

Discovering Hidden Evidence

Chet Hosmer

WetStone Technologies, Inc.,
Cortland, New York, USA

ABSTRACT Over the past decade, the advancement of a myriad of methods, techniques and technologies to conceal digital evidence and covertly communicate have increased at an alarming rate. In addition, new information suggests that the download of an arsenal of software tools that perform these functions further suggests greater interest and usage of such cyber weapons. Steganography is here, and combined with the Internet and peer to peer networking, it provides criminals, gangs and terrorists with a viable and covert method of communication with guaranteed evidence concealment. This article discusses, in detail, the state-of-the-art in the most advanced Steganography tools and techniques available to perpetrators today. We include statistics regarding Steganography expansion, growth and usage, and discuss the specific digital forensic artifacts that help lead to discovery and extraction.

All of the image files used to develop this article are available for free download from the publisher's online edition of *Journal of Digital Forensic Practice*. Audio files used in the development of this article, additional resources including a detailed listing of all known Steganography programs, and related informational resources are available from the author.

KEYWORDS covert communication, covert messaging, data hiding, steganalysis, steganography, stego

Over the past decade, methods, techniques, and technologies to conceal digital evidence and communicate covertly have increased alarmingly. Steganography, the Internet, and peer-to-peer networking provide criminals, gangs, and terrorists with covert methods of communication and concealment of evidence.¹

Encryption protects the privacy of a message or data by allowing only those who hold a secret key to decipher the ciphertext. The critical distinction between steganography and encryption is that steganography conceals even the existence of the message or incriminating data.

There is legitimate doubt about how extensively steganography is used by criminals and spies,² but the technology for covert communication and evidence concealment is readily available. One can find over 300 steganography programs (including multiple versions or editions) on the Web providing

Chet Hosmer is the CEO and Chief Scientist at WetStone Technologies, Inc. He can be contacted by e-mail at chet@wetstonetech.com.

covert channels within digital images, digital audio, text, and TCP/IP traffic. Most sites offer free anonymous download of their wares and over half include source code, allowing for rapid development of derivative works.

Steganography programs allow the user to select a carrier, which is an original image or audio file, that they wish to use as the vector to carry the hidden data. The combination of carrier and payload creates what we call the covert message. What's vitally important about the covert message is that it resembles the carrier so closely that detailed examinations, visual and otherwise, do not reveal any clues that it contains hidden information.

The two images depicted in Figure 1 illustrate the power of steganography to conceal information in a sin-

gle digital image using STOOLES, a well-known steganography program. STOOLES, like virtually all steganography programs, obscures the process further by first compressing, then encrypting the payload using a password entered by the user. The image on the left is the original the image (the carrier), and the image on the right, (the covert message) contains the hidden payload. STOOLES allows you to visually compare your finished work in order to verify the quality of the vector. It is important to note that in this case the images have all the same physical characteristics and file properties. To the naked eye they appear identical on the screen. The two images do, however, contain differing binary values resulting in different one-way hash values.

Because the images in Figure 1 are raw, or *true color* images, the image file properties should, and will, match

