



Paper Session V: Steganography and Terrorist Communications - Current Information and Trends - Tools, Analysis and Future Directions in Steganalysis in Context with Terrorists and Other Criminals

William Eyre
Purdue University

Marcus Rogers
Purdue University, rogersmk@purdue.edu

Follow this and additional works at: <https://commons.erau.edu/adfsl>

 Part of the [Computer Engineering Commons](#), [Computer Law Commons](#), [Electrical and Computer Engineering Commons](#), [Forensic Science and Technology Commons](#), and the [Information Security Commons](#)

Scholarly Commons Citation

Eyre, William and Rogers, Marcus, "Paper Session V: Steganography and Terrorist Communications - Current Information and Trends - Tools, Analysis and Future Directions in Steganalysis in Context with Terrorists and Other Criminals" (2006). *Annual ADFSL Conference on Digital Forensics, Security and Law*. 1.
<https://commons.erau.edu/adfsl/2006/session-v/1>

This Peer Reviewed Paper is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in Annual ADFSL Conference on Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

EMBRY-RIDDLE
Aeronautical University™
SCHOLARLY COMMONS

(c)ADFSL



Steganography and Terrorist Communications: Current Information and Trends - Tools, Analysis and Future Directions in Steganalysis in Context with Terrorists and Other Criminals

William Eyre, Marcus Rogers
Purdue University

Abstract

In ancient times, users communicated using steganography, "...derived from the Greek words *steganos*, meaning 'covered', and *graphein*, meaning 'to write.'" (Singh, 1999, p.5) Steganography facilitates secret, undetected communication. In modern times, in the context of the Global War on Terror, national intelligence and law enforcement agencies need tools to detect hidden information (steganography) in various types of media, most specifically to uncover the placement of hidden information in images. This paper will look at steganography in general terms, presenting the theory of some common steganographic techniques and touching on some theoretical work in steganography. Then a discussion of how to utilize detection tools will shed light on the question of how to make our nation more secure in light of this technology being used by nefarious individuals and organizations.

Keywords: Steganography, information hiding, computer forensics, terrorism, steganalysis, cryptography

1. INFORMATION HIDING: REAL WORLD CONCERN AND POLICY CONSIDERATIONS

Encryption and information hiding techniques have become ubiquitous due to our need for security and privacy in business and personal transactions. The cryptographic and steganographic genie is out of the bottle. In 1997, former NSA director Mike McConnell stated that those who were behind the "...passionate cries for privacy are tied back to somebody selling software or hardware." (Acherman, 1997, p.23)

However, four years before McConnell's statement Levy (1993, p. 6) reported that:

Recently, the head of the French intelligence service quite cheerfully admitted intercepting confidential IBM documents and handing them over to French government-backed competitors. (In cases like these, weak encryption -- which gives a false sense of security -- is worse than no encryption at all.)

Currently, all who wish to use strong encryption have access to tools to allow them able to do so. Encrypted messages attract attention, as they can be detected going across the wire. Steganography, which is difficult to detect if it is being looked for, has been used by terrorists, including "...recently arrested terrorists when they planned to blow up the U.S. Embassy in Paris." (Homer-Dixon, 2002, p. 54). Steganography has entered the arsenal of information age weapons which we acknowledge our enemies are using in the current geo-political environment.

2. INFORMATION HIDING: THEORY

In the traditional model of communications, Alice and Bob communicate via a channel and Eve is the attacker. Various parameters include the premise of who controls the channel, whether the channel is secure or not, and what abilities the attacker has.

In the case of cryptographic communications, Eve will know that Alice and Bob are communicating, and Eve may attempt various attacks to intercept, modify and decrypt communications between Alice

and Bob. These attacks are defined by the various amounts of knowledge Eve possesses. Examples include chosen plaintext and known plaintext attacks, attacks well-defined and understood from the literature on cryptography (Trappe & Washington, 2006).

For purposes of defeating cryptographic communications between terrorists and other known criminals, national security and law enforcement agencies have tools which can easily circumvent the necessity of actually breaking the specific encryption. These tools and methods include keyboard logging to intercept or recover key generating passphrases, bus monitoring software for the same purpose, and RF interception of monitor (CRT) signals. Additionally there are the options of monitoring alternate communications channels and actual surveillance. When the target is high value enough, resources for real time decryption are available, courtesy of the NSA.

