Examining the Encryption Threat

Jason Siegfried
Christine Siedsma
Bobbie-Jo Countryman
Chester D. Hosmer
Computer Forensic Research and Development Center

Abstract

This paper is the result of an intensive six-month investigation into encryption technologies conducted at the Computer Forensic Research & Development Center (CFRDC) at Utica College. A significant number of encryption applications were collected and cataloged. A roadmap for the identification of the unique characteristics of encrypted file formats was created. A number of avenues were explored and the results documented. The actual process is not outlined comprehensively due to proprietary needs; however, the following briefly details the process and the significance of our findings.

Introduction

In 2001, a firestorm of controversy erupted in the case of United States V. Nicodemo Scarfo Jr. At issue was the use of Carnivore, a covert key-logging tool that had been the subject of much scrutiny, and its sophisticated successor, Magic Lantern. Because the suspect used advanced encryption technology, law enforcement had to use a sniffing keystroke logging tool. The legal and covert deployment of carnivore and magic Lantern caused many law-abiding citizens to feel that the time of the Orwellian coined term, “Big Brother” had arrived. However, it became evident that law enforcement was unable to decrypt and access encrypted data. The Scarfo case concerning law enforcement’s need for such tactics as Carnivore or Magic Lantern produced fear in law abiding citizens and demonstrated that law enforcement did not have, nor currently has, a better option.

Law enforcement is currently at the mercy of criminal or terrorist entities that employ sophisticated encryption applications. The future success of Magic Lantern is questionable considering two factors: 1) law enforcement must be aware of criminal activities prior to installing the Magic Lantern tool; and 2) the hacker community will not allow such covert techniques to persist, as evidenced by the following quote obtained via Google’s cached feature from a website that is no longer available on the Internet,

> Seeing as how some antivirus software manufacturers will not be looking for the FBI’s Magic Lantern virus, it seems to me that the open source/free software community should be doing what it does best: doing it ourselves.[1]

[1] Investigating Cyber Knight. Posted 24 Nov 2001 by Pseudonym. Original URL <http://www.advogato.org/article/384.html> is no longer available, but access to

The hacking community's ability to defeat new technologies jeopardizes the success of Magic Lantern.

The progressive sophistication and strength of encryption technologies remains a significant obstacle to law enforcement efforts to obtain digital evidence protected by sophisticated mathematical manipulations. The strength of encryption applications consistently advances; the number of encryption applications continues to multiply, and the availability of these sophisticated applications via the Internet continues to increase. Regardless of the grandiose speeds of modern computing technologies, the ability to crack sophisticated encryption tools employed by criminal or terrorist entities remains mind-boggling. The following table demonstrates the machine power required to crack an encryption key in 1997.

Encryption Name & Strength	Time Taken to Crack Key	Machine Power Required to Crack Key	Maximum Speed Required to Crack Key
48 bit RC5	13 days	5000 max, 7000 overall	440,000,000 keys/sec
56 bit RC5	270 days	4000 teams, 10,000's machines	7,000,000,000 keys/sec
64 bit RC5	1,470 days	Not Available	88,000,000,000 keys/sec
Elliptic Curves (109 bit)	120+ days	9,500 in total, 5,000 active at one time	Not Available
RSA 512 bit	Polynomial selection – 2.2 months Factoring – 5.2 months	292 plus a Cray for the last stage	Not Available
56 bit DES	~90 days	Max: 14,000 in a single day	7,000,000,000 keys/sec

Table 1 – Required Time, Machine Power, and Speed in 1997 to Crack Encryption[2]

While 1997 data may seem outdated, the correlation of increasing encryption keys consistently increases along with computing power. In 1997, did law enforcement have the type of machine power, manpower, or financial support to devote such resources to cracking one single encryption key? How likely is it that law enforcement has the resources today to crack the encryption keys deployed in 2004? Furthermore, as the term "quantum encryption" is appearing in security conferences and underground hacker sites alike, law enforcement's ability to catch up to sophisticated encryption tools is nil.