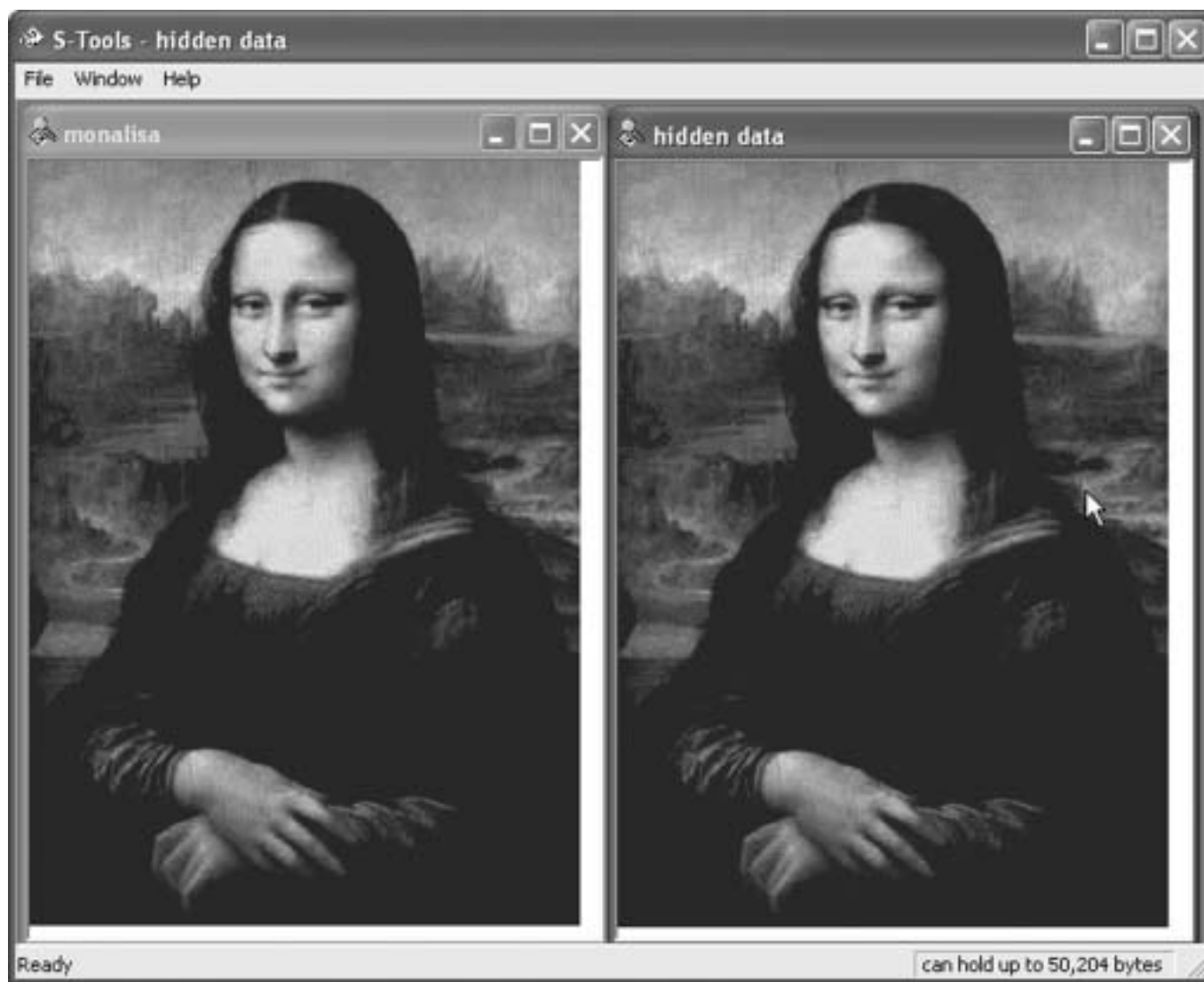


FIGURE 1 STOOLES example.

exactly. This is not universal for all images, however. When dealing with compressed formats such as GIF (Graphics Interchange Format) or JPEG (Joint Photographic Experts Group) some of the file properties will differ. For example, the number of unique colors used will be different, since the process of steganography creates altered colors when embedding the payload.

HOW DOES STEGANOGRAPHY WORK?

The images in Figure 1 are the simplest form of digital images to describe. True color images are also described as raw images. Each pixel in raw images contains all the information necessary to render that pixel. Each pixel has three bytes associated with the represented color on the screen. These correspond to the Red, Green and Blue (RGB) intensities assigned to the color.

The values range from 0 to 255, 0 representing no intensity and 255 representing high intensity for the selected RGB parameter. Thus an RGB value of 0, 0, 0 would result in the pixel color of black while the value of 255, 255, 255 would result in a pixel color of white. The possible combinations of values are computed as $2^8 \times 2^8 \times 2^8 \cong 16.8$ million possible color combinations for each pixel. By making slight modifications to the RGB value of each pixel (commonly referred to as the Least Significant Bit method, because only the LSB is changed), a new color is produced that is so close to the original that our eyes cannot discern the difference. Binary data can easily be hidden in the LSB values of any true color image.

In this example, only five of eight values had to be modified because three of the red pixel values were already in the correct state. Most steganography programs will compress or encrypt data prior to hiding. This process generates randomized binary data patterns that typically require modification of approximately 50% of the RGB values, making detection even more difficult. True Color LSB steganography doesn't change the file size because it simply alters values (no addition or deletion occurs).

INVESTIGATING TRUE COLOR IMAGES

Can we detect anomalies in images that would give us clues that steganography has been used? This short

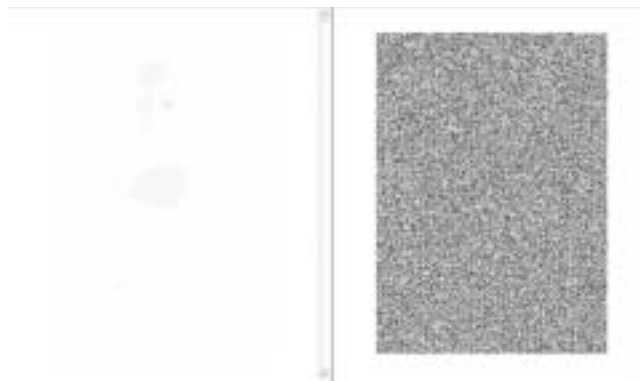


FIGURE 2 LSB rendering.

article does not delve into the details of the statistical and algorithmic detection methods used during steganalysis, but the following section summarizes methodologies for determining if steganography is in use.

What if you could visualize the image by not displaying all of the bits? Because we know that the LSB values, (or the “0” bit) are being altered (R, G, and B), we can simply filter out bits 7, 6, 5, 4, 3, 2, 1 and only render and display using bit 0 of the RGB values, since this is where the alteration allegedly took place. Figure 2 depicts this rendering.

The original carrier image is at the top and the rendering containing the covert message is below it. As you can see, the LSBs of the original image contain information relevant to the image; however, the rendering of the Stego'd image appears random—as we would suspect, because we are hiding compressed and encrypted data into these LSBs. For the selected image, we see very little data in the original image for the G or B values. This also is consistent with the absence of green and blue in the Mona Lisa image; however, the green and blue values of the stego'd image are rich with data.