When encrypted communications are observed, the encrypted communication and the parties using the encryption attract attention from those who are monitoring the channel. "...the existence of the messages provides some clues as to what's afoot" (Cole, 2003, p.8). Steganography is one technique of information hiding which relies on the premise that even though the attacker, Eve, may have complete control of the channel, Eve is not shutting the channel down. In this scenario we assume Eve will have access to all the messages and there is a large volume of non-threatening (to Eve) traffic on that channel. Messages with hidden information could pass through Eve's filters without detection. It may be possible for Eve to shut down the channel in which the steganographically modified messages were being sent and thus we must assume that there is a reason not to. (One reason would be the possibility of Alice and Bob opening an alternate channel where Eve would not have the control that Eve has with the existing channel.) It behooves the attacker in this situation to be able to detect the steganography, whether or not it can be extracted.

3. INFORMATION HIDING: LOCATIONS

Numerous places to hide information exist. A good treatment of locations for hiding data is in *A Roadmap for Digital Forensic Research* (2001, p.24).

The general categories of Data Hiding enumerated in Workshop 3 of that proceeding are as follows:

- Graphics - (least significant bit, audio, video, imagery, stego)
- Signals - (altered compression algorithms, stego, timing channels, sequencing)
- Applications - (compound doc formats, metadata - reserved structures, file slack)
- Disk Geometry - (marked bad clusters, maintenance track, extra tracks, hidden partitions)
- File Systems - (distributed systems, RAM slack, modified dir entries, unallocated space, boot sector)
- Communications Structures - (reserved packet offsets, email spam, protocols)
- Solid State - (BIOS, CMOS, RAM)
- Data Structures - (heap space)
- OS & Programming - (virus-like expression, rootkits altering system calls, system libraries, DLLs)
- Non-Digital - (perception, filenames, plain sight)

Technologically savvy types, such as programmers and network specialists, can write custom code for hiding information in any or all of these places. The emphasis in current research concerns hiding steganographic information in images and sound files. These files could be directly transmitted or the files could be posted to Web sites to be retrieved by the intended recipients. Cole (2003, p.9) states

that he "...randomly downloaded 500 images from eBay, and over 150 had data hidden in them."

4. INFORMATION HIDING: TECHNIQUES

There are crucial ideas concerning of information hiding that must be kept in mind when thinking about the detection and recovery of hidden messages. The first is that that most terrorist or criminal communication will likely be encrypted before it is hidden. This imbues the person communicating with the advantage of defense in depth. "Pure encryption algorithms are the best way to convert data into white noise. This alone is a good way to hide information in data." (Wayner, 2002, p.31) If it is determined that steganography exists, it is difficult to recover the message because the beginning and end of the message are obscured by the use of cryptography. If the correct message is extracted, the attacker must decrypt the message, which requires knowledge of the encryption algorithm that was used.

A simple method of hiding information in a file is to manipulate the least significant bits of the color of pixels in an image. This is useful to hide the presence of the information from human eyes - as with color depth at 16 or 32 bits the change of the least significant bit in the color will be imperceptible. Sophisticated algorithms use random subsets of pixels in the image to store the hidden information. Using this method, more than one person can use more than one random subset of pixels to store hidden information in the same image. If there are collisions (two people using the same pixel so that the information may be incorrect for one or both of their messages), error correction codes can be used to recover the information damaged in the collision.

Some image formats (.gif, .bmp, etc.) are suited for using the least significant bit method of information hiding. .gif and bitmap images are stored in the same format that they are rendered in, there is no compression. The JPEG format is constructed using lossy compression. That means that when the JPEG is compressed for storage and/or transmission, and then reinflated at the receiving/rendering end, the least significant bits can be lost and therefore the hidden message can be lost.

There is a way to hide information in the JPEG format. JPEG images use a Discrete Cosine Transform (DCT) compression scheme. "The compressed data is stored as integers, and the compression involves extensive floating-point calculations that are rounded at the end. When thus rounding occurs, the program makes a choice to round up or round down. By modulating these choices, messages can be embedded in the DCT coefficients." (Cole, 2003, p.119). J-Steg is a tool which hides data in JPEG files and is very easy to use.

