The Effect of File and Disk Encryption on Computer Forensics

Sarah Lowman
sarah@lowmanio.co.uk
January 2010

ABSTRACT

The advent of easy-to-use encryption programs like Bitlocker for Windows and FileVault for OS X, along with increasing awareness and knowledge of cryptography, means that forensic investigators increasingly must deal with strongly encrypted data. Directly attacking strong encryption is generally futile, but all is not lost: there are recent laws in the UK which force computer users to give up their encryption keys to police as part of an investigation, and there are a number of techniques that forensic investigators can utilise to try and discover keys or obtain plaintext copies of encrypted data. Additionally, users can become careless and may believe that if they use encryption software they need not take further security precautions. This can potentially make their computer easier to investigate.

This report will discuss file and full disk encryption and how they have had positive and negative effects on digital investigations in the past and present.

1. INTRODUCTION

1.1. DEFINITIONS

Encryption is a method of turning meaningful information (known as the *plaintext*) into an obscured format (the *ciphertext*) by means of an algorithm (*cipher*). Encryption algorithms use a *key* to obscure the data (*encryption)* and to recover the plaintext (*decryption*). Good modern algorithms make it infeasible to recover the plaintext from the ciphertext without obtaining the decryption key, and so to protect encrypted data only the key needs to be kept secret. They also make the ciphertext indistinguishable from random data, which can make the use of encryption difficult to prove. For these reasons, encryption is one of the best methods for concealing information and is increasingly used by criminals as a method for hiding their files. It is also used by ordinary people and organisations to minimise the risks of personally identifiable information getting into the wrong hands when, for example, a laptop is stolen (Casey & Stellatos, 2008). The terms *password* and *key* will be used interchangeably to mean either the key to decrypt the data or a password which unlocks a readily-available decryption key.

Full disk encryption (FDE) is when a whole hard drive or the entirety of a particular volume has been encrypted. This can be done using software or hardware. Software such as BitLocker, available on Windows Vista and Windows 7 Ultimate and Enterprise editions, will encrypt everything apart from the Master Boot Record (which it boots from). Hardware based methods can encrypt the disk completely, and are currently supported by a number of hard drive companies including Hitachi and Seagate. Both methods require a password on machine boot-up to decrypt the hard drive. When turned off, the whole hard drive is seemingly filled with random data, and without knowing the password is virtually impossible to decrypt.

Hardware which assists FDE includes the *Trusted Platform Module* (TPM) – a secure cryptoprocessor that can secure encryption keys. Many new laptops have the TPM chip built in, and it can be used by FDE software such as BitLocker as one of the methods of identification.

File encryption involves encrypting files within an operating system rather than entire disks or volumes. It can mean encryption of single files, such as an encrypted Microsoft Word document, or encryption of entire folder, for example with Windows Encrypted File System (EFS).

1.2. OVERVIEW

Having an exact copy of a hard drive which has been fully encrypted and belongs to a suspect who refuses to give up the key is totally useless to an investigator. Even when the type of encryption algorithm is known, a brute force attack on any good encryption key is infeasible (Clayton, 2001). Methods to guess the user's password may be attempted, but if the suspect has chosen one that is long and random, then it is impossible to recover any data.

Luckily, most people are not particularly good at remembering complicated passwords and often write them down or store them on a different medium for backup. People are generally more concerned with ease of use rather than security and so chose passwords that are generally short and contain words or phrases that are memorable to that particular user (Schneier, 2006). Police investigation may also discover passwords used for other services that have also been used for encryption.

Criminals who suspect an imminent police search may start employing encryption, or those who have already done so may destroy their keys. An example of this was *Operation Cathedral* – a major investigation into an online child pornography ring called The Wonderland Club. The Wonderland Club consisted of:

> *"180 members in 13 countries and had amassed over 750,000 images of paedophilia as well as 1,800 computerised videos depicting children suffering sexual abuse."*[1]

When the club learnt that they were being investigated, they began to use encryption to hide the images on their computers. The use of encryption severely stalled the investigators, and as a result the level of prosecution was very low relative to the number of suspects (Casey, 2004).

