ISSN:0975-9646

Shruti/ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3763-3768

Comparision in cover media under Stegnography Digital media by Hide and Seek Approach

Shruti
Deptt. Of Computer science
Guru Nank Dev University
Gurdaspur, India

Abstract:-Although every people have secrets in plain sight which is now a days known as stegnography. This recent growth in computational Power and technology stegnography become today's security technique. Thus embedding hidden content in unremarkable cover media so as not to arrose an eavesdropper suspicion. In this paper I m going to discuss about embedding technique as well as its certain properties like security, robustness and capacity. Moreover through this research article comparison of cover media is also being discussed.

Keywords:-Classical Stegnography System, DCT, DCT Coefficient, Hide and Seek Approch, Kerckhoff's Principle, Steganalysis

I. INTRODUCTION

Steganography is the art and science of hiding communication; a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper's suspicion. In the past, people used hidden tattoos or invisible ink to convey steganographic content. Today, computer and network technologies provide easy-to-use communication channels for steganography. Essentially, the information-hiding process in a steganographic system starts by identifying a cover medium's redundant bits (those that can be modified without destroying that medium's integrity)[1]. The embedding process creates a stego medium by replacing these redundant bits with data from the hidden message. Modern steganography's goal is to keep its mere presence undetectable, but steganographic systems—because of their invasive nature—leave behind detectable traces in the cover medium. Even if secret content is not revealed, the existence of it is: modifying the cover medium changes its statistical properties, so eavesdroppers can detect the distortions in the resulting stego medium's statistical properties. The process of finding these distortions is called statistical steganalysis. This article discusses existing steganographic systems and presents recent research in detecting them via statistical steganalysis. Other surveys focus on the general usage of information hiding and watermarking or else provide an overview of detection algorithms.[2,3] Here, I present recent research and discuss the practical application of detection algorithms and the mechanisms for getting around them.

II. THE BASICS OF EMBEDDING

Three different aspects in information-hiding systems contend with each other: capacity, security, and robustness[4]. Capacity refers to the amount of information that can be hidden in the cover medium, security to an eavesdropper's inability to detect hidden information, and robustness to the amount of modification the stego medium can withstand before an adversary can destroy hidden information. Information hiding generally relates to both watermarking and steganography. A watermarking system's primary goal is to achieve a high level of robustness—that is, it should be impossible to remove a watermark without degrading the data object's quality. Steganography, on the other hand, strives for high security and capacity, which often entails that the hidden information is fragile. Even trivial modifications to the stego medium can destroy it. A classical steganographic system's security relies on the encoding system's secrecy. An example of this type of system is a Roman general who shaved a slave's head and tattooed a message on it. After the hair grew back, the slave was sent to deliver the now-hidden message[5]. Although such a system might work for a time, once it is known, it is simple enough to shave the heads of all the people passing by to check for hidden messages—ultimately, such a steganographic system fails. Modern steganography attempts to be detectable only if secret information is known—namely, a secret key[2]. This is similar to Kerckhoffs' Principle in cryptography, which holds that a cryptographic system's security should rely solely on the key material[6]. For steganography to remain undetected, the unmodified cover medium must be kept secret, because if it is exposed, a comparison between the cover and stego media immediately reveals the changes.

Information theory allows us to be even more specific on what it means for a system to be perfectly secure. Here's an information-theoretic model for steganography that considers the security of steganographic systems against passive eavesdroppers[7].