Encryption applications have historically been deployed for legitimate purposes such as privacy, protection, and security. However, the utilization of advanced encryption

<http://216.239.37.104/search?q=cache:6EXloJTwLakJ:www.advogato.org/article/384.html+Investigating+Cyber+Knight&hl=en&ie=UTF-8> is available.

[2] Brute force attacks on cryptographic keys. <http://www.cl.cam.ac.uk/~rnc1/brute.html>. Accessed 21 January 2004.

algorithms has developed into a dual technology applied for legitimate as well as nefarious purposes. In 1997, Dorothy Denning and William Baugh made the following statement, "…our findings suggest that the total number of criminal cases involving encryption worldwide is at least 500, with an annual growth rate of 50 to 100 percent." [3] With the ease of use, current availability, and multiple hacking communities, it can be presumed that even Denning and Baugh understated the use of encryption technologies by criminal and terrorist entities. In the 1999-2000 document, Current U.S. Encryption Regulations: A Federal Law Enforcement Perspective, the author describes the threat as follows.

> …Absent some form of key recovery or recoverable method, a brute force attack will not meet law enforcement needs. If we are working on a terrorist case and intercept a communication that we believe to be in furtherance of criminal activity, and that communication is encrypted – say with PGP, which is 128 bit encryption, a brute force attack to decode one PGP message, using a Cray computer, would take nine trillion times the age of the universe… This is our greatest fear, that, one day, a terrorist attack will succeed because law enforcement could not gain immediate access to the plaintext of an encrypted message…[4]

Without the use of a covert key logging technology such as Carnivore or Magic Lantern, the use of sophisticated encryption applications can stop a digital investigation cold in its tracks. Encrypted data has become a clear obstacle to the furtherance of successful computer forensics investigations. This paper details an intensive six-month research effort, which identified a number of significant characteristics that can be incorporated into a digital forensics investigation. It is hoped that it will provide a number of benefits to law enforcement professionals.

The ability to identify encryption applications using forensic file identification techniques is one that has not yet been seriously explored. Although this six month manually intensive study did not produce an easy way to expedite the cracking of an encryption key or password, it certainly did produce a number of significant results that will expedite the identification of the utilization of an encryption application, among other characteristics of the encryption application.

Currently, random, unintelligible data, not immediately attributed to a file can be inadvertently identified as binary file remnants, previously deleted data, or partially overwritten files, while in fact, it is possible that remnant data can be attributed to encrypted data. The significance of this study's findings can support and assist investigators in quickly identifying the presence of an encryption application, the specific

---

[3] Dorothy Denning and William Baugh. "Encryption and Evolving Technologies as Tools of Organized Crime and Terrorism."

[4] Smith, Charles Barry. 1999-2000. Current U.S. Encryption Regulations: A Federal Law Enforcement Perspective. <http://www.law.nyu.edu/journals/legislation/articles/vol3num1/smith.pdf>. Accessed 21 January 2004.

encryption application used to encrypt digital data, and the signature and/or patterns associated between the encryption application and its subsequent encrypted data.

File Identification through Binary Analysis

A file header is the first portion of an electronic file that contains metadata, as opposed to data.[5] "Metadata is the background information that describes the content, quality, condition, and other appropriate characteristics of the data."[5] It is essentially "data about data." The file header itself is transparent to the user and can only be viewed with a low-level disk viewer/editor. It contains information necessary for the application to "recognize" and "understand" the file. The presence, byte size, and data content of file headers are unique to virtually every application. For example, a Microsoft Word document (.doc) contains very structured and lengthy headers and footers embedded throughout the file (10,752 bytes), as opposed to a basic text file (.txt) that does not even have a header or any other embedded data. Although file header content varies from application to application, the most consistent feature is the presence of a file signature.

File signatures, unlike file extensions, are not easily altered and thus the more accurate means of file identification. Additionally, file extensions are generally limited to only three or four characters; the extension itself tends to be reused for multiple file types.[6] Forensic file type identification is a process used by computer forensic investigators to examine the metadata that applications embed in the files that they create (file header and/or footer), and is the most reliable way of identifying the actual file type. Like any other application that creates files, it is assumed that the resulting encrypted file will have embedded metadata that the file encryption application would use to recognize it as "one of its own," not just by the file extension, but also, the addition of file header and/or footer information.[7]

One purpose of this study was to advance forensic file type identification to the next level through very deep and low-level analysis of encrypted files. The goal for this phase of the experiment was to expand the scope of research to identify not only file signatures, but other important metadata as well. The result was a process to recognize encrypted file signatures and extract detailed information from the encrypted file header.