Hiding Data in GIF Images

Graphic Interchange Format (GIF) documents are different than True Color representations. They use a palette represented by a single byte (8 bit) value, that serves as an index. This immediately reduces the size of the resulting image by approximately a third, even prior to compression. Palette images use a table of colors much like a painter's. Each pixel on the screen contains an index into the table or palette that defines

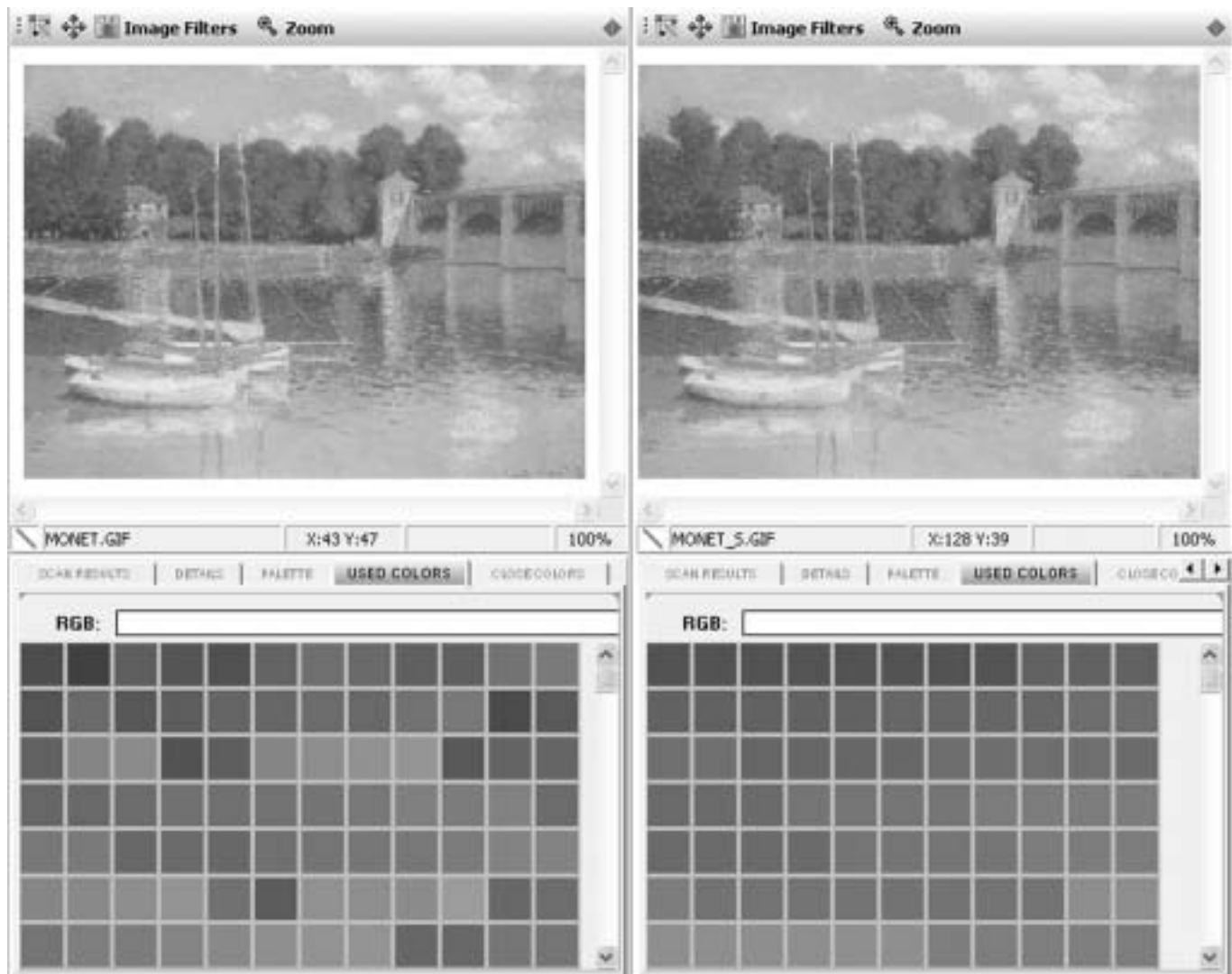


FIGURE 3 Close color pairs.

the color (R,G,B) value that will be displayed. The palette is limited in size (typically 256 colors). The colors contained in the palette can be any combination of R,G,B values - in other words any of 16.8 million colors - but only 256 can be used at a time. There are many methods for embedding within palette-based images. Here, I discuss the simplest method, which is typically referred to as *color reduction* or *close color pair creation*.

Color reduction techniques utilize colors that are close to existing colors in the original carrier image and store them in the palette by either replacing palette colors with low incidence in the image or using unused or unreferenced palette indices. The pixels that referenced the cloned color can display as either index, providing a very close rendering of the image. Thus one reference could represent an embedded bit value of one and the

second index could represent a bit value of zero. By strategically selecting index values (original or the slightly altered clone), we can toggle 1s and 0s enabling information hiding within the image. This approach can render images that look very similar. However, by examining the palette instead of the actual image, we can detect anomalies in the form of what are called close color pairs (Figure 3).