Finally, there is the concept of secret sharing. There are ways to break up information so that the secret message is not understandable unless a certain number of parts are known. A secret can be broken up so that with less than the requisite number of parts, the secret cannot be discovered. The simplest analogy to secrets with n parts in n -dimensional space, is the example of points, lines and axis intercepts. The example works as follows: the secret is in two parts, and is the point at which a line intercepts an axis (x or y , it doesn't matter as long as the line is not parallel to one or the other axis). Both intercepts are known when all the parts of the secret are known. The two parts of the secret are two points. Only one line can be drawn through the two points and it can only intercept each of the axes at one point each. Additionally, more than two points' coordinates can be given out, these points being on the same line, so that a number of people can each know the coordinate of a point and in this case, any two combining their information can draw the line through the two points and come up with the secret (the intercept point of the line and the axis). If the secret is such that the protocol needs three people, one uses other analogies (planes for 3-dimensions, and n -dimensional constructions for n -dimensional secrets). (Wayner, 2002) This example is a simplification and an analogy for how secrets can actually be broken up, or shared. A basic steganographic file system can be constructed to hold m files that are n bits long. (Wayner, 2002).

The importance of secret sharing and knowing the requisite number of parts of the secret to find the

secret is that if parts of the message that are hidden steganographically are found, but these parts are not interpreted properly or there are not enough of them to discover the secret (in this case the secret is the information being hidden), then the attacker will have accomplished only part of the goal of discovering what information is hidden.

5. INFORMATION HIDING: TOOLS

Most of the tools for embedding hidden information are freely or cheaply available. Some tools are open source and therefore it is trivial to modify the code to enable “custom” information hiding. These tools go by many names and some of the more common and well-known tools are EzStego, F5, Hide and Seek, Hide4PGP, Jpeg-Jsteg, OutGuess, Steganos, S-Tools-v4, and White Noise Storm (Wang and Wang, 2004, p.78)¹.

6. STEGANOGRAPHY: PERFECTLY SECURE IMPLEMENTATION

Cachin (2004) discusses the notion of a steganographic implementation which is *perfectly secure*. This notion of perfect security parallels Shannon's notion of *perfect secrecy* (Trappe & Washington, 2006) for cryptosystems. So there is the possibility that for one time use, there are steganographically hidden messages which will never be discovered.

The question of whether the attacker has access to an unmodified version of the cover text is crucial to this notion.

According to Cachin (2004, p. 49):

...the one-time pad stegosystem is equivalent to the basic scheme of visual cryptography. This technique hides a monochrome picture by splitting it into two random layers of dots. When these are superimposed, the picture appears. Using a slight modification of the basic scheme, it is also possible to produce two innocent-looking pictures such that both of them together reveal a hidden embedded message that is perfectly secure against an observer who has only one picture. Hence visual cryptography is an example of a perfectly secure stegosystem.

The implication is such that there is steganography which has no chance of being detected without access to what analogously would be the secret key (i.e. one time pad) in the (perfect secrecy) cryptography analogy. The implication for investigators needs no elaboration.

7. ATTACKS ON STEGANOGRAPHY

Statistical algorithmic analysis is a method which steganalysis tools can employ to discover the presence of hidden information. As the information hiding techniques are standardized in known applications and the places where the information is hidden are defined to the point of the parameters of the embedding program, it is obviously easier to write tools that make use of this information and therefore are better able to detect steganographic messaging. Given that anyone can read about the theory of steganography and look at the available algorithms, that someone (i.e. terrorist or criminal) could attempt to independently implement derivative masking systems and associated algorithms. These independently developed tools would tend to thwart detection efforts based on well known steganography tools. In this context, seizure of terrorist computers and the subsequent code analysis of the applications on these seized computers is critical in the effort to unmask and detect possible terrorist communications using steganography.

Not only could there be “home grown” standalone tools which would import and then modify images, sound files or similar commonly used cover files, but there could also be tools which would insert the information as an adjunct to normal image or sound processing or production. As an example, an application which would crop, rotate or change the color depth of an image could also be importing

¹ Resources for finding these tools change - but a good source for information is <http://www.jjtc.com>.

and distributing information throughout the cover file through some mechanism. It's also not difficult to envision a mechanism to add steganographically hidden information to a file such as a logo when creating an invoice form with a logo in an accounting package. Alternately, some open source word processing or slide show application could be modified to import an image, retrieve input from a file that was encrypted, and add that information as steganographically hidden information to that image in the thread that performs the importation and image placement in the document or slide show. These invoices or "business" documents could then be sent by email or be posted to some secure web site and be considered part of the normal course of business. And when the terrorist suspect's computer was seized, only a careful code analysis would uncover the mechanism which placed the information in the cover file.