Two popular file encryption applications were chosen to perform the deep, low-level analysis on. Two programs were chosen to achieve some diversity: RipCoder,[8] very

---

[5] http://inside.uidaho.edu/tutorial/overview/overview.htm

[6] As an example, the .doc extension; commonly recognized as the extension for Microsoft Word documents, a file with that extension could possible one of nine other known file types. See http://www.filext.com/detaillist.php?extdetail=doc

[7] Commonly referred to as 'file signatures.' For a sampling of file types and their associated file signatures, see http://www.garykessler.net/library/file_sigs.html

[8] RipCoder's homepage, http://kach.nm.ru/

basic, easy to use program and FineCrypt,[9] an advanced one with many user-defined options. These popular software programs were obtained freely and anonymously from the Internet. As can be seen from the illustration below, the webattack.com download site had FineCrypt listed as the featured download with RipCoder appearing as well.[10]

Figure 1 – Screenshot from webattack.com Download Site

Experiments were conducted by encrypting files from a standard dataset with combinations of user-defined parameters that are unique to virtually every application. The test dataset consisted of one, two, and eight-byte text files (.txt) along with a 256-byte binary file with each byte representing a different ASCII character starting with the hexadecimal value 00, and ending with the hexadecimal value FF. As the number of options increase with more advanced software, so too does the number of permutations of settings that must be tested. (The FineCrypt analysis required the production of more than 640 encrypted files.)

[9] FineCrypt's homepage, http://www.finecrypt.net/
[10] webattack.com's homepage, http://webattack.com/

