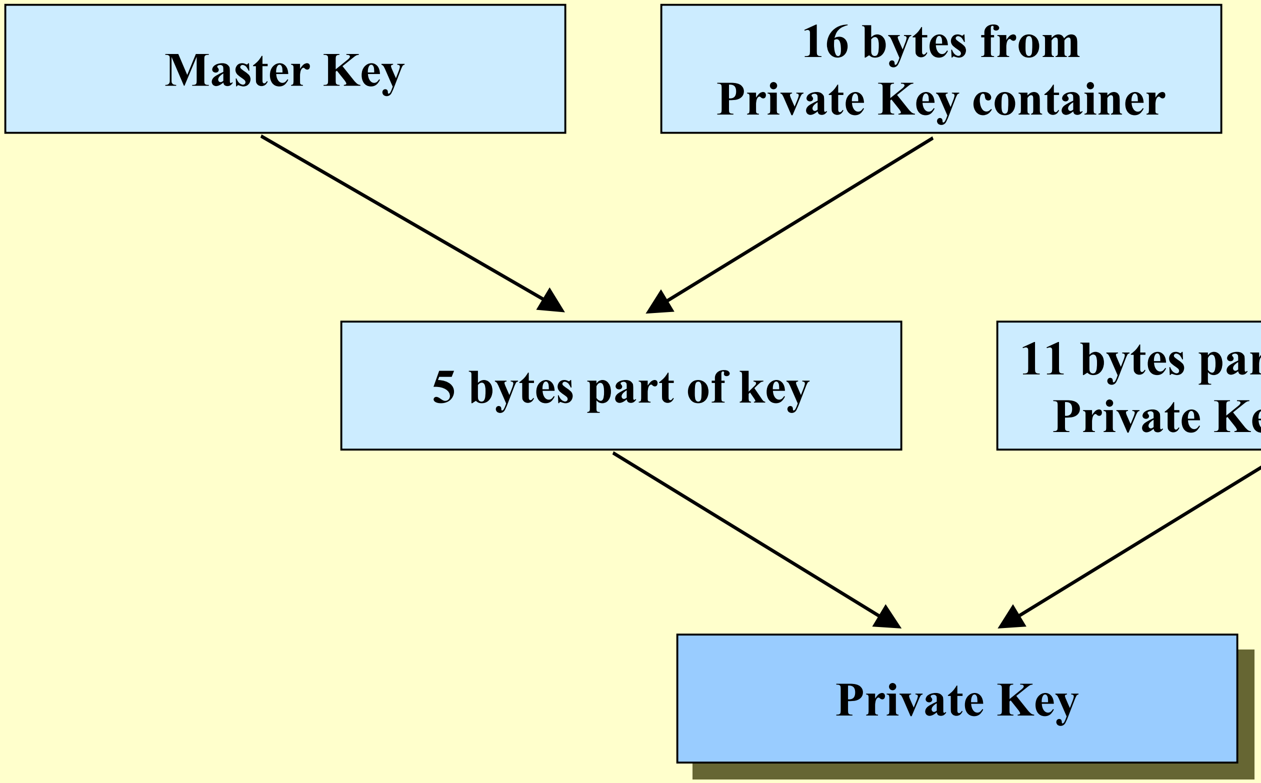
Master Key decryption process (Windows XP)



ElcomSoft

http://www.elcomsoft.com

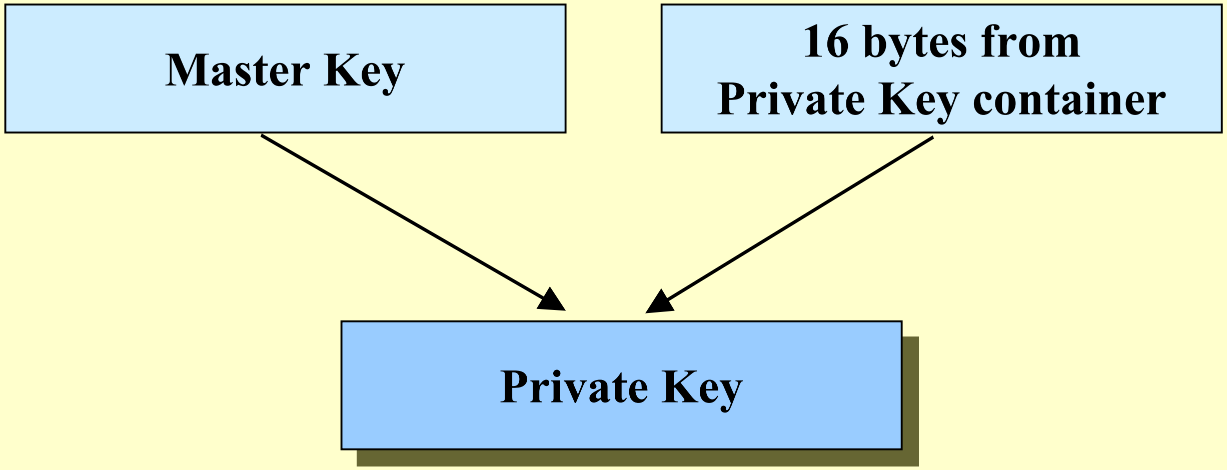
Private Key decryption process
(the original version of Windows 2000)


