Confronting Encryption in Computer Investigations

By

Eoghan Casey

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2002 USA

Syracuse, NY (Aug 6th - 9th)

Practical Approaches to Recovering Encrypted Digital Evidence

Eoghan Casey, MA

> *The threat [of encryption] is manifest in four ways: failure to get evidence needed for convictions, failure to get intelligence vital to criminal investigations, failure to avert catastrophic or harmful attacks, and failure to get foreign intelligence vital to national security. Encryption can also delay investigations, increase their costs, and necessitate the use of investigative methods which are more dangerous or invasive of privacy.* (Demming, Baugh, 1997a)

Abstract

As more criminals use encryption to conceal incriminating evidence, forensic examiners require practical methods for recovering some or all of the encrypted data. This paper presents lessons learned from investigations involving encryption in various contexts. By presenting successful and unsuccessful case examples, this paper gives forensic examiners a clearer understanding of the feasibility and limitations of various approaches to dealing with encryption. Additionally, by demonstrating how encryption has been successfully dealt with in past investigations, this paper provides examiners with techniques that we can apply in our work and encourages us to aggressively confront encryption.

1.0 Introduction

As criminals become more aware of the capabilities of forensic examiners to recover digital evidence they are making more use of encryption technology to conceal incriminating data. Organized criminals use readily available encryption software (United States v. Scarfo) and online child pornographers encrypt their communications and the files they exchange (McAuliffe, 2001). Terrorist groups such as Al Qaeda are making use of encryption to protect the contents of their computers and their Internet communications (Kelley, 2002). Also, the Web site of the Earth Liberation Front and encourages members to use encryption.[i]

Since criminals generally encrypt the more incriminating communications and stored data, it is often exactly this evidence that investigators seek. Therefore, in addition to an understanding of cryptography, it is critical for forensic examiners to develop practical techniques for dealing with encryption to obtain some, if not all, of this digital evidence. This paper presents lessons learned from investigations involving encryption in various contexts. As strong encryption becomes more widely used by criminals, it is infeasible to attack the encryption directly using brute force methods. Instead, practical approaches to recovering encrypted data using readily available tools are discussed such as locating unencrypted copies of data, obtaining encryption passphrases, and guessing encryption passphrases. Legal challenges that arise when dealing with encryption are discussed and directions for future tool development are proposed.

2.0 Overcoming Weak Encryption

Computer intruders often use simple encryption to obfuscate network traffic and portions of rootkits they install on compromised systems to conceal their presence. One common form of simple encryption used by intruders is to exclusive OR (XOR) each byte against the value 255 (0xFF), effectively inverting every byte in the file. In one case, examiners found a configuration file that they suspected was a key component of a rootkit but appeared to contain only binary data. Viewing the file using a hexadecimal viewer showed that all of the characters in the file were above decimal value 127. This absence of ASCII characters suggested some form of character substitution. Guessing that XOR was used, the examiners reversed the encryption to reveal the contents of the rootkit configuration file shown here:

```
# perl -e 'while (<>) { print ~$_; }' < uconf.inv
[file]
find=/usr/lib/boot/find
du=/usr/lib/boot/du
ls=/usr/lib/boot/ls
file_filters=01,lblibps.so,sn.l,prom

[ps]
ps=/usr/lib/boot/psr
ps_filters=ibmd,drone,psniff,psr
lsof_filters=ibmd,uconf.inv,psniff,psr,:13000,/dev/pts/01,sn.l,prom,lsof

[netstat]
netstat=/usr/lib/boot/netstat
net_filters=38290,25000,6667,8000,9000

[login]
su_pass=owned
su_loc=/usr/lib/boot/su
ping=/usr/lib/boot/ping
passwd=/usr/lib/boot/passwd
shell=/bin/sh
```

As another example, Back Orifice uses XOR to encrypt traffic between the client and server and it is relatively simple to decrypt these packets since the header of Back Orifice

packets always begins with the same string. The Back Orifice detector preprocessor in Snort can be configured to brute force the key space and decrypt the traffic.

Many applications, including early versions of Microsoft Word and Excel, use XOR to encrypt passwords that individuals can select to protect their files (Microsoft, 2001).[ii] These passwords are easily recovered using tools such as Access Data's Password Recovery Toolkit[iii] and NTI's Advanced Password Recovery Software Tool Kit.[iv] PalmOS provides another example of XOR. The user-selected password in PalmOS prior to version 4 is XOR-ed against a constant mask, allowing anyone who knows the mask to decrypt the password (Kingpin, 2000). The encrypted password can be extracted from Unsaved Preferences on a Palm device[v] or the users.dat file in Palm Desktop and decrypted using palmcrypt as shown here:

```
D:\>palmcrypt -d B8791D707A2359435082DA4E599FBE4BEE675CCE541B346C041B6C55AE81CDF

PalmOS Password Codec
kingpin@atstake.com
@stake Research Labs
http://www.atstake.com/research
August 2000

0x62 0x69 0x72 0x74 0x68 0x64 0x61 0x79  [birthday]
```

Substitution ciphers are another popular form of weak encryption that are relatively easy to reverse but can absorb a significant amount of time. Obviously, such simple encryption schemes are less appealing to criminals when they want to conceal incriminating evidence of more serious crimes.

