
Hiding Information in Images

The surprises in text documents are mostly things of which the authors were ignorant or unaware. Image documents provide unlimited opportunities for hiding things intentionally—hiding secrets from casual human observers, and obscuring open messages destined for human recipients so anti-spam software won't filter them out.

The Spam Wars

Many of us are used to receiving email pleas such as this one: *I am Miss Faatin Rahman the only child/daughter of late mrs helen rahman Address: Rue 142 Marcory Abidjan Cote d'ivoire west africa, I am 20 years old girl. I lost my parent, and I have an inheritance from my late mother, My parents were very wealthy farmers and cocoa merchant when they were alive, After the death of my father, long ago, my mother was controlling his business untill she was poisoned by her business associates which she suffered and died, ... I am crying and seeking for your kind assistance in the following ways: To provide a safe bank account into where the money will be transferred for investment....*

If you get such a request, don't respond to it! Money will flow out of, not into, your bank account. Most people know not to comply. But mass emails are so cheap that getting one person out of a million to respond is enough to make the spammer financially successful.

"Spam filters" are programs that intercept email on its way into the in-box and delete messages like these before we read them. This kind of spam follows such a standard style that it is easy to spot automatically, with minimal risk that any real correspondence with banks or African friends will be filtered out by mistake.

But the spam artists have fought back. Many of us have received emails like the one in Figure 3.14. Why can't the spam filter catch things like this?

Word-processing software includes the name and size of the font in conjunction with the coded characters themselves, as well as other information, such as the color of the letters and the color of the background. Because the underlying text is represented as ASCII codes, however, it remains relatively easy to locate individual letters or substrings, to add or delete text, and to perform other such common text-processing operations. When a user positions a cursor over the letter on the screen, the program can figure out the location within the file of the character over which the cursor is positioned. Computer software can, in turn, render the character codes as images of characters.

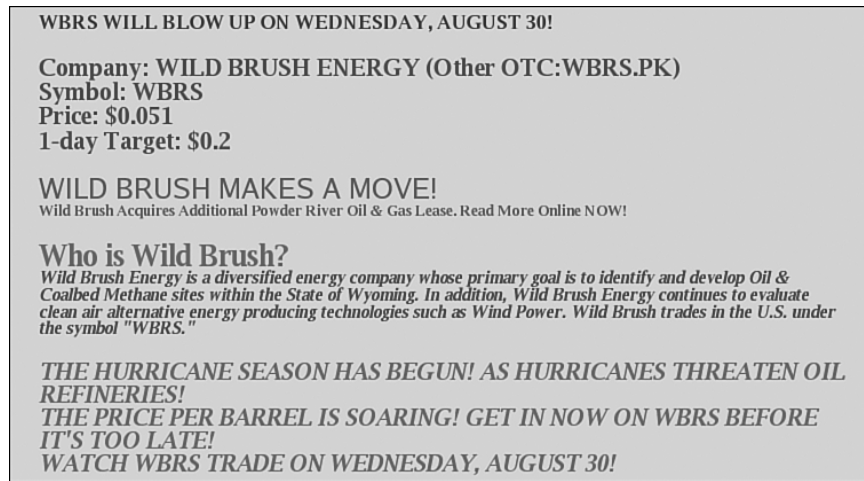


FIGURE 3.14 Graphic spam received by one of the authors. Although it looks like text, the computer “sees” it as just an image, like a photograph. Because it doesn’t realize that the pixels are forming letters, its spam filters cannot identify it as spam.

But just because a computer screen shows a recognizable letter of the alphabet, this does not mean that the underlying representation is by means of standard character codes. A digitized photograph of text may well look identical to an image rendered from a word-processing document—that is, the two utterly different representations may give rise to exactly the same image.

And that is one reason why, in the battle between spam producers and makers of spam filters, the spam producers currently have the upper hand. The spam of Figure 3.14 was produced in graphical form, even though what is represented is just text. As the underlying representation is pixels and not ASCII, spam like this makes it through all the filters we know about!

The problem of converting raster graphics to ASCII text is called *character recognition*. The term *optical character recognition*, or OCR, is used when the original document is a printed piece of paper. The raster graphic representation is the result of scanning the document, and then some character recognition algorithm is used to convert the image into a sequence of character codes. If the original document is printed in a standard typeface and is relatively free of smudges and smears, contemporary OCR software is quite accurate, and is now incorporated into commercially available scanners commonly packaged as multipurpose devices that also print, photocopy, and fax. Because OCR algorithms are now reasonably effective and widely available, the next generation of spam filters will likely classify emails such as Figure 3.14 as spam.

OCR and spam are merely an illustration of a larger point. Representation determines what can be done with data. In principle, many representations may be equivalent. But in practice, the secrecy of formatting information and the computation required to convert one format to another may limit the usefulness of the data itself.

Hiding Information in Plain Sight

During World War I, the German Embassy in Washington, DC sent a message to Berlin that began thus: “PRESIDENT’S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE.” U.S. intelligence was reading all the German telegrams, and this one might have seemed innocuous enough. But the first letters of the words spelled out “PERSHING,” the name of a U.S. Navy vessel. The entire telegram had nothing to do with embargoes. It was about U.S. ship movements, and the initial letters read in full, “PERSHING SAILS FROM N.Y. JUNE 1.”