Although the above example illustrates criminals successfully stepping up their game through the use of encryption, others can be drawn into a false sense of security and make careless mistakes that compromise its effectiveness (Leyden, 2007). For example, they may leave themselves logged onto their computer, bypassing any full disk encryption software, or not bother with any additional login passwords, making investigation easier. Some programs still use weak encryption, with flaws in the algorithms, protocols or key length which may allow the digital investigator to recover the plaintext.

Forensic examiners will deal with FDE and file encryption in different ways, but two common problems are acquiring the key from the suspect and plausible deniability, which are discussed in the next two sections. Although it is now the law in the UK that any encryption keys must be given to the police, this is not the case in other jurisdictions, and the punishment for not surrendering the keys may be far less severe than the potential punishment for any crime committed. Other methods of finding the key are discussed later, including using Password Recovery Toolkit (PRTK) and using the optional password recovery mode on some FDEs. The final two sections discuss specific issues arising in the analysis of FDE and file encryption and possible ways of dealing with these.

2. LEGALLY ACQUIRING THE PASSWORD

2.1. THE UK

On 1st October 2007, Part III of the *Regulation of Investigatory Powers Act* (RIPA) became law. This allows law enforcement agencies to give a person or organization a notice called a Section 49 that requires that they either decrypt any encrypted information or to reveal the key required to do so upon request. A Section 49 is only authorized if the encrypted data is subject to an investigation or operation (RIPA, 2007). In terms of a digital crime investigation, any encrypted data deemed relevant to the investigation would be subject to this, and the suspect would have to hand over either the decrypted files or the decryption key. The punishment for failure to comply is a maximum of two years in prison, or five if it involves national security.

[1] http://www.cambridgenetwork.co.uk/news/article/?objid=22791

An attempt was made to overturn this, with lawyers arguing that the data unlocked may be self-incriminating. However the Court of Appeal (Criminal Division) declared:

> *"...the key to the computer equipment is no different to the key to a locked drawer. The contents of the drawer exist independently of the suspect: so does the key to it. The contents may or may not be incriminating: the key is neutral."*[2]

2.2. THE USA

This is in contrast to American law which, due to the 5th Amendment of the United States Constitution (CRS, 2000), gives people the right to refuse to reveal information that may be self-incriminating. United States v Boucher was one of the first cases that addressed a suspect giving up an encryption key (United States v. Boucher, 2009). Boucher was found with child pornography images on his laptop whilst trying to enter America from Canada through custom control. When seized, the investigators could not reboot the laptop as Boucher had encrypted it with PGP. A Grand Jury Subpoenaed Boucher to supply the password; however this was overturned by a Magistrate Judge as it would constitute self-incrimination. Nevertheless on appeal the Government stated they did not want the password, they just wanted to reproduce the decrypted contents of the hard drive. The court agreed, deciding that Boucher's willingness to open the laptop at border control meant that asking him to produce the decrypted contents would not constitute self-incrimination.

In some cases, however, a suspect has been ordered by a court to reveal an encryption key. For example, in People v. Price (People v. Price, 1998), the prosecution successfully 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."*[3]

3. PLAUSIBLE DENIABILITY

Plausible deniability, applied to cryptography, is when a person denies the use of encryption by blaming the presence of ciphertext on others or denying that encrypted data exists at all.

This is an important issue in computer forensics as ciphertext created by a good encryption algorithm is indistinguishable from random data. Suspects may claim to have wiped the whole disk with random data and it may be impossible to tell if this is the case if no other evidence of full disk encryption is found.

[2] http://www.bailii.org/ew/cases/EWCA/Crim/2008/2177.html
[3] http://cryptome.org/hiding-db.htm

Suspects may also conveniently 'forget' their passwords, claiming they never turn their computer off so never need to remember it, or blame a different user or malicious rootkit for installing any encryption software and seemingly encrypting some files. Although these excuses can appear weak, without any contrary evidence there is nothing to refute them.

Some encryption programs incorporate plausible deniability as part of their operation. Examples include FreeOTFE and TrueCrypt, which allow nested encrypted data: The suspect is able to reveal the key to the outer encrypted partition, but more encrypted data could be hidden inside that. The existence of more encrypted data cannot be proven without knowledge of the decryption keys.