3.0 Strong Encryption: The Cost of Brute Force Attacks

Secret and public key encryption schemes provide offenders with a higher degree of protection, making it more difficult for forensic examiners to access evidence. When an encryption algorithm like DES is used, it is theoretically possible to try every possible key to decrypt a given piece of ciphertext. However, this approach requires significant computing power to run through the vast number (2^56, over 72 quadrillion) of potential decryption keys and can take an inordinate amount of time depending on the strength of the encryption.

There are some inexpensive solutions for brute forcing 40-bit encryption in certain file types. For instance, Access Data's Distributed Network Attack (DNA) application can brute force Adobe Acrobat and Microsoft Word/Excel files that are encrypted with 40-bit encryption as shown in Figure 1. Beowulf clusters[vi] are another inexpensive option for brute force attacks, making use of readily available computer equipment to create powerful parallel processing.

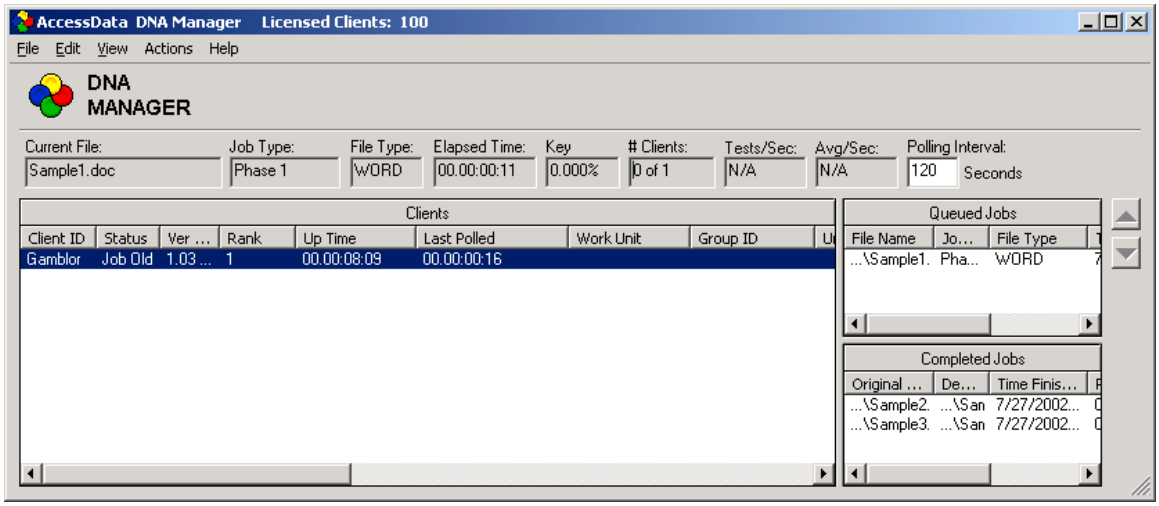

Figure 1: Decrypting MS Word files using Distributed Network Attack (DNA)

So, anyone with a cluster of approximately 100 off-the-shelf desktop computers and the necessary software can attempt every possible 40-bit key in 5 days. For example, the Wall Street Journal was able to decrypt files found on an Al Qaeda computer that were encrypted using the 40-bit export version of Windows NT Encrypting File System (UK Usborne, 2002).

However, Microsoft Windows EFS generally uses 128-bit keys and because each additional bit doubles the number of possibilities to try, a brute force search quickly becomes too expensive for most organizations or simply infeasible, taking million of years.[vii] Therefore, before brute force methods are attempted, some exploration should be performed to determine if the files contain valuable evidence and if the evidence can be obtained in any other way.

4.0 Practical Approaches to Recovering Encrypted Files

In theory strong encryption can create an insurmountable challenge for forensic examiners. In practice, encryption applications have weaknesses that can be exploited to recover some or all encrypted data. Additionally, general human use of encryption software introduces weaknesses such as selecting weak passphrases or writing strong passphrases down.

The crypt utility on Unix machines clearly demonstrates some of the weaknesses of relatively strong, secret key encryption.[viii]

```
% crypt -key 'birthday' < plaintext> ciphertext
```

One obvious weakness is the plaintext file. If the plaintext file is simply deleted rather than wiped, it may be possible to recover this copy from the hard disk. Furthermore, if the plaintext file was stored in memory, swapped to disk, or backed up to external media, it may be possible to retrieve some or all of this data.

Another obvious weakness in the above example is the secret key. If an easy to remember key such as "birthday" is used, it may be possible for someone to guess it and gain access to the encrypted data. If a difficult to remember key is used, it may be necessary for the user to write it down in a location that can be referenced the next time the data is decrypted, potentially exposing it to others. Additionally, it is possible for someone to observe the secret key as it is typed into the system or by capturing keystrokes or simply looking over the user's shoulder. It may be possible for an attorney or judge to convince or compel a suspect to disclose the secret key.

Public key algorithms like Pretty Good Privacy (PGP)[ix], S/MIME, and SSL have similar weaknesses that will be discussed in more detail in the following subsections. For instance, Ramsey Yousef, a major participant in several terrorist attacks including the 1993 bombing of the World Trade Center, stored information about planned attacks on his laptop in encrypted form but investigators were able to recover plaintext versions (Demming, Baugh, 1997b). Yousef's encryption passphrase was also obtained, enabling investigators to verify that the plaintext files matched the encrypted versions.