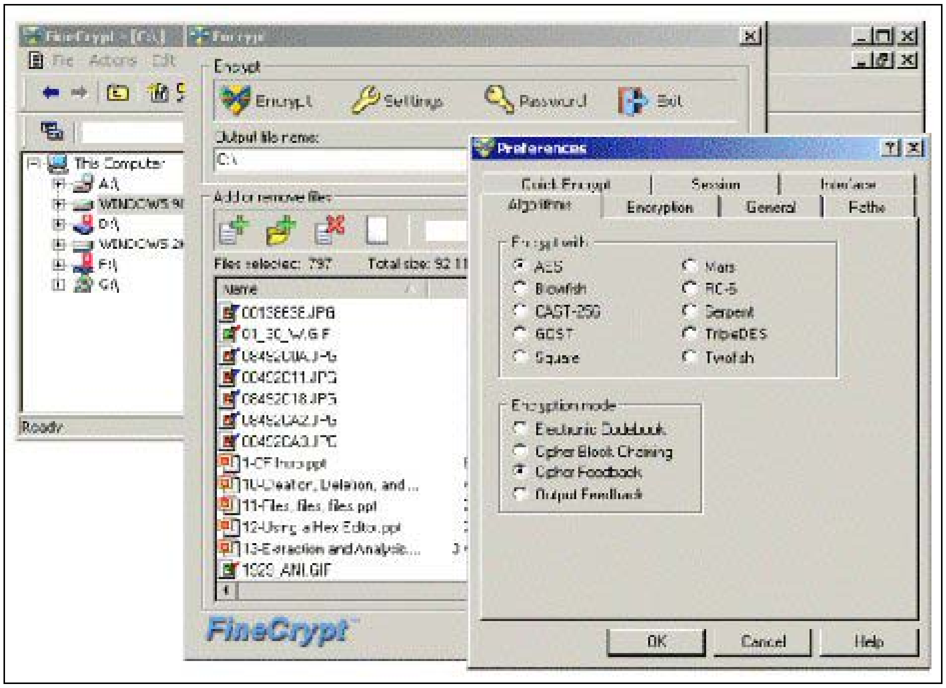

Figure 2 – FineCrypt Interfaces

The resulting encrypted files were then analyzed with a low-level disk viewer to identify metadata contained in the headers and footers of those files. The values in the headers of these files were examined as single byte and byte block values. The key to successful pattern analysis lay in the ability to identify the static header structure and associate the dynamic values with specific attributes of the unencrypted file and/or user-defined options. In addition to the test dataset, a number of files ranging from zero to several thousand bytes were created, encrypted, and analyzed at the experimenter's discretion to pursue predictable value patterns. In order to successfully and efficiently manage and track a dataset of that magnitude, a naming convention using fields based on user-defined options was established. The naming convention allowed for quicker comparisons between encrypted file characteristics and the resulting header values. The following illustrations are screenshots of RipCoder and FineCrypt files as seen with a low-level disk viewer.

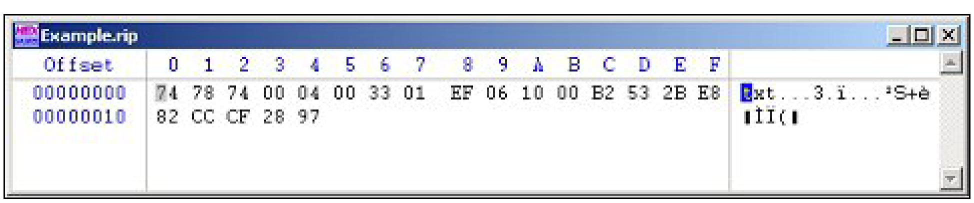

Figure 3 – RipCoder File in Low-Level Disk Viewer

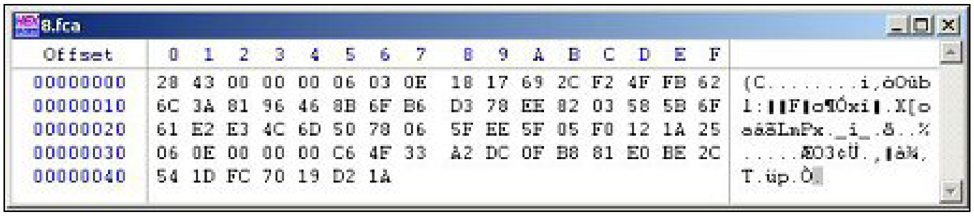

Figure 4 – FineCrypt File in Low-Level Disk Viewer

The analysis efforts were extremely successful. Significant details and characteristics of the unencrypted and encrypted payloads were identified through rigorous examination and analysis of the encrypted files and file headers. The following information can be *located* and *extracted* from the metadata contained in the above files:

- Application signature for positive program identification
- Encryption algorithm used to encrypt payload
- Encryption mode used to encrypt payload
- Password (yes/no) and location of password byte block data
- Key (yes/no) and location of key byte block data
- Compression (yes/no)
- File extension of unencrypted file
- Number of characters in unencrypted file name and location of the bytes representing the name (varies with size of name)
- Encrypted file size excluding four-byte checksum (location of checksum bytes was discovered)
- Number of bytes of cipher text and exact location
- 32-bit write-back option for DES+ algorithm (yes/no)

As an example, consider the FineCrypt header below and note the hexadecimal value of the highlighted offset.

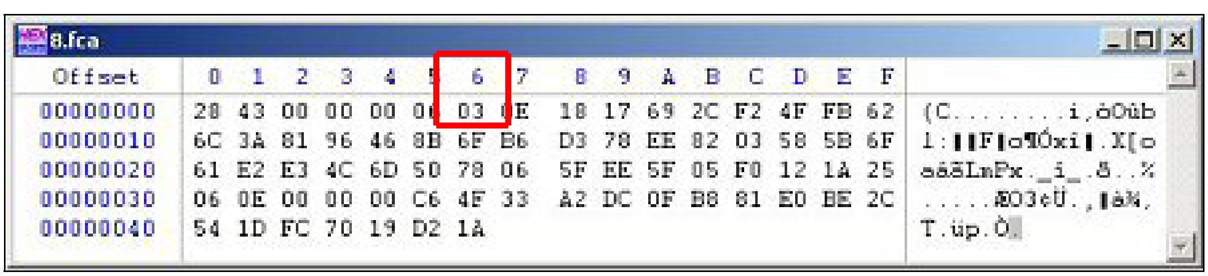

Figure 5 – FineCrypt File in Low-Level

The hexadecimal value of 03 indicates that the algorithm used to encrypt the file was AES and the encryption mode employed was Cipher Feedback. The value of offset 6 will always represent the algorithm and mode selection in FineCrypt files. The complete hexadecimal value matrix for offset 6 appears in the following table.