Steganography is the art of sending secret messages in imperceptible ways. Steganography is different from *cryptography*, which is the art of sending messages that are indecipherable. In a cryptographic communication, it is assumed that if Alice sends a message to Bob, an adversary may well intercept the message and recognize that it holds a secret. The objective is to make the message unreadable, except to Bob, if it falls into the hands of such an eavesdropper or enemy. In the world of electronic communication, sending an encrypted message is likely to arouse suspicion of electronic monitoring software. By contrast, in a steganographic message from Alice to Bob, the communication itself arouses no suspicion. It may even be posted on a web site and seem entirely innocent. Yet hidden in plain sight, in a way known only to Alice and Bob, is a coded message.

Steganography has been in use for a long time. The *Steganographia* of Johannes Trithemius (1462–1516) is an occult text that includes long conjurations of spirits. The first letters of the words of these mystic incantations encode other hidden messages, and the book was influential for a century after it was written. Computers have created enormous opportunities for steganographic communications. As a very simple example, consider an ordinary word-processing document—a simple love letter, for example. Print it out or view it on the screen, and it seems to be about Alice’s sweet nothings to Bob, and nothing more. But perhaps Alice included a paragraph at the end *in which she changed the font color to white*. The software renders the white text on the white background, which looks exactly like the white background.

But Bob, if he knows what to look for, can make it visible—for example, by printing on black paper (just as the text could be recovered from the electronically redacted Calipari report).

If an adversary has any reason to think a trick like this might be in use, the adversary can inspect Alice’s electronic letter using software that looks for messages hidden using just this technique. But there are many places to look for steganographic messages, and many ways to hide the information.

Since each Roman letter has an eight-bit ASCII code, a text can be hidden within another as long as there is an agreed-upon method for encoding 0s and 1s. For example, what letter is hidden in this sentence?

Steganographic algorithms hide messages inside photos, text, and other data.

The answer is “I,” the letter whose ASCII character code is `01001001`. In the first eight words of the sentence, words beginning with consonants encode 0 bits and words beginning with vowels encode 1s (see Figure 3.15).

Steganographic	algorithms	hide	messages	inside	photos,	text,	and	other	data.
0	1	0	0	1	0	0	1		

FIGURE 3.15 A steganographic encoding of text within text. Initial consonants encode 0, vowels encode 1, and the first eight words encode the 8-bit ASCII code for the letter “I.”

A steganographic method that would seem to be all but undetectable involves varying ever so slightly the color values of individual pixels within a photograph. Red, green, and blue components of a color determine the color itself. A color is represented internally as one byte each for red, green, and blue. Each 8-bit string represents a numerical value between 0 and 255. Changing the rightmost bit from a 1 to a 0 (for example, changing `00110011` to `00110010`), changes the numerical value by subtracting one—in this case, changing the color value from 51 to 50. That results in a change in color so insignificant that it would not be noticed, certainly not as a change in a single pixel. But the rightmost bits of the color values of pixels in the graphics files representing photographs can then carry quite large amounts of information, without raising any suspicions. The recipient decodes the message not by rendering the bits as visible images, but by inspecting the bits themselves, and picking out the significant 0s and 1s.

Who uses steganography today, if anyone? It is very hard to know. *USA Today* reported that terrorists were communicating using steganography in early 2001. A number of software tools are freely available that make steganography easy. Steganographic detectors—what are properly known as steganalysis tools—have also been developed, but their usefulness as yet seems to be limited. Both steganography and steganalysis software is freely available on the World Wide Web (see, for example, www.cotse.com/tools/stega.htm and www.outguess.org/detection.php).

The use of steganography to transmit secret messages is today easy, cheap, and all but undetectable. A foreign agent who wanted to communicate with parties abroad might well encode a bit string in the tonal values of an MP3 or the color values of pixels in a pornographic image on a web page. So much music and pornography flows between the U.S. and foreign countries that the uploads and downloads would arouse no suspicion!

The Scary Secrets of Old Disks

By now, you may be tempted to delete all the files on your disk drive and throw it away, rather than run the risk that the files contain unknown secrets. That isn't the solution: Even deleted files hold secrets!

A few years ago, two MIT researchers bought 158 used disk drives, mostly from eBay, and recovered what data they could. Most of those who put the disks up for sale had made some effort to scrub the data. They had dragged files into the desktop trash can. Some had gone so far as to use the Microsoft Windows FORMAT command, which warns that it will destroy all data on the disk.

Yet only 12 of the 158 disk drives had truly been sanitized. Using several methods well within the technical capabilities of today's teenagers, the researchers were able to recover user data from most of the others. From 42 of the disks, they retrieved what appeared to be credit card numbers. One of the drives seemed to have come from an Illinois automatic teller machine and contained 2,868 bank account numbers and account balances. Such data from single business computers would be a treasure trove for criminals. But most of the drives from home computers also contained information that the owners would consider extremely sensitive: love letters, pornography, complaints about a child's cancer therapy, and grievances about pay disputes, for example. Many of the disks contained enough data to identify the primary user of the computer, so that the sensitive information could be tied back to an individual whom the researchers could contact.