4.1 Finding Unencrypted Copies of Data

At some point before data is encrypted, it exists in unencrypted form. For example, while a file is being encrypted using EFS a temporary copy of the plaintext is made in case a problem is encountered during the encryption process. Also, the plaintext might be stored temporarily in a paging file (pagefile.sys) prior to encryption. If data was decrypted and re-encrypted for any purpose, it may have been stored temporarily on disk. For instance, if an EFS encrypted file is printed and the System32\Spool\Printers folder is not encrypted, spool files will contain unencrypted copies of the encrypted files.

In one case the offender used PGP to encrypt Microsoft Word documents. Although the original documents were wiped, fragments of the files were scattered around the disk in deleted MS Word temporary files, some of which could be found by searching for Microsoft Word headers.[x] The fragments that were located in this initial search contained metadata similar to the following:

.S.U.S.P.E.C.T...N.A.M.E...C.:.\.S.E.C.R.E.T.\.P.R.I.V.A.T.E...d.o.c...p.g.p...S.U.S.P.E.C.T...N.A.M.E.

Searching the disk for this and other similar metadata, the examiner found many additional fragments that were not retrieved in the initial search, the most interesting of which contained dates in a particular format (e.g, 1./.2.9./.0.1). Another search for occurrences dates in the same format revealed a large number of additional fragments.

Although it may not be possible to confirm that the recovered evidence is identical to an encrypted file, this may not be necessary once the incriminating evidence is in hand. This

issue becomes less clear when unencrypted versions of data are obtained by searching a database of known files for characteristics of pre-encrypted files such as names and original file sizes. For instance, in United States v. Hersh:

> *... encrypted files found on a high-capacity Zip disk. The images on the Zip disk had been encrypted by software known as F-Secure, which was found on Hersh's computer. When agents could not break the encryption code, they obtained a partial source code from the manufacturer that allowed them to interpret information on the file print outs. The Zip disk contained 1,090 computer files, each identified in the directory by a unique file name, such as "sfuckmo2," "naked31," "boydoggy," "dvsex01, dvsex02, dvsex03," etc., that was consistent with names of child pornography files. The list of encrypted files was compared with a government database of child pornography. Agents compared the 1,090 files on Hersh's Zip disk with the database and matched 120 file names. Twenty-two of those had the same number of pre-encryption computer bytes as the pre-encrypted version of the files on Hersh's Zip disk.* (Unites States v. Hersh)

This methodology is open to criticism given the probability of two different files having the same name and size. Finding 22 matches increases the chances that the encrypted files contain child pornography but such assumptions are difficult to verify without decrypting the files.

Another source of unencrypted data is in RAM. For instance, if the contents of an application window (such as Outlook's e-mail composition window) is encrypted using PGP, a copy of the plaintext is often held in memory by the application. Similarly, when PGP is used to encrypt or decrypt text on Windows 2000, a copy of the plaintext is held in memory by PGPtray for an indefinite period. The memory of this process can be dumped to a file using a program like pmdump[xi] and searched for unencrypted data as shown here:

```
D:\>pslist pgptray
Name         Pid    Pri Thd  Hnd    Mem     User Time     Kernel Time    Elapsed Time
PGPtray      1332   8   7    150    1264    0:00:00.060   0:00:00.270    2:20:33.466
D:\>pmdump 1332 pgptray.mem
D:\>less pgptray.mem
…
╫^@^@^@^@^@^@^@^@└^@^V^@^@^@^P^@└╧^V^@`Ç^V^@P Signature Status: good
*** Signer: Eoghan Casey <eco@corpus-delicti.com>
*** Signed: 7/20/2002 8:36:42 PM
*** Verified: 7/20/2002 8:41:17 PM
*** BEGIN PGP DECRYPTED/VERIFIED MESSAGE ***

Return-Path: <harold1jones@go.com>
Received: from webmailmta.go.com ([204.202.140.199])
     by lsh110.siteprotect.com (8.9.3/8.9.3) with ESMTP id SAA04960
     for <eco@corpus-delicti.com>; Thu, 11 Jul 2002 18:57:48 -0500
Received: from gomailjtp03 ([10.212.0.163])
     by mta07.seamail.go.com (Sun Internet Mail Server
     sims.4.0.2001.07.26.11.50.p9) with ESMTP id
     <0GZ3002K5Z8ZO3@mta07.seamail.go.com> for
eco@corpus-delicti.com; Thu,
     11 Jul 2002 16:43:48 -0700 (PDT)
Date: Thu, 11 Jul 2002 16:45:32 -0700 (PDT)
From: Harold Jones <harold1jones@go.com>
Subject: Test
To: eco@corpus-delicti.com
Message-ID: <6477825.1026431132801.JavaMail.harold1jones@gomailjtp03>
MIME-version: 1.0
X-Mailer: GoMail 3.0.0
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: 7BIT
X-Mozilla-Status: 8001
X-Mozilla-Status2: 00000000
X-UIDL: WBZAgrMiFHAeVlSFpWCqRCEs

Testing

___________________________________________________
GO.com Mail
Get Your Free, Private E-mail at http://mail.go.com


*** END PGP DECRYPTED/VERIFIED MESSAGE ***
^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^L^@<8D>
…
```

However, forensic examiners rarely have an opportunity to extract information from RAM before the system is rebooted. Even if such access were available, the examiner would have to operate the computer, altering its state and potentially erasing valuable information. In practicality, memory dumps are most useful when they occur accidentally as described in the next section. Also, as analysis tools evolve, it may be feasible to extract information from RAM during postmortem analysis (see Section 6.0: Future Considerations).