8. DETAILED EXAMPLE OF OPEN SOURCE MODIFICATION AND POSSIBLE DETECTION METHODOLOGIES

In the previous section a general example of how information hiding could be embedded in what might appear to be normal business processes was offered. There are many ways to implement these general ideas, and the specific implementations would only be limited by the users' imaginations. In the following example, which could be modified in several ways, each step of the process can be thought of as being optional. The examples following will be option-rich and each option can be thought of as being implemented or being not implemented so that any use of this model may incorporate all, or only some of the features enumerated.

An example of a specific implementation might look like the following:

Company A uses an open source or custom accounting program to generate invoices. These invoices have bitmap logos (much like Quickbooks Pro 2000 for example [although Quickbooks is not open source]). Company A could be an import-export company with distinct entities each with a different logo, or not.

There may be real or dummy invoices or both. Company A might have its own web server with password protected areas of its web site for customer companies (real or not) to view invoices, or the invoices could be distributed via email or any other digital means (physical CDs are a possibility).

Is information hidden in the logo? How could it be injected?

Assume that every time Company A generates an invoice for Customer X, Company A wants to have the option to embed stego. To embed stego in the logo that the user types the (physical) address in a slightly different way than when no stego is to be embedded, so the difference in mailing addresses is not obvious – or they address it to a different division, buyer or office number.

The accounting package, seeing that the input tells it to do something different (i.e. input that tells it to put some stego in the logo we're attaching to the invoice), now looks for the file with the message to embed in the logo.

Depending on the sophistication of the stego input mechanism, it will find the input file and inject the information contained therein into the logo image. The application may or may not alert the user if the file is not found. The file may contain a dummy message. The file may even contain other instructions to the accounting package. In the basic scenario, it tells the accounting package to get a file, input the data from the file into the logo steganographically, and all of this happens under the rubric of a basic accounting package with no obvious stego software involved.

The user posts or emails the invoice and the bad guys have now communicated. It's possible that the company is fairly large so there is a fair amount of data traveling in and out of their domain (an energy producer perhaps).

How do we detect this?

Someone first has to suspect something. One method to start with is to take hash values of the logos

attached to the invoices. If the hashes don't match, there is the possibility of communication taking place. Other ideas would involve traffic analysis and traditional methods for observing those who are under suspicion.

Once the computer generating these documents and logos is seized or imaged, the application's behavior must then be observed. Investigators would need to run the application, input previously known input strings and watch for any behavior inconsistent with 'normal' functionality. And then when disassembling the code, (assuming the source code is not available), look for embedded strings in the constant section of the data segment, or in the code segment itself, and determine if these strings are input or output values, or filenames or paths. Finally there would be a search for obvious encrypted files, and other artifacts external to the actual code (source or object) of the application.

This example demonstrates the need for extensive and thorough code analysis on all the applications extent on a suspect's computer. It is critical to actually seize the computers of the terrorists and criminals who are creating messages with this hidden information

9. STEGANALYSIS: TECHNIQUES

The classes of attacks on steganography are roughly analogous to attacks on cryptography conducted under cryptanalysis.

These attacks are generally classified in the following ways according to Kessler (2004, p.15):

- Stego-only attack: The stego medium is the only item available for analysis.
- Known carrier attack: The carrier and stego media are both available for analysis.
- Known message attack: The hidden message is known.
- Chosen stego attack: The stego medium and algorithm are both known.
- Chosen message attack: A known message and stego algorithm are used to create stego media for future analysis and comparison.
- Known stego attack: The carrier and stego medium, as well as the stego algorithm, are known.
- Stego methods for digital media can be broadly classified as operating in the image domain or transform domain. Image domain tools hide the message in the carrier by some sort of bit-by-bit manipulation, such as LSB insertion. Transform domain tools manipulate the stego algorithm and the actual transformations employed in hiding the information, such as the DCT coefficients in JPEG images.

10. STEGANALYSIS: TOOLS

Techniques implemented by information hiding tools are known, and as such many experts have written and marketed these detection tools based on knowledge of the tools used to embed steganography for attempting to detect steganography.