In order to produce the best possible image with only 256 colors, the GIF's palette will typically contain a diverse set of colors, which will allow for representation of the range of colors present in the original image. Software that converts (or compresses) true color images into palette based images would not normally select colors that are only a single LSB apart, as they have goal-seeking algorithms that seek to maximize the available color palette. By examining the



FIGURE 4 Image distortion caused by palette color reduction.

palette of suspect images we can quickly identify these artifacts generated by steganography algorithms. Additionally, palette-based images that have steganography applied will often have slightly distorted visual artifacts, which are the result of the color reduction (Figure 4).

Hiding Data in JPEG Images

The methods of insertion and detection of steganography are more complex in the case of JPEG³ images. JPEG images are “Lossy Compressed” meaning that information is lost during the compression process. When you convert a JPEG file back to a true color rendering, such as when you display or print the image, the original image obviously cannot be returned, only an approximation is displayed. Therefore hiding information in the previously described manner for GIF’s and true color images is not possible because the “lossiness” of the encoding process would also destroy the payload.

The compression method employed for JPEG images utilizes a mathematical function called a Discrete Cosine Transform or DCT. Each image is broken up into 8×8 pixel blocks, and coefficients are calculated and stored for the DCT that approximates the true color rendering for the block. The coefficients are then quantized (given a numeric value to represent the entire block) and stored as shown in Figure 5. The steganographer applies small changes to these quantized coefficient values (the circled numbers in Figure 5), slightly altering the rendering of the block, with unnoticeable results. Because the original quantization pro-

	1	2	3	4	5	6	7	8
1	DC	1	5	6	14	15	27	28
2	2	4	7	13	16	26	29	42
3	3	8	12	17	25	30	41	43
4	9	11	18	24	31	40	44	53
5	10	19	23	32	39	45	52	54
6	20	22	33	38	46	51	55	60
7	21	34	37	47	50	56	59	61
8	35	36	48	49	57	58	62	63

Quantized Coefficients Selected for Altering

FIGURE 5 Quantized DCT coefficients.

cess was an approximation, this new approximation is usually quite good. Although the quantization table goes through one more round of compression, it is lossless (no information is lost), and the altered values will remain constant. As with GIF images, this final stage of compression will alter the file size from the original, thus the carrier file size does not equal the size of the covert message.

Investigating JPEG steganography proves to be very difficult and challenging. One method is to examine what are called 2nd order artifacts, such as the hue and saturation of the image. Figure 6 shows one type of distortion that can occur to these properties when modifications are made to original quantized values. These distortions are caused because most cameras and software attempt to “normalize” the image characteristics

for hue, as an example. This is especially obvious in areas of a sky. We call these 2nd order artifacts because they become apparent after the image has been converted from the stego'd quantized values (1st order) into an RGB rendering of hue (2nd order).

In order to examine the first order artifacts, we must examine the quantization table where the modifications are actually made by the steganography algorithms. One very effective method is to generate a histogram of quantization values for a particular image. Figure 7 shows histograms representing the carrier and the covert messages. At first glance the histograms look similar; however, by zooming in on one of the peaks more closely we notice some obvious changes. The peak for the carrier image is much sharper and has a larger discrete value than that of the stego'd image. This would be expected as each block should represent a single value that takes the adjacent values into account. However in steganography, modifications are being made (+1 or -1) for certain quantization values. Peak values represent the highest number of occurrences for specific values found in the quantization tables. Because applying steganography to the "normal" values results in fewer occurrences at the peak and additional occurrences in the values immediately adjacent to the "normal" value, the peak tends to be lowered and wider, due to the adjacent values. These adjustments are what cause these artifacts, as you can see in this illustration.

Hiding Data in Music Files

Do music files offer a viable storage medium for hidden information? If so, the explosion of the popularity of portable audio devices like MP3 players, and the downloadability of virtually every song ever made, has created a potentially huge haystack of music—on the Internet, on personal computers, and within handheld devices. As with digital images, LSB encoding of digital audio is potentially a considerable threat. However, there are some significant problems associated with using this technique for audio files.

First, our sense of hearing is quite good at detecting noise and distortion within the audible range. However, a slight amplitude or phase shift of the signal is more difficult to discern. Digital audio is recorded by sampling the analog signal produced by audio and then documenting the samples. Digital Signal Processors, or DSPs, are capable of providing high-resolution sampling of audio signals. The question becomes just how many samples are enough?

The sampling rate that was chosen for CD quality audio or WAV files is roughly twice the human audible range ($22.5 \text{ kHz} \times 2$) or 44,100 samples per second. This is based on the Nyquist theorem.⁴ Each sample is recorded as a signed 16-bit value ranging from -32767 to $+32768$.

The sample size for audio is one order of magnitude larger than that of images. We have 8-bit RGB values versus 16-bit audio samples, meaning that only minute changes to the audio signal will occur when LSB steganography is applied. Figure 8 depicts the waveform of the original carrier and the covert message. There is very little observable difference. Also, in our stereo-based world, we have a left and a right channel to use. The math gives us 44,100 samples per second times two channels, yielding 88,200 samples for every 1 second of audio to hide or conceal information. Performing the necessary calculations, using LSB steganography, we can easily hide the entire Old Testament in a single 6-minute song.