Offset 06					
Value	**Mode**	**Algorithm**	**Value**	**Mode**	**Algorithm**
00	??????????????????	??????????	15	Electronic Codebook	MARS
01	Electronic Codebook	AES	16	Cipher Block Chaining	MARS
02	Cipher Block Chaining	AES	17	Cipher Feedback	MARS
03	Cipher Feedback	AES	18	Output Feedback	MARS
04	Output Feedback	AES	19	Electronic Codebook	RC-6
05	Electronic Codebook	Blowfish	1A	Cipher Block Chaining	RC-6
06	Cipher Block Chaining	Blowfish	1B	Cipher Feedback	RC-6
07	Cipher Feedback	Blowfish	1C	Output Feedback	RC-6
08	Output Feedback	Blowfish	1D	Electronic Codebook	Serpent
09	Electronic Codebook	CAST-256	1E	Cipher Block Chaining	Serpent
0A	Cipher Block Chaining	CAST-256	1F	Cipher Feedback	Serpent
0B	Cipher Feedback	CAST-256	20	Output Feedback	Serpent
0C	Output Feedback	CAST-256	21	Electronic Codebook	3DES
0D	Electronic Codebook	GOST	22	Cipher Block Chaining	3DES
0E	Cipher Block Chaining	GOST	23	Cipher Feedback	3DES
0F	Cipher Feedback	GOST	24	Output Feedback	3DES
10	Output Feedback	GOST	25	Electronic Codebook	Twofish
11	Electronic Codebook	Square	26	Cipher Block Chaining	Twofish
12	Cipher Block Chaining	Square	27	Cipher Feedback	Twofish
13	Cipher Feedback	Square	28	Output Feedback	Twofish
14	Output Feedback	Square			

Table 2 – Offset 6 Signature Values

The file header structure and value associations remained consistent regardless of the unencrypted file type. Additional tests were run using Microsoft Word, Power Point, and Excel files. Image files were also considered and tested to ensure consistency (.jpeg, .gif, and .bmp). The structures and values remained consistent with very large binary files as well (600 MB random binary file.)

Additional Testing

The deep, low-level analysis of these two file encryption applications produced a significant amount of data. The additional phases of testing involved monitoring file and registry activity during encryption, examining slack space, swap space and unallocated space for passwords and encrypted file content, byte boundary analysis of encryption algorithm and mode padding schemes, and finally, identifying and locating files and registry keys that remained on the test computer after uninstalling the application. A brief discussion of the install/uninstall monitoring results follows.

While RipCoder is a stand-alone executable and does not require installation because it runs from its own program folder, FineCrypt requires its system files to be installed on the computer. We developed a process using installation monitoring software and a text comparison utility to capture and analyze all file and registry activity during installation and uninstallation of applications. The table below summarizes the installation results.

FineCrypt Installation	Files	Registry Keys
Added	48	672
Modified	5	24
Deleted	8	32

Table 3 – FineCrypt Installation Data

After the application was uninstalled, 118 registry keys and eight (8) files remained on the computer. After the system was rebooted, all 118 registry keys remained, but only one of the eight (8) files was present. Although RipCoder runs as a stand-alone application, two ".rip" folders were created in the registry and remained even after the program was deleted from the system. After uninstalling and deleting these applications, file and registry remnants resided on the system as conclusive evidence of prior existence.

Conclusion

Enabling law enforcement to easily identify encrypted files on a suspect machine is only the beginning of what should be continuing research efforts. Although the probability of developing a unique process to easily crack encryption keys or passwords remains quite unlikely, the significant findings produced by these research efforts suggest that small steps can be taken to assist and support law enforcement efforts in analyzing and extracting critical digital evidence in the presence of an encryption application. This research effort produced several significant outcomes. The following are the accomplishments to date.