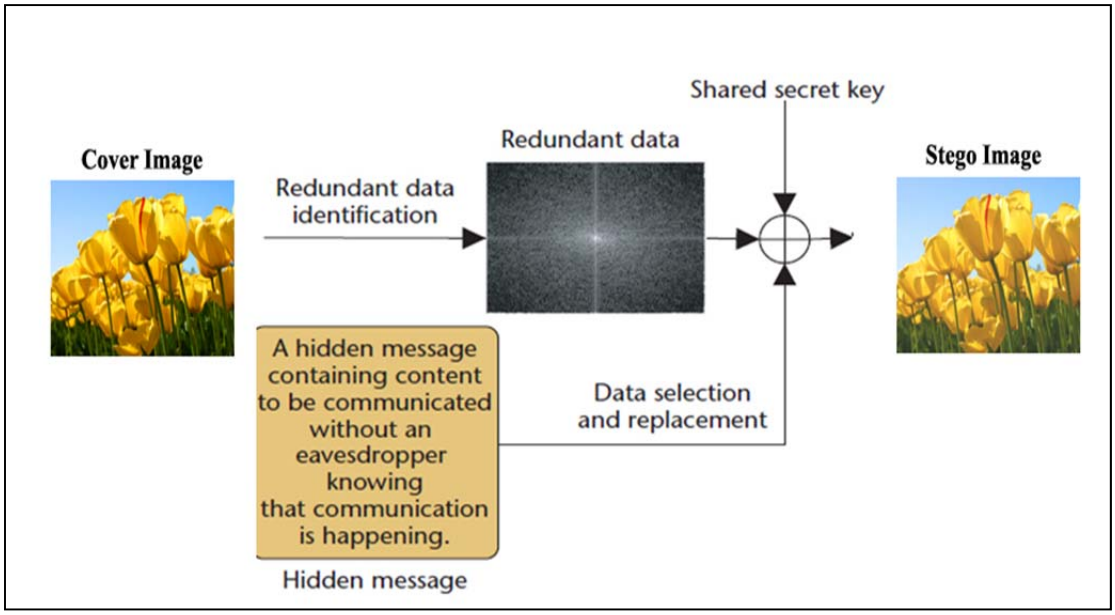

Fig.1 Modern steganographic communication. The encoding step of a steganographic system identifies redundant bits and then replaces a subset of them with data from a secret message.

In this model, you assume that the adversary has complete knowledge of the encoding system but does not know the secret key. His or her task is to devise a model for the probability distribution PC of all possible cover media and PS of all possible stego media. The adversary can then use detection theory to decide between hypothesis C (that a message contains no hidden information) and hypothesis S (that a message carries hidden content). A system is perfectly secure if no decision rule exists that can perform better than random guessing. Essentially, steganographic communication senders and receivers agree on a steganographic system and a shared secret key that determines how a message is encoded in the cover medium. To send a hidden message, for example, Alice creates a new image with a digital camera. Alice supplies the steganographic system with her shared secret and her message. The steganographic system uses the shared secret to determine how the hidden message should be encoded in the redundant bits. The result is a stego image that Alice sends to Bob. When Bob receives the image, he uses the shared secret and the agreed on steganographic system to retrieve the hidden message. Fig 1 shows an overview of the encoding step; as mentioned earlier, statistical analysis can reveal the presence of hidden content[8–12].

III. HIDE AND SEEK APPROCH

Although steganography is applicable to all data objects that contain redundancy, in this article, we consider JPEG images only (although the techniques and methods for steganography and steganalysis that we present here apply to other data formats as well). People often transmit digital pictures over email and other Internet communication, and JPEG is one of the most common formats for images. Moreover, steganographic systems for the JPEG format seem more interesting because the systems operate in a transform space and are not affected by visual attacks[8] (Visual attacks mean that you can see steganographic messages on the low bit planes of an image because they overwrite visual structures; this usually happens in BMP images). Through this research paper I showed that steganographic systems for palette-based images leave easily detected distortions[9]. Let's look at some representative steganographic systems and see how their encoding algorithms change an image in a detectable way. I compare the different systems and contrast their relative effectiveness.