Forensic Innovations, Inc. claim to have found a way to detect headerless TrueCrypt files and distinguish them from random data (ForensicInnovations, 2009). Given that it is usually considered impossible to categorically tell the difference between random and encrypted data, this claim is dubious. Many of the comments appearing on their blog claim large numbers of false positives. As a crude evaluation of their claims, two simple experiments were carried out using their *File Investigator Tools* software. Three files of exactly the same size (20MB) were created. `test1` was a hidden, headerless TrueCrypt volume. `test2` was a file created by a Windows tool called *Random Data File Creator*[4] and `test3` was created by running the command:

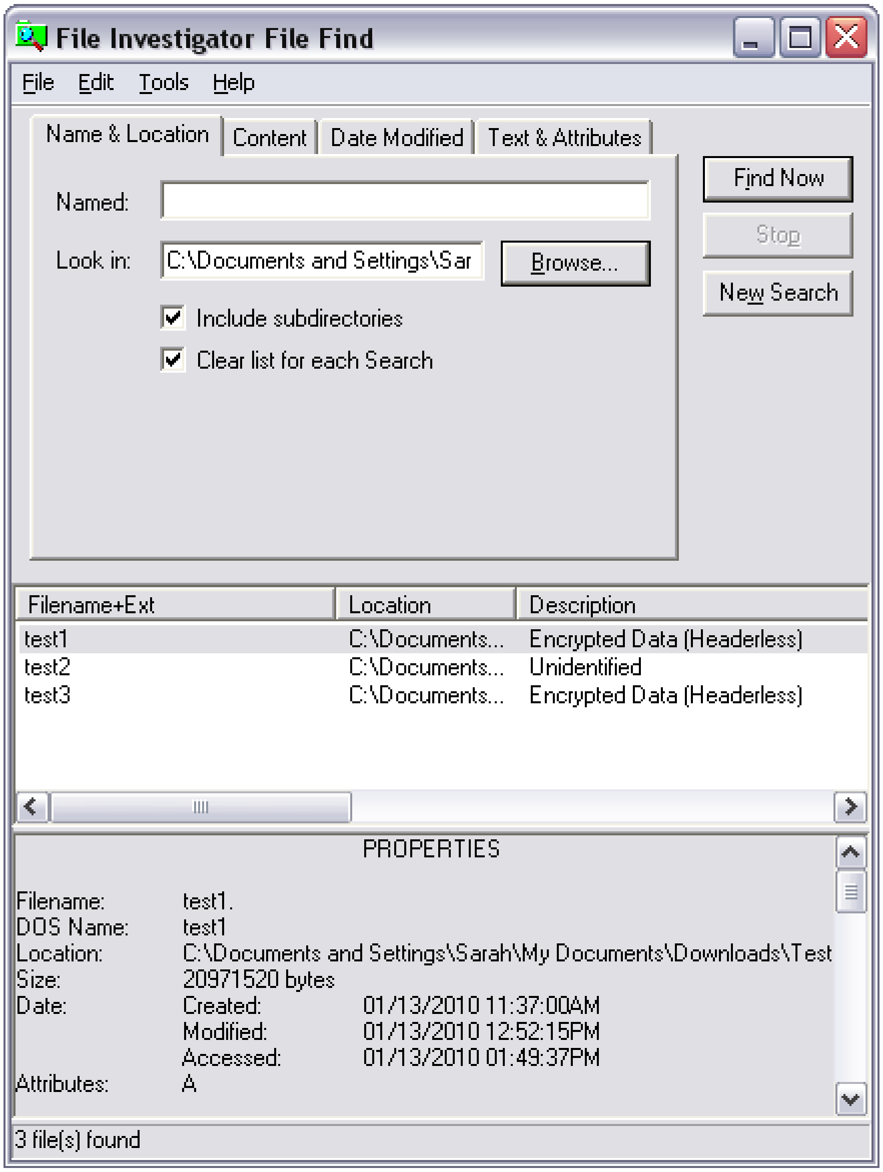
```
> head -c 20m /dev/urandom > test3
```

Since the File Investigator Tools developers claim file extensions are not used to determine file type, they were omitted from the three files. File Investigator Tools was run on the folder containing these three files. It scanned them and output the detected file types. `test1` was correctly identified as headerless encrypted data, but `test3` was falsely identified as encrypted data too. The results can be seen in Figure 1.