Master Key
16 bytes from Private Key container
5 bytes part of key
11 bytes part of key from Private Key container
Private Key

ElcomSoft

http://www.elcomsoft.com

Private Key decryption process
(version of Windows 2000 with Service Pack)



Master Key

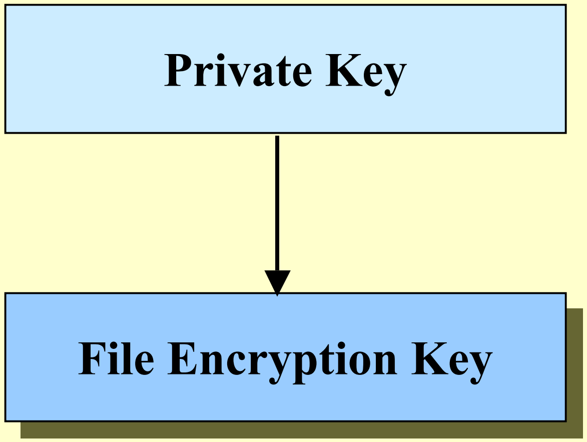
16 bytes from
Private Key container

Private Key

ElcomSoft

http://www.elcomsoft.com

FEK encryption/decryption process



Small cryptanalysis of EFS encryption

Generally algorithms used in the EFS are cryptographically stable when the key length is enough to make a brute-force attack impossible. Only one exclusion is an RC4 algorithm used to encrypt the Private Key in the original version of Windows 2000. In this case the key length is 128 bits but only 40 bits are dependent from the Master Key. Other 88 bits are well-known and stored in the Private Key Container. Therefore in the original version of Windows 2000 each Private Key is protected by 40-bits key which can be found by a brute-force attack during the reasonable time. After that the File Encryption Key can be found instantly and therefore it’s possible to decrypt the file. This vulnerability is fixed in the Windows 2000 Service Pack 2. There is strong 160-bit Triple DES encryption of a Private Key. Windows XP don’t have this vulnerability at all because EFS uses SHA1 password hash instead of MD4 hash to decrypt a Master Key. SHA1 hash is not stored on the physical disk so it cannot be stolen by an unauthorized user.

Possible attacks to decrypt EFS-encrypted files

Although the algorithms used in the EFS are cryptographically stable there are several attacks which can be used to decrypt files:

1. There are files which contain Private and Master Keys, SAM and System records of the Registry. Any user passwords are unknown. For example we have a hard disk with installed Windows 2000 and we don’t know user passwords as well as Administrator password.

   ↓

   All files can be decrypted instantly.

2. There are files which contain Private and Master Keys.

   ↓

   We need to find a user password and then decrypt the Private Key.

3. There is a Private Key container only.

   ↓

   We can brute-force the 40-bits RC4 key to decrypt the Private Key.

ElcomSoft

http://www.elcomsoft.com

Recovery possibilities

Using Private and Master Key containers as well as SAM and System Registry records allow to recover a data in cases when even a user or recovery agent certificate is lost. When these files are accessible we can decrypt any EFS-encrypted file of any user. EFS in Windows 2000 does not allow to secure the files when an intruder have an access to these files but at the same time there are good recovery possibilities in a cases when recovery certificates are lost.

Advanced EFS Data Recovery

Advanced EFS Data Recovery is a program to recover (decrypt) files encrypted on NTFS (EFS) partitions created in Windows 2000. Files are being decrypted even in a case when the system is not bootable and so you cannot log on, and/or some encryption keys (private or master) have been tampered. Besides, decryption is possible even when Windows is protected using **SYSKEY**. AEFSDR effectively (and instantly) decrypts the files protected under all versions of Windows 2000 (including Service Packs 1, 2 and 3).

ElcomSoft

http://www.elcomsoft.com