IV. DISCRETE COSINE TRANSFORM

For each color component, the JPEG image format uses a discrete cosine transform (DCT) to transform successive 8 × 8 pixel blocks of the image into 64 DCT coefficients each. The DCT coefficients F(u, v) of an 8 × 8 block of image pixels f(x, y) are given by, where C(x) = 1/ when x equal 0 and C(x) = 1 otherwise. Afterwards, the following operation quantizes the coefficients

$$F(u,v) = \frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) * \cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right],$$

$$F^{Q}(u,v) = \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$$

where Q(u,v) is a 64-element quantization table. We can use the least-significant bits of the quantized DCT coefficients as redundant bits in which to embed the hidden message. The modification of a single DCT coefficient affects all 64 image pixels. In some image formats (such as GIF), an image's visual structure exists to some degree in all the image's bit layers. Steganographic systems that modify least-significant bits of these image formats are often susceptible to visual attacks[8]. This is not true for JPEGs. The modifications are in the frequency domain instead of the spatial domain, so there are no visual attacks against the JPEG image format. The JSteg algorithm. As it runs, the algorithm sequentially replaces the least-significant bit of discrete cosine transform (DCT) coefficients with message data. It does not require a shared secret.

Input: message, cover image

Output: stego image

while data left to embed

do

get next DCT coefficient from cover image

if

DCT ≠ 0 and DCT ≠ 1

then

get next LSB from message

replace DCT LSB with message LSB

end if

insert DCT into stego image

end while

V. SEQUENTIAL

.

Derek Upham's JSteg was the first publicly available steganographic system for JPEG images. Its embedding algorithm sequentially replaces the least-significant bit of DCT coefficients with the message's data (see above algorithm)[13]. The algorithm does not require a shared secret; as a result, anyone who knows the steganographic system can retrieve the message hidden by JSteg. Andreas Westfeld and Andreas Pfitzmann noticed that steganographic systems that change least-significant bits sequentially cause distortions detectable by steganalysis[8]. They observed that for a given image, the embedding of high-entropy data (often due to encryption) changed the histogram of color frequencies in a predictable way. In the simple case, the embedding step changes the least significant bit of colors in an image. The colors are addressed by their indices i in the color table; we refer to their respective frequencies before and after embedding as ni and ni*. Given uniformly distributed message bits, if n2i > n2i+1, then pixels with color 2i are changed more frequently to color 2i + 1 than pixels with color 2i + 1 are changed to color 2i. As a result, the following relation is likely to hold:-

$|n2i - n2i+1| \geq |n2i^* - n2i+1^*|$

In other words, embedding uniformly distributed message bits reduces the frequency difference between adjacent colors. The same is true in the JPEG data format. Instead of measuring color frequencies, we observe differences in the DCT coefficients' frequency. Fig2, displays the histogram before and after a hidden message is embedded in a JPEG image. We see a reduction in the frequency difference between coefficient –1 and its adjacent DCT coefficient –2. We can see a similar reduction in frequency difference between coefficients 2 and 3