To give an idea of the false positive rate, the command used to create `test3` was run 10 more times, creating 10 files containing random data. File Investigator Tools falsely identified all 10 as encrypted data. It is fairly clear that the tool cannot in fact correctly distinguish random data from encrypted data. It can therefore not be used to disprove a claim of plausible deniability, and if used in court could damage an investigator's credibility. It could also lead to investigators wasting time trying to decrypt files that are not encrypted data. That said, there is a place for tools that can identify files containing what looks like random data, as this provides a set of potentially encrypted files upon which possible decryption keys and passwords can be tried.

[4] Download at: http://www.bertel.de/software/rdfc/index-en.html

Some systems encrypt data without the user even knowing the key. It is possible for operating systems support encrypting their virtual memory files/partitions. For example, OS X will do this when *secure swap* is turned on. At each startup, a new swap file is created which is encrypted with a newly generated random key. This is only stored in memory, which means nobody can know the key or recover the contents of a swap file if the system has been switched off, not even the user. The only chance of decrypting swap is on a live system, so significant care must be taken if a Mac is found turned on.



Named:

Look in: C:\Documents and Settings\Sar

☑ Include subdirectories

☑ Clear list for each Search

Filename+Ext	Location	Description
test1	C:\Documents...	Encrypted Data (Headerless)
test2	C:\Documents...	Unidentified
test3	C:\Documents...	Encrypted Data (Headerless)

```
                         PROPERTIES

Filename:      test1.
DOS Name:      test1
Location:      C:\Documents and Settings\Sarah\My Documents\Downloads\Test
Size:          20971520 bytes
Date:          Created:       01/13/2010 11:37:00AM
               Modified:      01/13/2010 12:52:15PM
               Accessed:      01/13/2010 01:49:37PM
Attributes:    A
```

3 file(s) found

Figure 1 Results after running File Investigator Tools on the three files

4. DIFFERENT METHODS OF ACQUIRING KEYS

In most cases encrypted data is completely inaccessible without the decryption key. If the suspect refuses to give their key or pleads plausible deniability, the investigator will have to try other methods to acquire the key. Since everything depends on recovering the password, concentrated effort may prove worthwhile. As mentioned in the overview, strong passwords and keys are long and random. They are generally not breakable by any brute force method since they take too much time and effort, even with the help of supercomputers (Casey, 2002). However, the longer the password, the harder it is to remember, so the more likely the user has stored in on a different medium, written it down, or chosen a memorable phrase.

Any diaries, notebooks and post-it notes should be photographed and examined for passwords at the scene. It is difficult to know in advance if any encryption software will be running on any seized computer, but it is better to obtain as much as possible under a warrant before the suspect has a chance to destroy any potentially valuable evidence. In any case such extrinsic evidence may be useful in other aspects of investigating a computer.

4.1. FULL DISK ENCRYPTION KEYS

Keeping the key available is of crucial importance with FDE, as if the key is lost, *all* the user's data goes with it. Therefore, accessibility and convenience issues surrounding FDE for the user may make it much easier for forensic investigators to find the key compared to file encryption.

All external hardware such as USB devices should be included in any warrant for computers. PGP prompts the user to store any keys on a USB device as a backup and the only BitLocker authentication mechanism that does not require the Trusted Platform Module (TPM) is via a USB device.

Some FDE programs have optional disaster recovery modes which may help investigators get into the system. BitLocker requires a recovery key if the main key (on the USB stick) is lost, which the user may make much shorter and more memorable, and TrueCrypt requires the user to build a rescue disk that can be used to boot the system if the volume header gets damaged (Yegulalp, 2008). In corporate environments the system administrator may be able to provide recovery keys (Casey & Stellatos, 2008).

4.2. FILE ENCRYPTION KEYS

Password and key managers like OS X's *Keychain* provide a single encrypted store for all the user's keys, certificates and passwords (including websites, email, SSH keys, application passwords etc.). These are unencrypted and accessed via a single master password. This provides the forensic investigator with a single point of failure which can be attacked. Also, the files in which the encrypted data is stored are generally of a known and easy-to-spot format (for instance, an XML file with the extension ".keychain" in the case of Keychain). If the user uses a password manager like this, it is possible they will also use it to manage their encryption keys. Using a tool like this means the user never has to remember their individual passwords, and coupled with Keychain's inbuilt random password generator this may mean the protected passwords are extremely complex. However, in every-day operation, the master password usually must be typed fairly regularly. Therefore, for convenience, it is possible the user has made their master password short and quick to type.