In this example we chose a payload that would cause every sample to be altered. Yet there is no audible difference between the recordings (as this is a printed publication, you'll have to take my word for it). Visual examination of the WAV file reveals no obvious clues, as seen in Figure 8.

However, one issue that provides us clues are the areas of silence. By examining the silence area in the first 100 samples, as shown in Figure 9, it is quite obvious that binary data has been recorded in the silent portion of the audio. For each of the samples altered in the silence area of the song, the sample values are discretely either 0 or 1. This is a strong indication that the file may contain steganography. From this illustration we can see that this particular steganography program only alters the LSB of the audio samples. More sophisticated statistical modeling techniques can be used in order to detect steganography where this obvious flaw in the first 100 samples has been corrected by the steganographer.

MP3 and a number of other audio formats employs a lossy compression method, similar to what we saw with JPEG images. In these formats the steganographer stores information as part of the compressed data after the lossy stage, but before the lossless stage. Because these compressed formats yield an approximation of the original, many options exist for compression, resulting in different recorded results. There are many formats for ripped audio that use differing lossy encoding methods, making the challenge even greater. Because of this complexity, steganalysis of ripped audio is an ongoing research challenge.

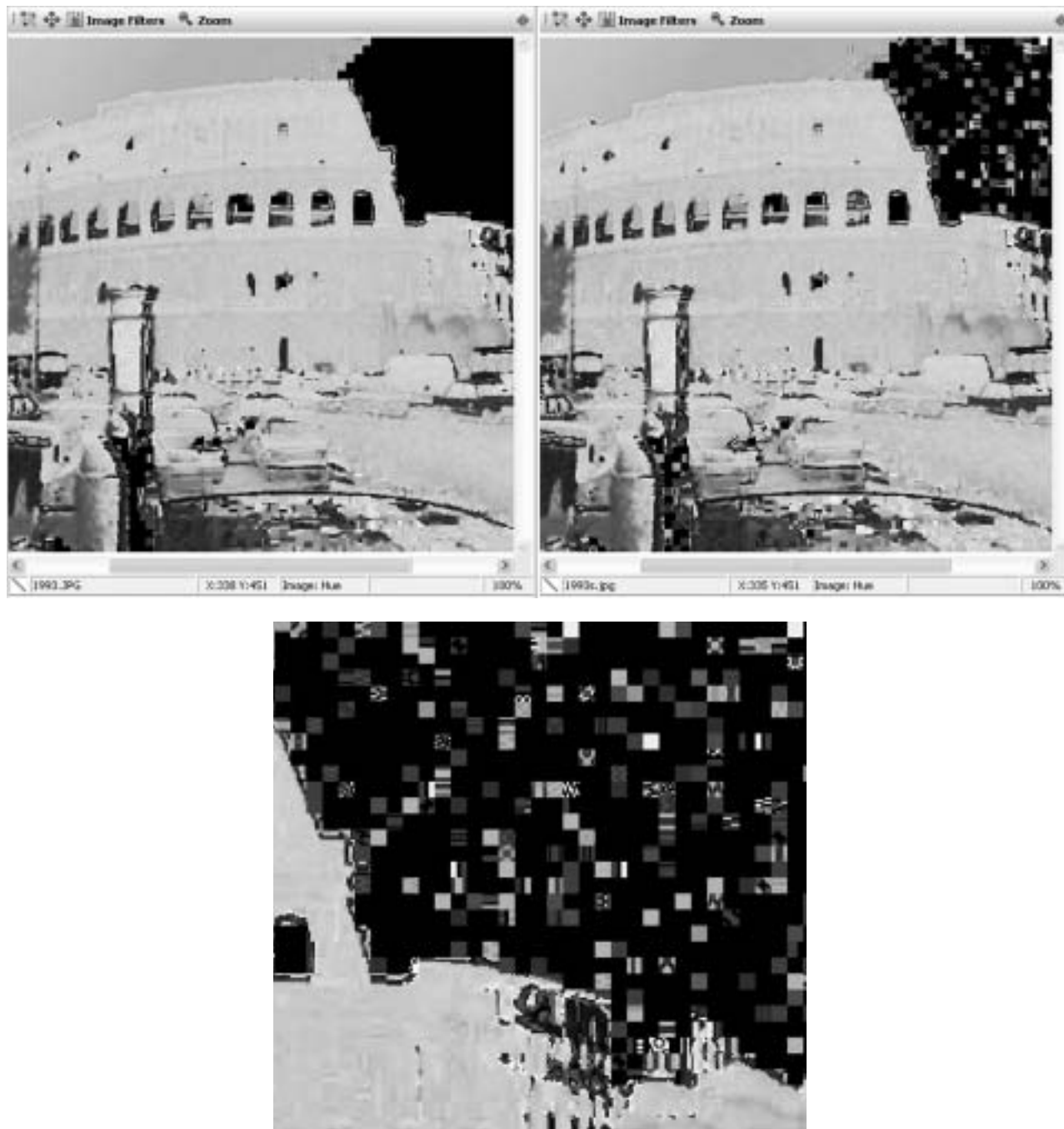


FIGURE 6 Hue distortion caused by DCT coefficient modification.