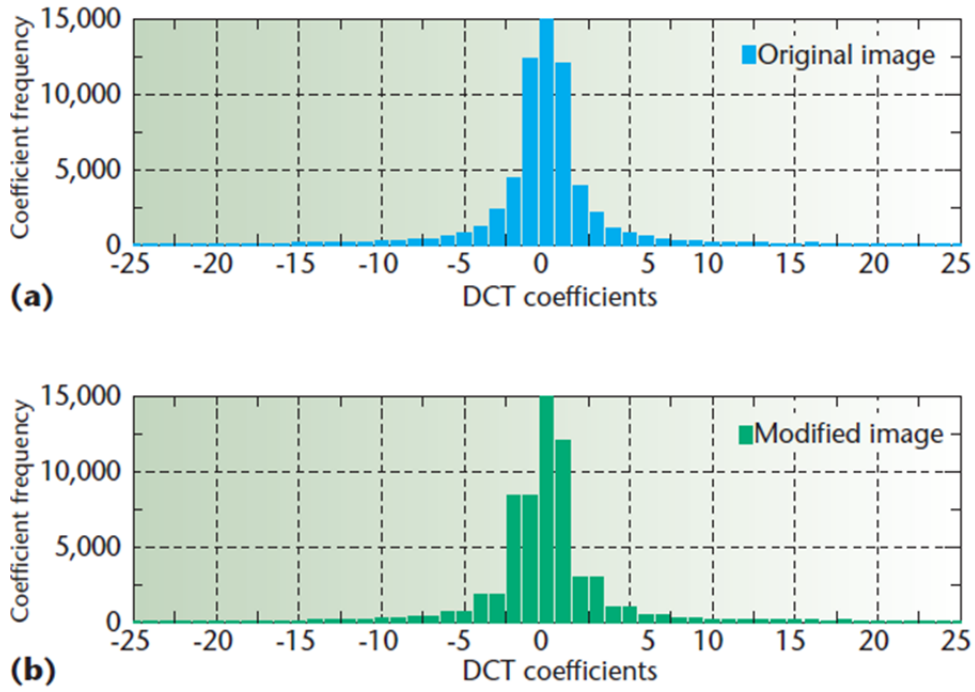

Fig2. Frequency histograms. Sequential changes to the (a) original and (b) modified image's least-sequential bit of discrete cosine transform coefficients tend to equalize the frequency of adjacent DCT coefficients in the histograms.

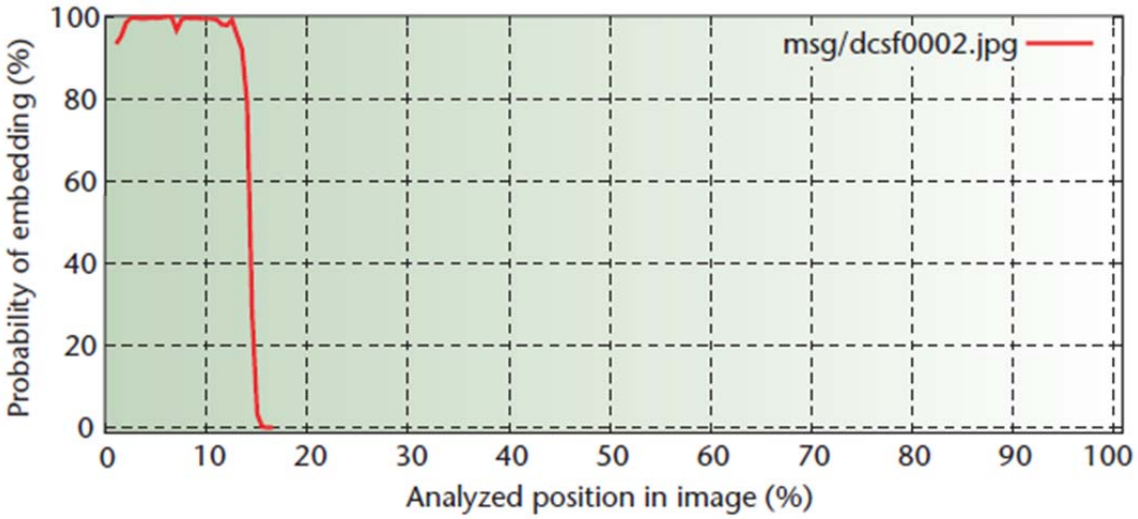

Fig3. A high probability of embedding indicates that the image contains steganographic content. With JSteg, it is also possible to determine the hidden message's length.

The probability of embedding is determined by calculating *p* for a sample from the DCT coefficients. The samples start at the beginning of the image; for each measurement the sample size is increased. Fig3, shows the probability of embedding for a stego image created by JSteg. The high probability at the beginning of the image reveals the presence of a hidden message; the point at which the probability drops indicates the end of the message.

VI. STEGANOGRAPHY-DETECTION ON THE INTERNET

How can we use these steganalytic methods in a real world setting—for example, to assess claims that steganographic content is regularly posted to the Internet?[13–15] To find out if such claims are true, we created a steganography detection framework[16] that gets JPEG images off the Internet and uses steganalysis to identify subsets of the images likely to contain steganographic content.