Many people reuse the same passwords, or simple variations thereof for multiple accounts. Life assistance company CPP warned:

> *"46 per cent of British internet users, 15.6 million, have the same password for most web-based accounts and five per cent, or 1.7 million, use the same password for every single website."*[5]

Simple techniques to try and crack a password involve running `strings` on the hard drive image to generate a list of keywords and then feeding these into software like Password recovery ToolKit (PRTK). Most modern web browsers come with a password manager with the master password turned off by default, making password recovery trivial. These web passwords can be tried against encrypted data and investigated to look for variations of a common password.

Sometimes keys can be brute forced if weak encryption settings have been used, such as a 40 bit key instead of 128 bit. This was the case for the Wall Street Journal, which came into possession of two hard drives from laptops owned by Richard Reid, who was later convicted by a US federal court of attempting to blow up an aircraft in-flight by detonating explosives hidden inside his shoes (Usborne, 2002). The hard drives contained many files encrypted using Windows Encrypted File System (EFS), but the Wall Street Journal was able to decrypt them:

> *"Luckily, the PC had a version of Windows 2000 with an 'export-quality' key only 40-bits long, rather than the 'US' quality, which being 128-bits long would have been billions of times harder to crack. Even so, it took the equivalent of a set of supercomputers running for five days, 24 hours a day, to find the key."*[6]

Finally, if the password cannot be guessed or brute forced, there may be a vulnerability in the system which can be exploited. Prior to XP, Windows EFS private keys were weakly protected. It was possible to gain access to the data by replacing the NT login password with a known value using a tool like `chntpw`

[5] http://www.telegraph.co.uk/technology/news/6922207/Almost-16-million-use-same-password-for-every-website-study-finds.html

[6] http://www.independent.co.uk/news/world/americas/has-an-old-computer-revealed-that-reid-toured-world-searching-out-new-targets-for-alqaida-663609.html

(Casey, 2002). Microsoft Office used very weak encryption in older versions, just XOR-ing with the password to access files (Microsoft, 2006).

5. EXAMINATION OF FULLY ENCRYPTED HARD DRIVES

It is important when seizing computers to find out if any disks are using full disk encryption, as it impacts the way the computer should be dealt with. Often it is difficult to tell if FDE is being used, even when the computer is switched on, as the software can be very discreet. If FDE is in use, turning off the computer can potentially make all forensic information inaccessible. An additional complication is that directly searching for FDE software will cause the fidelity of the live system to worsen.

If the computer is turned off, the two ways of knowing if the hard drive is encrypted are to turn it on and discover a password is needed to boot, or to inspect the hard drives. Software FDE programs usually leave signatures at the beginning of the hard drive. For example SafeBoot uses the word “safeboot” in sector 0 and PointSec uses the word “Protect” in sector 63 (Casey & Stellatos, 2008). There may be specific FDE boot loaders present and the hard drive will appear to be random data with no structure.

If the investigator knows the key and the type of encryption software used, it is possible to unlock the hard drive by connecting it to a computer with similar software installed and using the key. Although the data is still encrypted at the physical layer, the investigator can then use a forensic acquisition tool to access the data at the logical level.

Another method would be to boot the machine up and enter the decryption key, and then immediately use a forensic boot disk such as `Helix` to take a logical copy of the disk. Using a forensic boot disk has the advantage of not altering any of the data and metadata on the disk. By connecting the hard drive up as a live volume (even as read only), the host operating system may change the access times of certain files whilst decrypting / copying over from the read-only disk. A virtual machine can also be used to explore the hard drive if it will boot successfully. In cases where there is not enough time to use a forensic boot disk or the encryption software will not decrypt the drive as a mounted volume (or the encryption software is proprietary / unavailable), a bit-for-bit copy of the hard drive can be put into a different machine with the same hardware. This is a likely scenario in a corporate environment where all the machines are the same and there is little time to do on-site imaging with a forensic boot disk (Casey & Stellatos, 2008). This is not possible, however, when the encryption requires the Trusted Platform Module.