- Encryption applications were collected and cataloged, establishing a large data set on which to conduct further analysis (455 applications).
- Using this collection, a database of hash values was created (10,529 files), as a tool to aid in the forensic identification of encryption applications.

- Processes and procedures were developed for the identification and extraction of encrypted file metadata.
- Processes and procedures were developed for all other phases of testing including, but not limited to, application remnant identification, system monitoring during encryption, swap and slack space analysis, and cipher text padding analysis.
- A geographical study was launched into the origins of current encryption technologies.
- A roadmap was laid for continued research into the area.

It is imperative that research and development efforts continue to advance the innovative solutions available to law enforcement to combat the strength of modern and continuously progressive encryption applications. The findings produced by this research effort significantly mitigate the time consuming processes of manually identifying encryption applications and what encryption algorithms were used. As research continues, the potential to overcome the impressive leads that criminal and terrorist entities currently maintain with the use of encryption could be significant, without the need to work against the law-abiding public.

For information on obtaining a complete copy of the Encryption Report, please contact Christine Siedsma at the Computer Forensics Research and Development Center. (CFRDC) csiedsma@utica.edu

About the Authors

Chet Hosmer (chet@wetstonetech.com) is a co-founder, President and CEO of WetStone Technologies, Inc. He has over 25 years of experience in developing high technology software and hardware products, and during the last 11 years, Chet has focused exclusively on information security technologies. This focus has resulted in technology innovations in secure time stamping, steganography, network and host based intrusion detection systems, digital watermarking and digital forensics. Chet is a co-chair of the Technology Working Group, one of the seven working groups of National Cybercrime and Terrorism Partnership Initiative sponsored by the National Institute of Justice. He is the Director of the Computer Forensics Research and Development Center (CFRDC) of Utica College, and holds a B.S. degree in Computer Science from Syracuse University.

Jason Siegfried is a Computer Forensics Specialist at WetStone Technologies, Inc. He specializes in Computer Forensics and Information Assurance research. His most recent work includes investigating, collecting, and documenting digital forensic tools and cyber weapons. He completed his internship while working on the Encryption Threat project at the Computer Forensics Research and Development Center during the summer of 2003. He joined WetStone Technologies, Inc. in the fall of 2003. Jason graduated from the Economic Crime Investigation program with a concentration in Computer Security from Utica College of Syracuse University. He graduated with summa cum laude honors in receiving his B.S. in Criminal Justice from Utica in August of 2003. Prior to completing his B.S. degree, he graduated from M.V.C.C. with an Associates degree in Liberal Arts and Science.

Bobbie-Jo Countryman is a Digital Forensics Analyst at WetStone Technologies, Inc. She specializes in the research and investigation of various Information Assurance and Digital Forensics topics. Her most recent project involves a research and investigation effort into the cyber-technologies deployed in America's Critical Infrastructures. She currently acts as a team leader to produce in-depth, detailed reports to expose potential vulnerabilities within America's Critical Infrastructures that provide opportunities to hostile adversaries. Additionally, she assisted research efforts to investigate a number of biometric devices, their capabilities for integration, and their potential for Information Assurance security. She recently earned two Associate of Science degrees consecutively, for Computer Information Systems and Computer Forensics.

Christine Siedsma is the project Manager of the Computer Forensic Research & Development Center at Utica College, where she has supported a number of research efforts in the fields of Computer Security and Digital Forensics. She is also an adjunct instructor of Computer Forensics, and maintains the E-Evidence.info website. Christine earned her B.S. degree in Criminal Justice from Utica College of Syracuse University.

Project Team

The following organizations and individuals contributed their time, effort, and expertise to complete this project:

Air Force Research Laboratory (AFRL)
Joseph Giordano, Peer Review Committee

Northrop Grumman TASC
Dan Kalil, Peer Review Committee

Syracuse University
Dr. Shiu-Kai Chin, Peer Review Committee

Computer Forensic Research and Development Center (CFRDC)
Chester D. Hosmer, Director
Christine Siedsma, Program Manager
Jason Siegfried, Intern
Sabina Bajic, Intern
John Sallustio, Intern