A. Steganographic systems in use

To test this framework on the Internet, I started by searching the Web and Usenet for three popular steganographic systems that can hide information in JPEG images: JSteg (and JSteg-Shell), JPHide, and OutGuess. All these systems use some form of least-significant bit embedding and are detectable with statistical analysis. JSteg-Shell is a Windows user interface to JSteg first developed by John Korejwa. It supports content encryption and compression before JSteg embeds the data. JSteg-Shell uses the RC4 stream cipher for encryption (but the RC4 key space is restricted to 40 bits). JPHide is a steganographic system Allan Latham first developed that uses Blowfish as a PRNG.[17,18] Version 0.5 (there's also a version 0.3) supports additional compression of the hidden message, so it uses slightly different headers to store embedding information. Before the content is embedded, the content is Blowfish-encrypted with a user-supplied pass phrase.

B. Detection framework

Stegdetect is an automated utility that can analyze JPEG images that have content hidden with JSteg, JPHide, and OutGuess 0.13b. Stegdetect's output lists the steganographic systems it finds in each image or writes "negative" if it couldn't detect any. We calibrated Stegdetect's detection sensitivity against a set of 500 non-stego images (of different sizes) and stego images (from different steganographic systems). On a 1,200-MHz Pentium III processor, Stegdetect can keep up with a Web crawler on a 10 MBit/s network. Stegdetect's false-negative rate depends on the steganographic system and the embedded message's size. The smaller the message, the harder it is to detect by statistical means. Stegdetect is very reliable in finding images that have content embedded with JSteg. For JPHide, detection depends also on the size and the compression quality of the JPEG images. Furthermore, JPHide 0.5 reduces the hidden message size by employing compression. Fig4, shows the results of detecting JPHide and JSteg. For JSteg, we cannot detect messages smaller than 50 bytes. The false-negative rate in such cases is almost 100 percent. However, once the message size is larger than 150 bytes, our false-negative rate is less than 10 percent. For JPHide, the detection rate is independent of the message size, and the false-negative rate is at least 20 percent in all cases. Although the false-negative rate for OutGuess is around 60 percent, a high false-negative rate is preferable to a high false-positive rate, as we explain later

.