*4.2 Obtaining Encryption Passphrase*s

Another practical approach to gaining access to encrypted data is to obtain the passphrase that protects the private key. Investigators may be able to obtain the decryption passphrase by searching the area surrounding a system for slips of paper containing the passphrase[xii], interviewing the suspect, or surreptitiously monitoring the suspect's computer use. Also, if it is possible to obtain passwords that the suspect uses to protect other personal data such as e-mail and personal digital assistants, these passwords should be tried since people often use the same password for multiple purposes.

As noted in the previous section, accidental memory dumps may disclose information relating to encryption. For instance, when PGP crashes on Windows 2000, the Dr. Watson application creates a memory dump (e.g. C:\Documents and Settings\All Users\Documents\DrWatson\user.dmp) that can contain encrypted and plaintext versions of data as well as passphrases as shown here:

```
C:\Documents and Settings\All Users\Documents\DrWatson>strings user.dmp
```

```
…
kernel32.dll
RASAPI32
C:\WINNT\tracing
C:\Documents and Settings\Administrator\My Documents\PGP\pubring.pkr
C:\Documents and Settings\Administrator\My Documents\PGP\secring.skr
 &   !
IN PGP MESSAGE-----
Version: PGP 7.1
qANQR1DBwU4DSL6Q3OHRwOYQB/9pKnnhZGQRFwykWzBO1EWkzW336QOkUaHj0aVj
P1MgxDWQWi3kZpOfGnDg6kbQriWBiIgD/z8p5xGN+WcksytlLJv8OxvTGMepx7u8
h5aVRXZd8YPM+h5ROpbnNw+SiT/w9oCy/ChWeiCHV1swQSzwBHx2Ye+yxO70Moxc
…
frAG3nM7kOnChQp4jxhv2J0p7fL1vteI9EGbcimC9QCVBwC1U++mQIqbTyIw5gWK
Io11yl8P+wKjcHsLfi2hTE+NIRb+VORWhVoCDHgNKV1nSFNTK0LEnvz84OFyRc1z
-----END PGP MESSAGE-----
<pgppassphrase!>
…
```

Since there may be other ways in which encryption applications expose passphrases (e.g., in swap files), a more systematic method of searching the disk for possible passphases is desirable. For instance, using the Forensic Toolkit (FTK) from Access Data it is possible to generate a list of keywords found on the disk and import this list into the Password Recovery Toolkit (PRTK) as shown in Figure 2.[xiii] In this way, if the user purposefully or unintentionally stored their passphrase on disk or an application wrote the passphrase to disk, it will be available in the keyword list. In one case, the PGP passphrase was found on an apparently blank floppy disk.

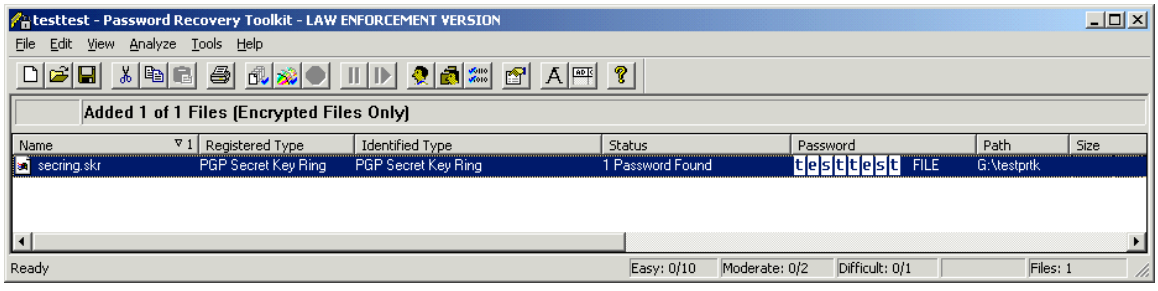

Figure 2: A PGP passphrase identified by PRTK from a list of keywords harvested from hard drive.

Privacy concerns related to overly broad searches are a potential problem when scouring a disk for keywords to be used in passphrase guessing. Overly broad searches of computers have been used to exclude evidence (Carey, 2000). However, generating an exhaustive list of keywords from all available media has a specific purpose and is distinct from an exploratory search of media for incriminating evidence.

If the passphrase cannot be obtained from the computer, a good attorney with the assistance of a forensic examiner may be able to persuade a suspect to cooperate. In one case, the suspect claimed that, in a fit of paranoia, he had changed his passphrase to something he could not recall just prior to being apprehended. To verify this claim, the forensic examiner checked the modification time of the file containing the private key and found that the passphrase had never been changed. When the prosecuting attorney presented this proof to the suspect and explained that any further lack of cooperation would make matters worse, the suspect provided the correct passphrase.