If the machine is switched on a live copy must be created in case the key is never retrieved, otherwise all the data may be lost when the machine is powered down. A live image can be made usings programs such as `FTK imager lite`. The image produced is also logical and not a physical copy.

All three methods of retrieving the file system produce logical and not physical copies of the hard drive, and it can be argued this is not forensically sound. Hidden information such as deleted files will not be recovered. However in cases where *all* the physical data will be lost due to the machine being turned off, any information – whether 100% forensically sound or not – will still be useful. As long as the forensic investigator can explain what they did and the impact it had on the data, the logical image will be admissible (ACPO, 2007).

6. EXAMINATION OF ENCRYPTED FILES

Even if there is no way to get the full plaintext data from the encrypted data files, all hope is not lost. Fragments of files or even entire files can sometimes be found in unencrypted form on the computer. Before files are encrypted they have to exist in plaintext. During the encryption, temporary plaintext copies may also exist. For example, EFS creates a temporary copy in case a problem occurs during the encryption process (Casey, 2002). Files may also be decrypted and re-encrypted several times. The decrypted copies could be temporarily stored on disk. With a physical analysis of the hard drive images these files may be recovered.

Suspects might also be careless enough to leave encrypted files in decrypted format on the system. An example of this was Ramzi Yousef, convicted of being behind the 1993 bombing of the World Trade Center. When his laptop was seized, the investigators found encrypted files, but also found much of the relevant information in unencrypted format. The encrypted files were successfully decrypted and matched to the plain files (Denning & Baugh, 1997).

Fragments of the plain files may also be found scattered on the disk. For example, Microsoft Word leaves a lot of temporary files behind which can be found by searching for headers and other metadata. File fragments may also exist in the computer's swap file or partition. Partial data can be just as useful as the full files for a conviction, such as in the instance of United States v. Hersh. Hersh was found with several child pornography images on his computer, and had a zip drive seized with encrypted images also suspected to be child pornography. The investigators could not decrypt the images on the zip drive. They received partial source code from the encryption software company (F-Secure) which allowed the headers of the images to be interpreted (United States v Hersh, 2002):

> *"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."*[7]

This method is open for debate as it is possible for two files to have to same name and size, but considering there were 22 matching files and several un-encrypted images had already been found, this amounted to substantial and convincing evidence. The assumptions made about the nature of the encrypted images would not have been possible without the plaintext images on his computer, and it is likely the name and size evidence alone would not be admissible in court.

Fragments or entire files can also be recovered from RAM. If the contents of an application are encrypted, it is likely the plaintext version will be kept in memory for the duration that the application is open. For example, PGPTray can keep copies of plaintext in memory indefinitely (Casey, 2002) and this plaintext can be recovered by using a memory dumping program such as `pmdump`. This is particularly useful if the computer is turned on when seized.

7. FUTURE DEVELOPMENTS

It is likely that the Trusted Platform Module will become a commonplace and well-supported part of all computers in the future. If this is utilised as part of full disk encryption, decryption will have to take place using that computer's particular hardware. Forensic software approaches such as using a virtual machine or mounting the disk as an encrypted volume will not work. Hardware based FDE is currently mostly being used by the military, but may also see mainstream usage in the future. Therefore, it is increasingly likely that specific hardware as well as the key will be needed to decrypt any FDE.

Current attempts at brute-forcing keys do not work due to the significant length of the keys involved. However, with the advent of quantum computers, all existing encryption algorithms based on the difficulty of certain mathematical problems such as finding factors can be broken. *Shor's algorithm*, for example, is a quantum algorithm that finds factors of large numbers in polynomial time compared to the exponential time current brute force methods take (Wolf, 1999).

Recommended key lengths have been increased as computer hardware has gotten faster. This is likely to happen again in the future, and keys and algorithms which are at the moment infeasible to break may become breakable as hardware gets faster and weaknesses are found in current encryption algorithms. Perhaps these breakthroughs can be used to solve cold cases in much the same way as advances in DNA technology are currently being used to solve crimes committed years ago.