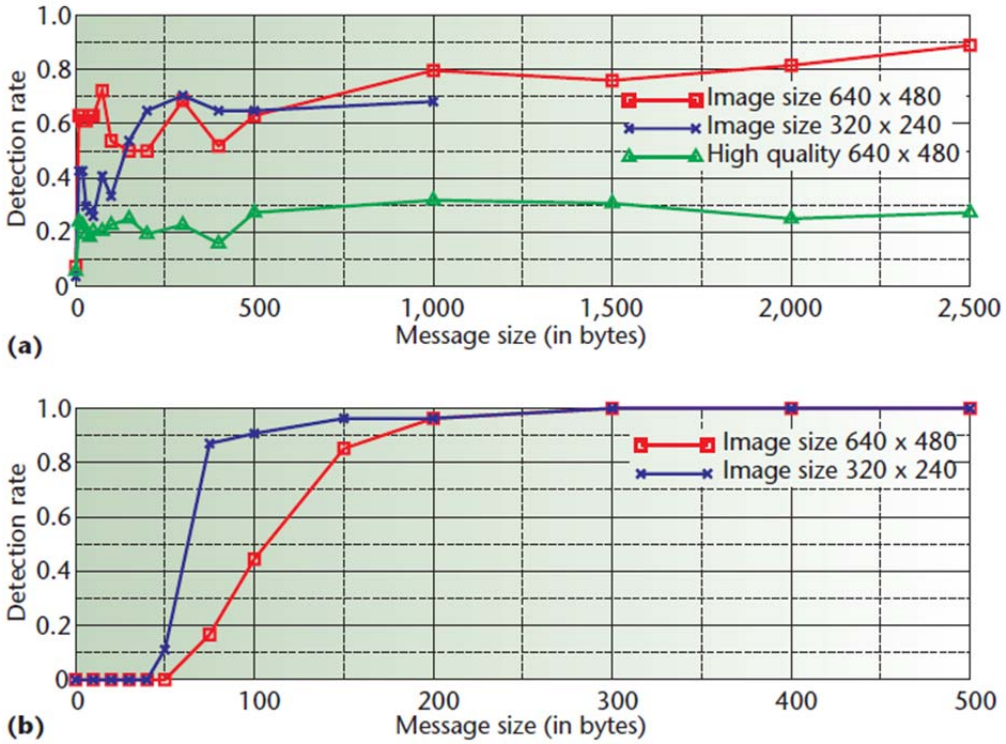

Fig4. Using Stegdetect over the Internet. (a) JPHide and (b) JSteg produce different detection results for different test images and message sizes.

Table1:-Percentage of (false) positives for analysed images

TEST	EBAY	USENET
JSteg	0.003	0.007
JPHide	1	2.1
OutGuess	0.1	0.14

C. Finding images

To exercise our ability to test for steganographic content automatically, we needed images that might contain hidden messages. We picked images from eBay auctions (due to various news reports)[13,14] and discussion groups in the Usenet archive for analysis.[19] To get images from eBay auctions, a Web crawler that could find JPEG images was the obvious choice. Unfortunately, there were no open-source, image-capable Web crawlers available when we started our research. To get around this problem, we developed Crawl, a simple, efficient Web crawler that makes a local copy of any JPEG images it encounters on a Web page. Crawl performs a depth-first search and has two key features:

- Images and Web pages can be matched against regular expressions; a match can be used to include or exclude Web pages in the search.
- Minimum and maximum image size can be specified, which lets us exclude images that are too small to contain hidden messages. We restricted our search to images larger than 20 Kbytes but smaller than 400.

We downloaded more than two million images linked to eBay auctions. To automate detection, Crawl uses stdout to report successfully retrieved images to Stegdetect. After processing the two million images with Stegdetect, we found that over 1 percent of all images seemed to contain hidden content. JPHide was detected most often (see Table 1).

We augmented our study by analyzing an additional one million images from a Usenet archive. Most of these are likely to be false-positives. Stefan Axelsson applied the base-rate fallacy to intrusion detection systems and showed that a high percentage of false positives had a significant effect on such a system's efficiency.[20] The situation is very similar for Stegdetect. We can calculate the true-positive rate—the probability that an image detected by Stegdetect really has steganographic content—as follows, where $P(S)$ is the probability of steganographic content in images, and $P(\neg S)$ is its complement. $P(D|S)$ is the probability that we'll detect an image that has steganographic content, and $P(D|\neg S)$ is the false-positive rate. Conversely, $P(\neg D|S) = 1 - P(D|S)$ is the false-negative rate. To improve the true-positive rate, we must increase the numerator or decrease the denominator. For a given detection system, increasing the detection rate is not possible without increasing the false-positive rate and vice versa. We assume that $P(S)$—the probability that an image

contains steganographic content—is extremely low compared to $P(\neg S)$, the probability that an image contains no hidden message. As a result, the false-positive rate $P(D|\neg S)$ is the dominating term in the equation; reducing it is thus the best way to increase the true-positive rate. Given these assumptions, the false-positive rate also dominates the computational costs to verifying hidden content. For a detection system to be practical, keeping the false-positive rate as low as possible is important.