Although refusing to disclose an encryption passphrase does not necessarily imply guilt, it does reflect badly on the suspect in court and can shift the burden of proof onto the defense. Faced with such a risk, offenders can be persuaded to decrypt data in exchange for leniency in sentencing or plea bargain. However, 5th Amendment concerns arise when a court requires an uncooperative suspect to provide information such as a PGP passphrase since it could be construed as self-incrimination. Although knowledge of a PGP passphrase is highly incriminating, it can be argued that another person could have

encrypted and/or signed data with a copy of the key. Also, immunity can be arranged under certain conditions.

> *In People v. Price in Yolo County, California Superior Court prosecutors successfully compelled production of the passphrase protecting the defendant's PGP key. In this case, however, the key was not sought for the purpose of acquiring evidence for conviction, but rather to determine whether the defendant's computer should be released from police custody. He had already been convicted of annoying children and wanted his computer back. The police argued it should not be released as there was reason to believe it contained contraband, specifically PGP-encrypted files containing child pornography… The defendant was unsuccessful in arguing a 5th Amendment privilege. The prosecution argued that the contents of the file had already been uttered and, therefore, were not protected under the 5th Amendment. As long as prosecutors did not try to tie the defendant to the file by virtue of his knowing the passphrase, no incrimination was implied by disclosing the passphrase.* (Demming, Baugh, 2000)

As a last resort, the suspect's machine can be monitored using software or hardware in an effort to obtain the desired passphrase. Commercial software programs like Spector Pro[xiv] and free programs such as SubSeven and Back Orifice enable key logging, screen captures, and remote file access, enabling investigators to obtain encrypted files remotely.

Hardware devices like KeyGhost[xv] and KeyKatcher[xvi] have internal memory and record keystrokes when they are connected between the keyboard and the CPU.

The advantage of hardware key logging devices over software is that they record every keystroke, even those used to protect the BIOS. However, these devices are not designed for Macintosh or Sun systems and do not work on laptops or personal digital assistants since the keyboard is integrated. Also, physical access to the machine is required to install and retrieve hardware devices and they are visible to the alert user. A tamper-evident seal can be attached to the device, making it more difficult for an individual to remove the key logger without some visible sign of tampering as noted by KeyKatch.[xvii]

Monitoring software and key logging are invasive and can raise privacy issues. For instance, in United States v. Scarfo, the defense argued that the FBI violated wiretap statutes when they installed a key logger system on the Scarfo's computer. Although full details of the monitoring system are protected under the Classified Information Procedures Act, court records indicate that the system only captured keystrokes while the computer was not connected to the Internet via the modem. This explanation satisfied the court during an *in camera, ex parte hearing* but most key loggers do not function in this manner and this technique is of limited effect when a computer is continuously connected to the Internet or when the suspect writes e-mail offline and only connects to the Internet to send the messages. The court addressed this concern by comparing key logging to searching a closet or file cabinet.

> *That the KLS (Key Logging System) certainly recorded keystrokes typed into Scarfo's keyboard other than the searched-for passphrase is of no consequence. This does not, as Scarfo argues, convert the limited search for the passphrase into a general exploratory search. During many lawful searches, police officers may not know the exact nature of the incriminating evidence sought until they stumble upon it. Just like searches for incriminating documents in a closet or file cabinet, it is true that during a search for a passphrase "some innocuous [items] will be at least cursorily perused in order to determine whether they are among those [items] to be seized."* (United States v. Scarfo)

4.3 Passphrase Guessing

It is sometimes feasible to attempt to guess the passphrase used to protect a private key manually. For instance, when a small number of passphrases are likely, it may be feasible to type in all permutations. If this approach is taken, it is important to document the process both to avoid entering a passphrase more than once and to show attorneys, jurists, and other examiners which passphrases were attempted.

When manual passphrase guessing fails, an automated approach may be used with a list of common passphrases, then a dictionary in the language(s) of the suspect, and then more sophisticated permutation techniques. For instance, the Password Recovery Toolkit (PRTK) from Access Data can be configured to use various dictionaries and customized suspect profiles as shown in Figure 3. The PRTK then generates possible passphrases

using entries in the dictionary, suspect profile, and various combinations of these strings as depicted in Figure 4.

