

## 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.

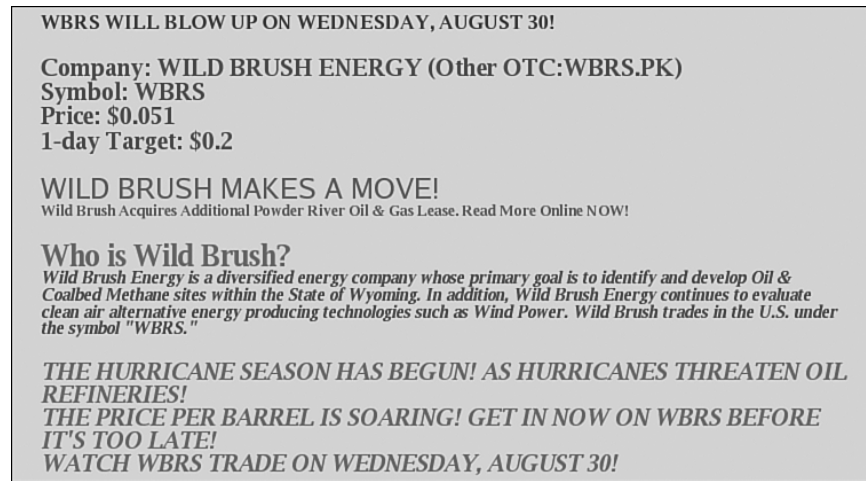


FIGURE 3.14 Graphic spam received by one of the authors. Although it looks like text, the computer “sees” it as just an image, like a photograph. Because it doesn’t realize that the pixels are forming letters, its spam filters cannot identify it as spam.

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](http://www.cotse.com/tools/stega.htm) and [www.outguess.org/detection.php](http://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.

**CLOUD COMPUTING**

One way to avoid having problems with deleted disk files and expensive document-processing software is not to keep your files on your disks in the first place! In "cloud computing," the documents stay on the disks of a central service provider and are accessed through a web browser. "Google Docs" is one such service, which boasts very low software costs, but other major software companies are rumored to be exploring the market for cloud computing. If Google holds your documents, they are accessible from anywhere the Internet reaches, and you never have to worry about losing them—Google's backup procedures are better than yours could ever be. But there are potential disadvantages. Google's lawyers would decide whether to resist subpoenas. Federal investigators could inspect bits passing through the U.S., even on a trip between other countries.

The users of the computers had for the most part done what they thought they were supposed to do—they deleted their files or formatted their disks. They probably knew not to release toxic chemicals by dumping their old machines in a landfill, but they did not realize that by dumping them on eBay, they might be releasing personal information into the digital environment. Anyone in the world could have bought the old disks for a few dollars, and all the data they contained. What is going on here, and is there anything to do about it?

Disks are divided into blocks, which are like the pages of a book—each has an identifying address, like a page number, and is able to hold a few hundred bytes of data, about the same amount as a page of text in a book. If a document is larger than one disk block, however, the document is typically not stored in consecutive disk blocks. Instead, each block includes a piece of the document, and the address of the block where the document is continued. So

the entire document may be physically scattered about the disk, although logically it is held together as a chain of references of one block to another. Logically, the structure is that of a magazine, where articles do not necessarily occupy contiguous pages. Part of an article may end with "Continued on page 152," and the part of the article on page 152 may indicate the page on which it is continued from there, and so on.

Because the files on a disk begin at random places on disk, an *index* records which files begin where on the disk. The index is itself another disk file, but one whose location on the disk can be found quickly. A disk index is very much like the index of a book—which always appears at the end, so readers know where to look for it. Having found the index, they can quickly find the page number of any item listed in the index and flip to that page.

Why aren't disks themselves organized like books, with documents laid out on consecutive blocks? Because disks are different from books in two important respects. First, they are dynamic. The information on disks is constantly being altered, augmented, and removed. A disk is less like a book than like a three-ring binder, to which pages are regularly added and removed as information is gathered and discarded. Second, disks are perfectly re-writable. A disk block may contain one string of 0s and 1s at one moment, and as a result of a single writing operation, a different string of 0s and 1s a moment later. Once a 0 or a 1 has been written in a particular position on the disk, there is no way to tell whether the bit previously in that position was a 0 or a 1. There is nothing analogous to the faint traces of pencil marks on paper that are left after an erasure. In fact, there is no notion of "erasure" at all on a disk—all that ever happens is replacement of some bits by others.

Because disks are dynamic, there are many advantages to breaking the file into chained, noncontiguous blocks indexed in this way. For example, if the file contains a long text document and a user adds a few words to the middle of the text, only one or two blocks in the middle of the chain are affected. If enough text is added that those blocks must be replaced by five new ones, the new blocks can be logically threaded into the chain without altering any of the other blocks comprising the document. Similarly, if a section of text is deleted, the chain can be altered to "jump over" the blocks containing the deleted text.

Blocks that are no longer part of any file are added to a "pool" of available disk blocks. The computer's software keeps track of all the blocks in the pool. A block can wind up in the pool either because it has never been used or because it has been used but abandoned. A block may be abandoned because the entire file of which it was part has been deleted or because the file has been altered to exclude the block. When a fresh disk block is needed for any purpose—for example, to start a new file or to add to an existing file—a block is drawn from the pool of available blocks.