Wetstone purportedly achieved the Holy Grail of these tools.

Stego Suite is such a tool that identifies the presence of steganography without prior knowledge of the steganography algorithm that might have been used against the target file. Known as "blind steganography detection, this capability is exclusive to Stego Suite." (Wetstone, 2006, <http://www.wetstonetech.com/catalog/item/1104418/619451.htm>)

It is useful in grappling with the problem of terrorist and criminal communication to have the ability to run a general purpose tool without regard to the actual steganographic routine employed and to detect the presence of steganographic communication. If there is a steganographic communication we can attempt to extract and decrypt, or destroy the communication.

11. CONCLUSION

Steganography has its origins in antiquity and in the digital age can take many forms. There are many types of locations and many vectors which can be exploited to execute data hiding strategies, and any of these vectors could be considered steganography by strict definition. Steganography has recently come to be understood to mean the hiding of information in image and sound files. Tools for hiding information in images and sound files are freely and cheaply available. The concepts and techniques of hiding information are well documented and well understood. Information regarding these concepts and tools are therefore available to criminals and terrorists as well as law-abiding organizations and individuals.

The use of steganography complicates the task of monitoring terrorist communication. Steganography is difficult to detect, and coupled with the necessity of breaking the encryption, as most hidden information is expected to be encrypted, it is difficult to extract the information. As new techniques are developed to detect steganography, the developers writing software to embed steganography incorporate knowledge of how the current detection tools work into the design of newer tools for embedding hidden information. Thus the tools to embed steganography become more powerful and more apt to hide information in a way that current detection tools cannot detect.

Terrorists and criminals can design their own information hiding tools, and these tools could act in uncommon ways – ways in which the known available steganographic information hiding tools do not act. This fact makes it essential that investigators conduct extensive code analysis on the computers seized from terrorists.

Communication methods which allow terrorists to plan and execute attacks are of great concern to law enforcement and national intelligence agencies. Pursuing the detection of steganographic communications must become a matter of policy. In studying the theoretical and technical hurdles inherent in detecting steganography, it becomes incumbent upon policy makers to designate the appropriate resources and apply them to solving the problem of real-time steganography detection.

12. REFERENCES

- Acherman, R. K. (1997). Security Balances Needs of Privacy, Law Enforcement. *Signal*, 51 (6), 23.
- Cachin, C. (2004). An Information Theoretic Model for Steganography. *Information and Computation*, 192, 41-56.
- Cole, E. (2003). *Hiding in Plain Site*. Indianapolis: Wiley Publishing.
- Homer-Dixon, T. (2002). The Rise of Complex Terrorism. *Foreign Policy*, 128, 52-62.
- Kessler, G.C. (2004). An Overview of Steganography for the Computer Forensics Examiner. Retrieved February 26, 2006, from <http://www.wetstonetech.com/f/stego-kessler.pdf>.
- Levy, S. (May/June 1993). Crypto Rebels. *Wired Magazine*, 1.03. Retrieved February 26, 2006, from http://wired-vig.wired.com/wired/archive/1.02/crypto.rebels.html?pg=6&topic=&topic_set=
- Moskowitz, I.S., Longdon, G.E. & Chang, L. (2000). A New Paradigm Hidden in Steganography. *New Security Paradigm Workshop*. Ballycotton, Co Cork, Ireland. 41-50.
- Palmer, G. (2001). Workshop 3 - Detection and Recovery of Hidden Data. *A Roadmap for Digital Forensic Research*. Air Force Research Laboratory, Rome Research Site. 23-26.
- Singh, S. (1999). *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. New York: Anchor Books.
- Trappe, W. and Washington, L. C. (2006). *Introduction to Cryptography with Coding Theory*. 2nd ed.. Upper Saddle River: Pearson Prentice Hall.
- Wayner, P. (2002). *Disappearing Cryptography - Information Hiding: Steganography and*

Watermarking. 2nd ed. Boston: Morgan Kaufmann Publishers.

Wang, H. & Wang, S. 2004. Cyber Warfare: Steganography vs. Steganalysis. *Communications of the ACM*, 47 (10), 76-82.

Wetstone Site. (n.d.). *Stego Suite™ - Commercial*. retrieved February 26, 2006, <http://www.wetstonetech.com/catalog/item/1104418/619451.htm>.