VII. CONCLUSION

As Stegnography is one of the security technique, which is used to hide secrets in plain sight. In this paper I represents stegnography by various types of cover images like .bmp, .giff and .jpeg and out of them .jpeg is concluded to be the best one because as far as security concerns .jpeg is best cover media because when we try to embed message in .jpeg cover image then quality of image will not suffer at all i.e image will not be distorted. In this paper I work on steganalysis also by using various methods like JSteg, JP Hide and Outguess and I take two properties i.e negative or positive. This will be ranked if and only if msg. is not detected or is msg is detected. Out of various methods of steganalysis and JP Hide is prove to be the best method. But sender needs to be carefull regarding the length of meassge because as far I judge if message is large enough then it is easy to detect by any method so that's why I prefer that message which is to be hidden it must be of shorter length so that it is not easily detected. At last ultimately I colcluded from my research that Positive rate is as low as possible just to make secure communication between sender and reciver.

REFERENCES

[1]. R.J. Anderson and F.A.P. Petitcolas, "On the Limits of Steganography," J. Selected Areas in Comm., vol. 16, no. 4, 1998, pp. 474–481.

[2]. F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Information Hiding—A Survey," Proc. IEEE, vol. 87, no. 7, 1999, pp. 1062–1078.

[3]. J. Fridrich and M. Goljan, "Practical Steganalysis—State of the Art," Proc. SPIE Photonics Imaging 2002, Security and Watermarking of Multimedia Contents, vol. 4675, SPIE Press, 2002, pp. 1–13.

[4]. B. Chen and G.W. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," IEEE Trans. Information Theory, vol. 47, no. 4, 2001, pp. 1423–1443.

[5]. N.F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," Computer, vol. 31, no. 2, 1998, pp. 26–34.

[6]. A. Kerckhoffs, "La Cryptographie Militaire (Military Cryptography)," J. Sciences Militaires (J. Military Science, in French), Feb. 1883.

[7]. C. Cachin, An Information-Theoretic Model for Steganography, Cryptology ePrint Archive, Report 2000/028, 2002, www.zurich.ibm.com/˜cca/papers/stego.pdf.

[8]. A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," Proc. Information Hiding—3rd Int'l Workshop, Springer Verlag, 1999, pp. 61–76.

[9]. N.F. Johnson and S. Jajodia, "Steganalysis of Images Created Using Current Steganographic Software," Proc. 2nd Int'l Workshop in Information Hiding, Springer-Verlag, 1998, pp. 273–289.

[10]. H. Farid, "Detecting Hidden Messages Using Higher- Order Statistical Models," Proc. Int'l Conf. Image Processing, IEEE Press, 2002.

[11]. S. Lyu and H. Farid, "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines," Proc. 5th Int'l Workshop on Information Hiding, Springer-Verlag, 2002.

[12]. N. Provos, "Defending Against Statistical Steganalysis," Proc. 10th Usenix Security Symp., Usenix Assoc., 2001, pp. 323–335.

[13]. J. Kelley, "Terror Groups Hide Behind Web Encryption," *USA Today*, Feb. 2001, www.usatoday.com/life/cyber/ tech/2001-02-05-binladen.htm.

[14]. D. McCullagh, "Secret Messages Come in .Wavs," *Wired News*, Feb. 2001, www.wired.com/news/politics/ 0,1283,41861,00.html.

[15]. J. Kelley, "Militants Wire Web with Links to Jihad," *USA Today*, July 2002, www.usatoday.com/news/world/ 2002/07/10/web-terror-cover.htm.

[16]. N. Provos and P. Honeyman, "Detecting Steganographic Content on the Internet," *Proc. 2002 Network and Distributed System Security Symp.*, Internet Soc., 2002.

[17]. B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Fast Software Encryption, Cambridge Security Workshop Proc.*, Springer-Verlag, 1993, pp. 191–204.

[18]. A. Latham, "Steganography: JPHIDE and JPSEEK," 1999; http://linux01.gwdg.de/˜alatham/stego.html.

[19]. "The Internet Archive: Building an 'Internet Library'," 2001; www.archive.org.

[20]. S. Axelsson, "The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection," *Proc. 6th ACM Conf. Computer and Comm. Security*, ACM Press, 1999, pp. 1–7.