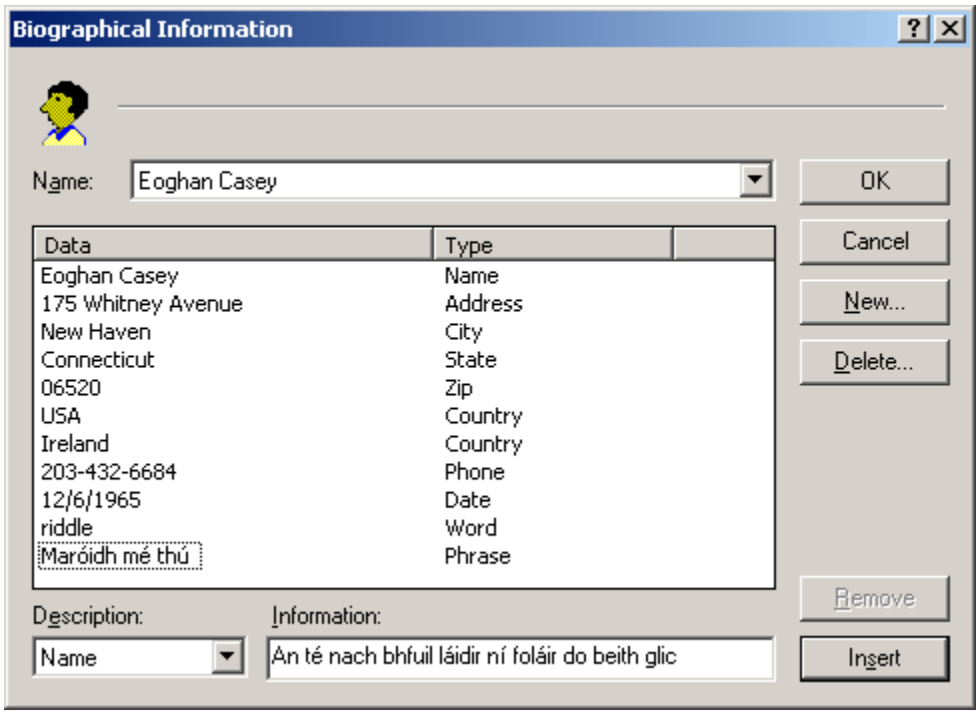

Figure 3: A biographical profile in PRTK supports international characters.

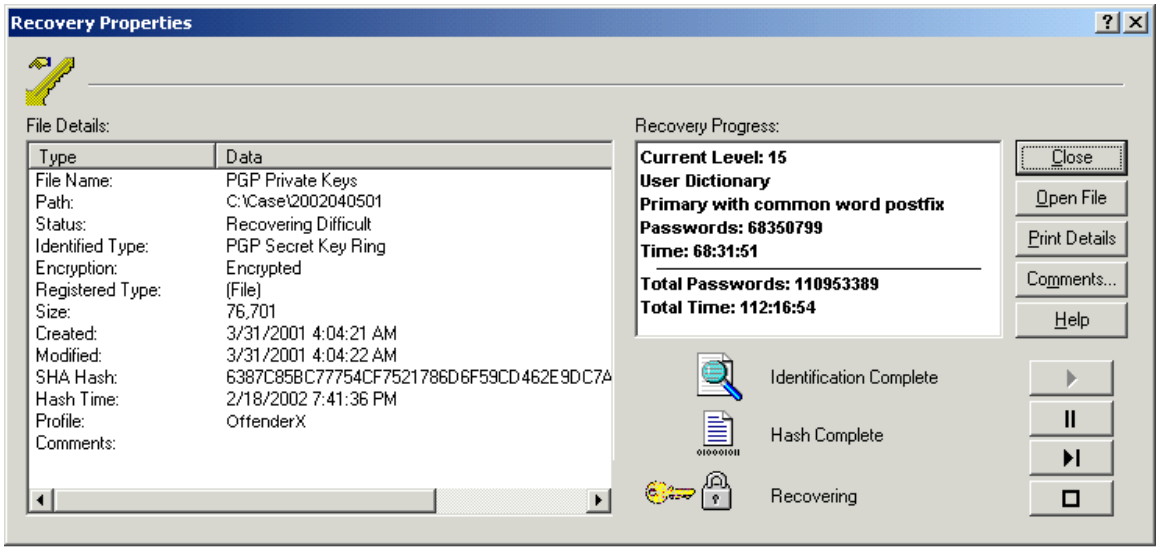

Figure 4: PRTK attempting to guess passphrase of PGP private key

Keep in mind that individuals may change their passphrases periodically. Therefore, examiners should attempt to access all backup copies of private keys in case some have weaker passphrases than others. For instance, when individuals create new PGP keys, the PGP application prompts them to save a copy of the keys to a medium other than the

main drive. Therefore, a backup copy of the private key file may be found on a floppy disk or other external media. Also, PGP periodically makes backup copies of key files on the hard disk, any of which could have a weak passphrase.

As another example, when Windows Encrypting File System (EFS) is used, forensic examiners may be able to guess the user's NT logon password using a tool like L0phtCrack.[xviii] This passphrase can then be used to logon to a restored copy of the suspect's PC and view files protected with EFS. Also, when EFS is used, Windows automatically assigns an encryption recovery agent that can decrypt messages when the original encryption key is unavailable (Microsoft, 1999). In Windows 2000, the built-in Administrator account is the default recovery agent (an organization can override the default by assigning a domain-wide recovery agent provided the system is part of the organization's Windows 2000 domain). The EFS recovery agent of a file can be displayed using the efsinfo Resource Kit utility as shown here:

```
D:\>efsinfo /r /u D:\EFS

EFS: Encrypted
  Users who can decrypt:
    COMPUTER1\username (CN=username,L=EFS,OU=EFS File Encryption Certificate)
  Recovery Agents:
    COMPUTER\Administrator (OU=EFS File Encryption Certificate, L=EFS, CN=Administrator)
```

If the recovery agent is the local Administrator account as in this example, the password for this account may also be attacked using L0phtCrack.

Such password guessing programs cannot be relied on as the only method of recovering encrypted data since they can take a significant amount of time and often have limitations

that are not immediately evident. For instance, in one case the PRTK failed to guess the correct PGP passphrase because it did not combine two potential passphrases entered in the suspect profile independently.