THE ART OF STEGANOGRAPHY

The number and sophistication of digital steganography methods continues to accelerate. The most recent tools and techniques now include countermeasures to statistical detection. In the past, the object was to simply fool the senses by making sure that the object presented would look or sound exactly as it should. Today, the latest hiding methods must consider the statistical significance of the changes the steganography program

is making, and adjust accordingly. Further, steganography programs predict the amount of information that can be hidden within a selected carrier, based on the size of the payload, the size of the carrier, and the suitability of the carrier to conceal information and avoid detection.

Steganography is moving beyond digital images and digital audio. Digital video utilizes carriers that offer significant cover for those attempting to hide larger and

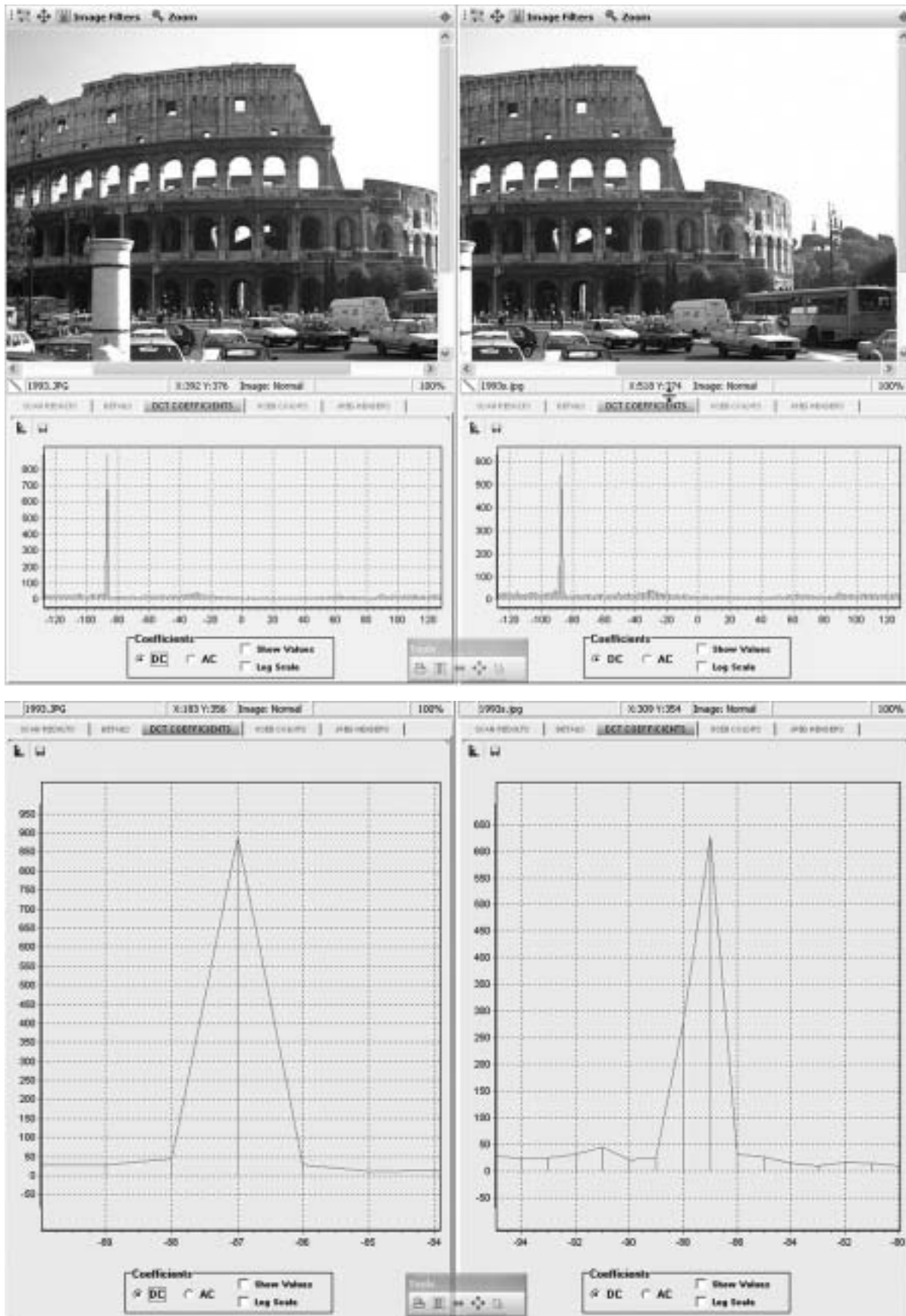


FIGURE 7 DCT coefficient histograms.