[7] http://caselaw.lp.findlaw.com/cgi-bin/getcase.pl?court=11th&navby=docket&no=0014592opn

8. CONCLUSION

The incorporation of encryption as part of operating systems and hardware makes it increasingly simple for individuals to protect their data. As a result, investigators have to modify the way they investigate computers, especially when they are found switched on. Guidelines recommend shutting the machine down (ACPO, 2007), yet the presence of full disk encryption may mean this renders the machine useless.

However, the more important encrypted files are, the more important keeping the key accessible becomes. Keys are now often stored on USB sticks or written down nearby the computer. The same password or simple variations are often used for multiple purposes, and most encryption programs have disaster recovery modes. Even without the key, information or fragments of encrypted files can often be recovered from unencrypted data left in memory or on disk. Encryption can also give a false sense of security, leading users to become lax about other security features on their computer. This can make systems relatively easy to investigate if the key can been determined.

The increased use of strong encryption in the criminal world has led governments to create new laws compelling people to either give up their keys or provide data in unencrypted format. Even in the United States, where encryption keys are protected by the 5th Amendment, several cases have seen suspects reveal their decrypted data without needing to reveal the key.

Forensic investigators have to be vigilant of plausible deniability and assess all the options before coming to a conclusion about whether a disk contains encrypted data. They also have to be aware of the capabilities of any encryption software discovered, as many programs now allow the hiding of encrypted data inside other encrypted data.

Word count: 5074

Bibliography

ACPO. (2007). *Good Practice Guide for Computer-Based Electronic Evidence.*

Casey, E. (2004). *Digital Evidence and Computer Crime.*

Casey, E. (2002). Practical Approaches to Recovering Encrypted Digital Evidence. *International Journal of Digital Evidence* , Volume 1, Issue 3.

Casey, E., & Stellatos, G. J. (2008). *The Impact of Full Disk Encryption on Digital Forensics.* New York: ACM.

Clayton, R. (2001). *Brute force attacks on cryptographic keys*. Retrieved Jan 2010, from http://www.cl.cam.ac.uk/~rnc1/brute.html

CRS. (2000). *Fifth Amendment - self incrimination*. Retrieved Jan 2010, from http://www.law.cornell.edu/anncon/html/amdt5afrag6_user.html#amdt5a_hd24

Denning, D. E., & Baugh, W. E. (1997, Oct). *CASES INVOLVING ENCRYPTION IN CRIME AND TERRORISM*. Retrieved Jan 2010, from http://www.cs.georgetown.edu/~denning/crypto/cases.html

ForensicInnovations. (2009, Apr). *TrueCrypt is now Detectable*. Retrieved Jan 2010, from http://www.forensicinnovations.com/blog/?p=7

Leyden, J. (2007, Feb). *Vista encryption 'no threat' to computer forensics - Who needs a backdoor when users leave the Windows open?* Retrieved Jan 2010, from The Register: http://www.theregister.co.uk/2007/02/02/computer_forensics_vista/

Microsoft. (2006). *Changes in encryption file properties in Office 2003 and Office 2002*. Retrieved Jan 2010, from http://support.microsoft.com/kb/290112

People v. Price, C057243 (Yolo County California Superior Court June 20, 1998).

RIPA. (2007). *RIPA Encryption*. Retrieved Jan 2010, from http://security.homeoffice.gov.uk/ripa/encryption/index.html

Schneier, B. (2006, Dec). *Real-World Passwords*. Retrieved Jan 2010, from http://www.schneier.com/blog/archives/2006/12/realworld_passw.html

United States v Hersh, 97-08051 CR-ASG (US Court of Appeals July 17, 2002).

United States v. Boucher, 2007 WL 4246473 (United States District Court for the District of Vermont Nov 29, 2009).

Usborne, D. (2002). *Has an old computer revealed that Reid toured world searching out new targets for al-Qa'ida?* Retrieved Jan 2010, from http://www.independent.co.uk/news/world/americas/has-an-old-computer-revealed-that-reid-toured-world-searching-out-new-targets-for-alqaida-663609.html

Wolf, R. d. (1999). *Quantom Computation and Shor's Factoring Algorithm.*

Yegulalp, S. (2008). *Hands-On With TrueCrypt 5*. Retrieved Jan 2010, from http://www.informationweek.com/blog/main/archives/2008/02/handson_with_tr.html