5.0 Recovering Encrypted Network Traffic

As with encrypted files, obtaining plaintext versions of encrypted communications is often more viable than brute force methods. Although it may be possible decrypt some forms of Internet, wireless, and mobile phone communications by exploiting weaknesses in the encryption protocols (Biryukov, A., Shamir, A., Wagner, D., 2000; Stubblefield, Ioannidis, Rubin, 2001), it is generally impractical to decrypt communications particularly as stronger protocols are used.

Instead of attempting brute force attacks, unencrypted data can be sought at end points of a communication channel. When confronted with encryption, many computer intruders attack the end points in favor of more sophisticated attacks. Rather than intercepting credit card numbers transmitted to a commercial Web server via SSL, intruders target the databases used to store the credit cards on the server. Also, the growth in popularity of Secure Shell (SSH) makes it more difficult to obtain passwords as they are transmitted on a network. Computer intruders have adapted to this challenge by attacking the end points: the client and server. On the client side, intruders are gaining unauthorized access to individuals' personal computers and capturing passwords using programs like SubSeven and Back Orifice mentioned in Subsection 4.2. On the server side, intruders are breaking into computers and replacing the SSH server with a modified version that stores

passwords in a file when users connect to the system. Intruders have even compromised the official distributions of SSH servers (CERT, 2002).

For instance, when encrypted e-mail services like Hushmail are used, the communication is most exposed at the sender's and recipient's computers when they compose and read the message, respectively. If voice over IP (VoIP) is tunneled through IPSec the communication is most exposed at either end of the communication channel where monitoring can take place. Traffic on wireless WANs that is encrypted using Wireless Transport Level Security (WTLS) can be obtained at the WAP WTLS gateway or at the Web server endpoint that is joined to the WTLS gateway using SSL. Similarly, mobile phone communication may be encrypted between the device and the core network but are unencrypted at the Mobile Switching Center (MSC) at the core of a mobile communication provider network. Also, it is better to obtain data from the MSC since tapping into base stations and microwaves can be costly and intercepts other peoples' communications, raising privacy concerns (Gibbs, 2002).

6.0 Future Considerations

Existing encryption detection tools are not effective even when the presence of encryption is not concealed. Examiners can search a disk for some PGP related files using the Unix `file` command, `ispgp` from Maresware[xix] or IsEncrypted from Access Data. However, these applications only look for a limited number of encryption types, miss PGPDisk and BestCrypt files, have false positives, and only search allocated space. Ideally, encryption detection tools would find a wider variety of encrypted data on all

areas of a disk.[xx] Similarly, automated mechanisms for detecting encrypted network traffic are essential to help isolate encrypted evidence in large volumes of captured data.

Because plaintext is often wiped after it is encrypted, it is desirable to be able to analyze the media using scanning probe or magnetic force microscopes to recover unencrypted copies of the data. Similarly, it is desirable to be able to perform postmortem analysis on RAM. With the necessary equipment data can be extracted from RAM using debug modes even after power is removed. The hope is that the need for this type of equipment will drive down costs and make it more widely available, making media analysis a viable solution for encryption. As this equipment becomes more generally available, analyzing media for overwritten data will become more common.

Passphrases are one of the weakest components in many encryption applications. Password finding and guessing tools that can access a wider range of file types and intelligently combine keywords to create complex passphrases will make this one of the most viable ways to decrypt evidence. Similarly, as powerful brute force tools become less expensive and easier to use, it will become more practical for smaller organizations to utilize this approach.

Of course, as our ability to recover encrypted evidence improves, criminals will adapt by using stronger encryption. For instance, photon-based quantum cryptography make it impossible to decrypt communications because the act of monitoring alters the transmission. Fortunately, as such techniques evolve, they will also improve our ability to

deal with encryption. For instance, quantum computers would significantly reduce the time it takes to break existing encryption algorithms (Schneier, 1996).

Conclusion

Although encryption can be a formidable hurdle in a forensic examination of digital evidence, it is not insurmountable. Providing attorneys and investigators with plaintext fragments of encrypted documents gives them leverage in a case and may even be sufficient to obtain a conviction. Furthermore, with the knowledge and tools described in this paper, a forensic examiner may be able to obtain or guess encryption passphrases, enabling them to decrypt all associated evidence. However, with the growing number and sophistication of encryption and data hiding tools, it is difficult for an individual forensic examiner to keep pace. This difficulty can be reduced by improvements in tools and increased information sharing among examiners.