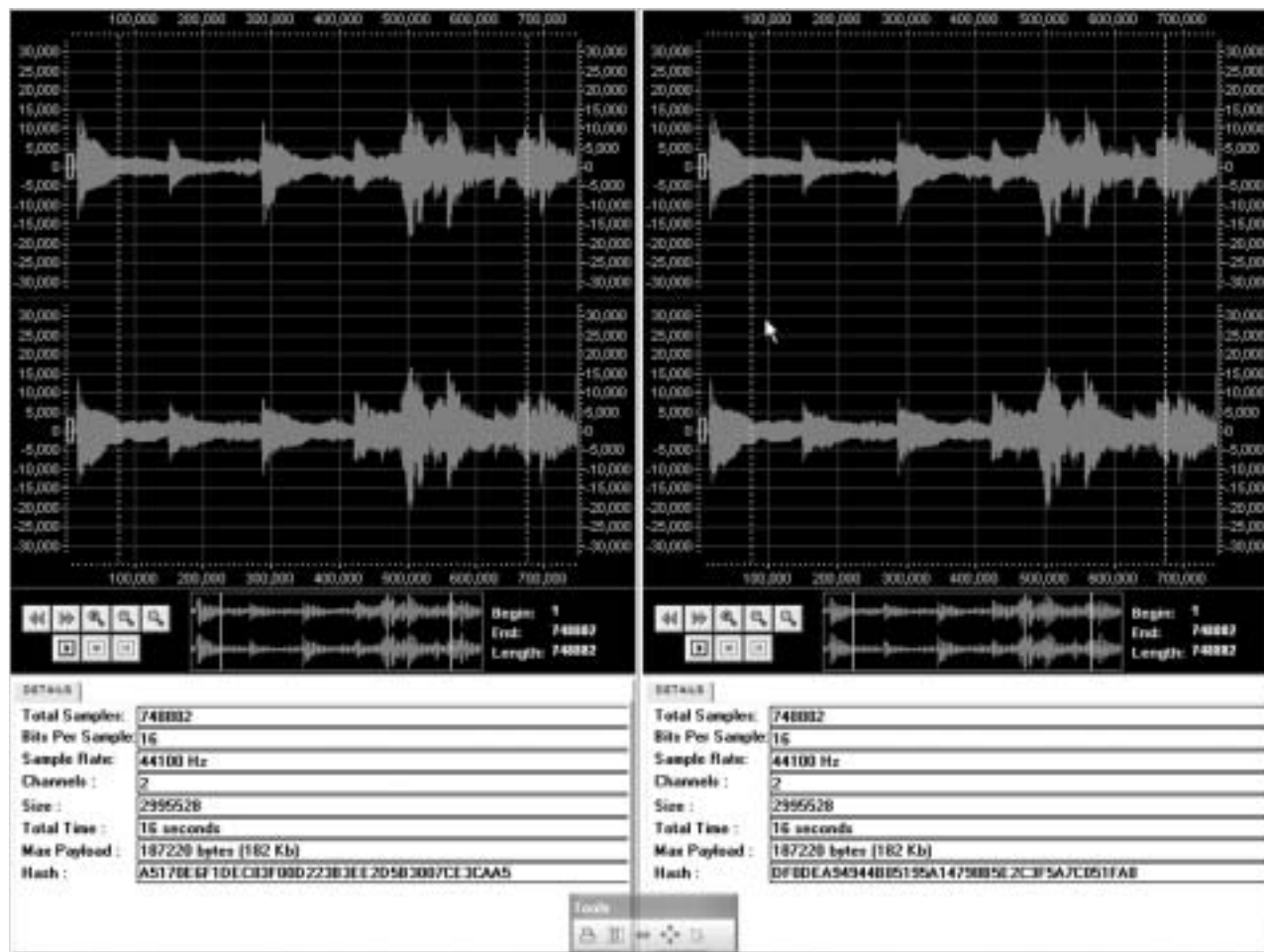


FIGURE 8 Digital audio sample rendering.

larger payloads for communication or concealment. As the use of these carriers, as well as the bandwidth increases, video steganography may become a significant issue. We enjoy being able to carry thousands of songs or dozens of movies with us in our iPods. How many of these carry a hidden payload?

The state of the art in steganography continues to advance. It has done so since its predigital infancy over 2000 years ago. It has been used, in one form or another, during conflict or discourse throughout the millennia. Its continuing evolution is not merely likely, but guaranteed.

What Practical Steps Can Investigators Take?

All investigators should be aware of the concealment methods used in steganography and how they work. Consideration should be given to routinely search for the presence of known steganography programs as part

of your standard operating procedure. Identifying these will provide you with two valuable pieces of information:

1. Knowledge that the suspect is familiar with and using known steganography programs.
2. Insight into the type of steganography they are using.

This knowledge will help in narrowing the collection and examination of possible carriers. For example if they are using a steganography tool that only creates JPEG covert messages, you can initially focus your examination on JPEG files that exist within the seized evidence.

Once you have determined that steganography is in play, you know more about the potential sophistication of the suspect and need to investigate their motive and intent for concealing the existence of information. This knowledge should help during the questioning of the suspect.