References

Biryukov, A., Shamir, A., Wagner, D. (2000), "Real Time Cryptanalysis of A5/1 on a PC", Fast Software Encryption Workshop 2000

CERT, 2002 CA-2002-24 Trojan Horse OpenSSH Distribution [online] Available: http://www.cert.org/advisories/CA-2002-24.html

Demming, D., Baugh, W. E. (1997a), “Encryption and Evolving Technologies as Tools of Organized Crime and Terrorism”, National Strategy Information Center's US Working Group on Organized Crime [online]. Available: http://www.cs.georgetown.edu/~denning/crypto/oc-abs.html

Demming, D., Baugh, W. E. (1997b), “Cases Involving Encryption in Crime and Terrorism” [online]. Available: http://www.cs.georgetown.edu/%7Edenning/crypto/cases.html

Demming, D., Baugh, W. E. (2000) “Hiding Crimes in Cyberspace” in *Cybercrime*, B. D. Loader and D. Thomas (eds.), Routledge [online]. Available: http://www.cs.georgetown.edu/~denning/crypto/hiding1.doc

Gibbs, K. E., Clark, D. F. (2002), “Wireless Network Analysis” in Handbook of Computer Crime Investigation, Casey E. (ed.), London: Academic Press

Kelley, J. (2002), “Militants Wire Web with Links to Jihad Islamic Groups”, USA Today, July 10, 2002

Kingpin (2000) "Palm OS Password Retrieval and Decoding," @stake Security Advisory [online]. Available: http://www.atstake.com/research/advisories/2000/a092600-1.txt

Microsoft (2001) “General Information about Microsoft Office XP Encryption”, Microsoft KB Article Q290112 [online]. Available: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q290112

Microsoft (1999) “Back Up the Recovery Agent Encrypting File System Private Key in Windows 2000”, Microsoft KB Q241201 [online]. Available: http://support.microsoft.com/default.aspx?scid=kb;[LN];Q241201

McAuliffe, W. (2001) “Wonderland used encryption to swap child abuse pictures”. February 2001. ZDNet UK. (Available online at http://news.zdnet.co.uk/story/0,,t269-s2084388,00.html)

Schneier, B. (1996), Applied Cryptography: Protocols, Algorithms, and Source Code in C, New York, NY: John Wiley & Sons.

Stubblefield, A., Ioannidis, J., Rubin, A. (2001) "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP", ATT Labs Technical Report, TD4ZCPZZ, Revision 2. (Available online at http://www.cs.rice.edu/~astubble/wep/)

United States v. Carey, 172 F.3d 1268, 1271 (10th Cir. 1999) [online]. Available: http://laws.findlaw.com/10th/983077.html

United States v. Goossens 84 F.3d 697, 699-704 (4th Cir. 1996) [online]. Available: http://laws.findlaw.com/4th/955520p.html

Unites States v. Marvin Hersh, (11th Cir. 2002) [online]. Available:

http://laws.findlaw.com/11th/0014592opn.html

United States v. Scarfo, (3rd Cir. 2001) WL 1650936 [online]. Available:

http://laws.findlaw.com/3rd/004313.html and

http://www.epic.org/crypto/scarfo/opinion.html)

Usborne, D. (2002), "Has an old computer revealed that Reid toured world searching out new targets for al-Qa'ida?", UK Independent (Available online at

http://www.independent.co.uk/story.jsp?story=114885)

van der Knijff, R. (2002) "Embedded Systems Analsysis" in Handbook of Computer Crime Investigation, Casey E. (ed.), London: Academic Press

[i] In 2001 the FBI put the Earth Liberation Front (http://www.earthliberationfront.com/media/) at the top of the list of North American terrorism threats.
[ii] More recent versions of Microsoft Office incorporate stronger encryption as discussed in Section 3.0: Strong Encryption.
[iii] http://www.accessdata.com/
[iv] http://www.forensics-intl.com/breakers.html
[v] Unsaved Preferences can be exported from a Palm device using the Palm Debugger or using PDA Seizure, pilot-link, and other similar tools.
[vi] http://www.beowulf.org
[vii] A brute force attack against 128-bit encryption involves attempting 2^128 possible keys (assuming an ideally secure encryption algorithm), whereas with a 40 bit key there are only 2^40 possible keys. Thus 128-bit encryption has 2^88 more possibilities.
[viii] Original versions of crypt simulated the Enigma encrypting machine used by the Germans in WWII. Methods of attack on such machines are widely known and newer versions of crypt use stronger encryption schemes such as DES.
[ix] http://www.pgpi.com
[x] Encrypting an entire folder using EFS ensures that temporary copies of encrypted files that it contains are also encrypted.
[xi] http://ntsecurity.nu/toolbox/pmdump/

[xii] If a suspect's password notes are available, the forensic examiner should obtain them in photocopy/facsimile form rather than transcribed since symbols and graphical notes may not be accurately conveyed over the phone or in type written form.
[xiii] Although the Password Recovery Toolkit is one of the most functional password cracking programs available, it has unavoidable limitations. Using a keyword list, PRTK will not find complex passphrases comprised of many words and punctuation. Also, PRTK cannot process less common file types such as Netscape private key databases (key3.db) or files encrypted with Apple File Security (AFS).
[xiv] http://www.spectorsoft.com
[xv] http://www.keyghost.com
[xvi] http://www.keykatcher.com
[xvii] http://www.danwegner.com/keykatch/faq.html
[xviii] http://www.atstake.com/research/lc/
[xix] http://www.maresware.com/maresware/gk.htm#ISPGP
[xx] It may also be fruitful to look for unencrypted copies of private keys on disk. On hardware devices such as SIMs and HSMs, private keys are stored unencrypted on the device and can be obtained directly but expensive equipment is required (van der Knijff, 2002). Some devices such as the iButton will destroy the key if physical tampering is detected but it may be possible to undermine the autodestruct feature by first disabling the power source that would be used to erase the memory of the device.