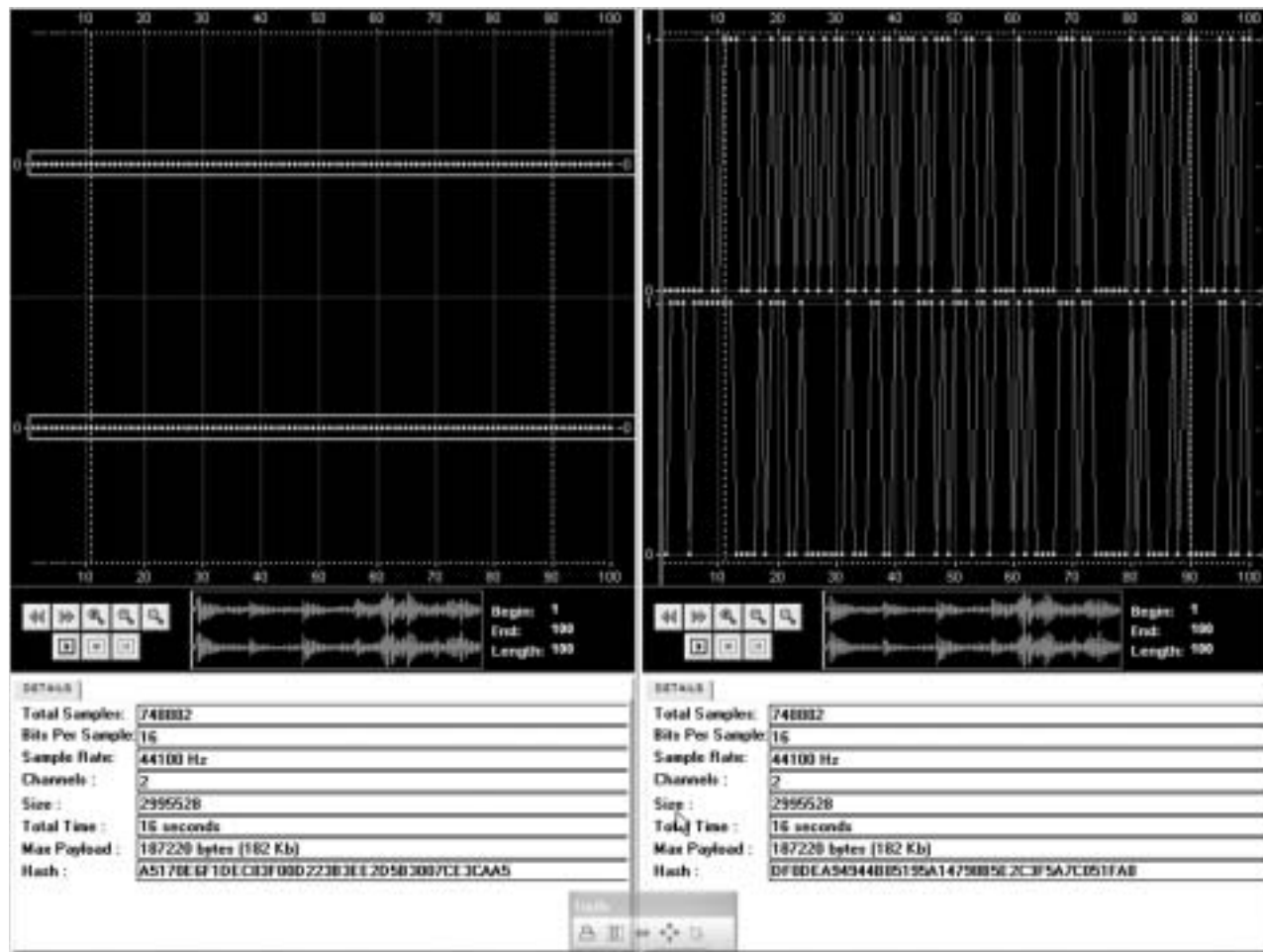


FIGURE 9 Digital audio silence area containing steganography.

Begin your analysis of the suspect digital images and audio files that might contain hidden information. Isolate the highly suspicious images and audio files and begin the process of attempting to crack the images. If you were able to determine the steganography tool that was used you can begin to use the tool itself to attempt to uncover the hidden information from the suspect images or audio files. You must guess the password at this point if the suspect used one, but at this point you may have other information regarding the suspects use of passwords that may help reduce the set of passwords you need to try. Once you have successfully guessed one password, you will be able to uncover the hidden payload as well as other password protected files.

It will not always be possible to recover the hidden information. However, determining that steganography programs are in use, and accurately identifying

which images or audio files likely contain the hidden information may provide critical evidence regarding the suspect's capabilities, motives, and intent to conceal.

NOTES

1. For an introductory overview of steganography, see "Steganography: Seeing the Unseen" by Neil F. Johnson and Sushil Jajodia. IEEE Computer, February 1998: 26–34. Available as a PDF download from <http://www.jjtc.com/pub/r2026.pdf>.
2. Bagnall, R. J. (2002). "Reversing the Steganography Myth in Terrorist Operations: The Asymmetrical Threat of Simple Intelligence Dissemination Techniques Using Common Tools." Information Security Reading Room (SANS Institute). Download as PDF from <http://www.sans.org/rrr/whitepapers/steganography/556.php>.
3. JPEG stands for the Joint Photographic Experts Group, see: <http://www.jpeg.org/>.
4. <http://cnx.rice.edu/content/m10791/latest/>.