### ***What Happens to the Data in Deleted Files?***

*Disk blocks are not re-written when they are abandoned and added to the pool.* When the block is withdrawn from the pool and put back to work as part of another file, it is overwritten and the old data is obliterated. But until then, the block retains its old pattern of zeroes and ones. The entire disk file may be intact—except that there is no easy way to find it. A look in the index will reveal nothing. But "deleting" a file in this way merely removes the index entry. The information is still there on the disk somewhere. It has no more

been eradicated than the information in a book would be expunged by tearing out the index from the back of the volume. To find something in a book without an index, you just have to go through the book one page at a time looking for it—tedious and time-consuming, but not impossible.

And that is essentially what the MIT researchers did with the disks they bought off eBay—they went through the blocks, one at a time, looking for recognizable bit patterns. A sequence of sixteen ASCII character codes representing decimal digits, for example, looks suspiciously like a credit card number. Even if they were unable to recover an entire file, because some of the blocks comprising it had already been recycled, they could recognize significant short character strings such as account numbers.

Of course, there would be a simple way to prevent sensitive information from being preserved in fragments of “deleted” files. The computer could be programmed so that, instead of simply putting abandoned blocks into the

#### THE LAW ADJUSTS

Awareness is increasing that deleted data can be recovered from disks. The Federal Trade Commission now requires “the destruction or erasure of electronic media containing consumer information so that the information cannot practically be read or reconstructed,” and a similar provision is in a 2007 Massachusetts Law about security breaches.

pool, it actually over-wrote the blocks, perhaps by “zeroing” them—that is, writing a pattern of all 0s. Historically, computer and software manufacturers have thought the benefits of zeroing blocks far less than the costs. Society has not found “data leakage” to be a critical problem until recently—although that may be changing. And the costs of constantly zeroing disk blocks would be significant. Filling blocks with zeroes might take so much time that the users would complain about how slowly their machines were running

if every block were zeroed immediately. With some clever programming the process could be made unnoticeable, but so far neither Microsoft nor Apple has made the necessary software investment.

And who has not deleted a file and then immediately wished to recover it? Happily for all of us who have mistakenly dragged the wrong file into the trash can, as computers work today, deleted files are not immediately added to the pool—they can be dragged back out. Files can be removed only until you execute an “Empty trash” command, which puts the deleted blocks into the pool, although it does not zero them.

But what about the Windows “FORMAT” command, shown in Figure 3.16? It takes about 20 minutes to complete. Apparently it is destroying all the bits on the disk, as the warning message implies. But that is not what is happen-



ing. It is simply looking for faulty spots on the disk. Physical flaws in the magnetic surface can make individual disk blocks unusable, even though mechanically the disk is fine and most of the surface is flawless as well. The FORMAT command attempts to *read* every disk block in order to identify blocks that need to be avoided in the future. Reading every block takes a long time, but rewriting them all would take twice as long. The FORMAT command identifies the bad blocks and re-initializes the index, but leaves most of the data unaltered, ready to be recovered by an academic researcher—or an inventive snooper.

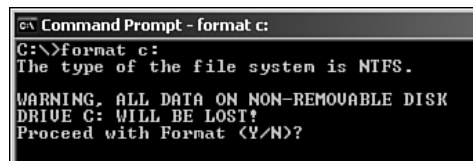


FIGURE 3.16 Warning screen of Microsoft Windows FORMAT command. The statement that all the data will be lost is misleading—in fact, a great deal of it can be recovered.

As if the problems with disks were not troubling enough, exactly the same problems afflict the memory of cell phones. When people get rid of their old phones, they forget the call logs and email messages they contain. And if they do remember to delete them, using the awkward combinations of button-pushes described deep in the phone's documentation, they may not really have accomplished what they hoped. A researcher bought ten cell phones on eBay and recovered bank account numbers and passwords, corporate strategy plans, and an email exchange between a woman and her married boyfriend, whose wife was getting suspicious. Some of this information was recovered from phones whose previous owners had scrupulously followed the manufacturer's instructions for clearing the memory.

#### SOFTWARE TO SCRUB YOUR DISK

If you really want to get rid of all the data on your disk, a special "Secure empty trash" command is available on Macintosh computers. On Windows machines, DBAN is free software that really will zero your disk, available through [dban.sourceforge.net](http://dban.sourceforge.net), which has lots of other useful free software. Don't use DBAN on your disk until you are sure you don't want anything on it anymore!

In a global sense, bits turn out to be very hard to eradicate. And most of the time, that is exactly the way we want it. If our computer dies, we are glad that Google has copies of our data. When our cell phone dies, we are happy if our contact lists reappear, magically downloaded from our cellular service provider to our replacement phone. There are upsides and downsides to the persistence of bits.

Physical destruction always works as a method of data deletion. One of us uses a hammer; another of us prefers his axe. Alas, these methods, while effective, do not meet contemporary standards for recovery and recycling of potentially toxic materials.

### ***Can Data Be Deleted Permanently?***

#### **COPIES MAKE DATA HARD TO DELETE**

If your computer has ever been connected to a network, destroying its data will not get rid of copies of the same information that may exist on other machines. Your emails went to and from other people—who may have copies on their machines, and may have shared them with others. If you use Google's Gmail, Google may have copies of your emails even after you have deleted them. If you ordered some merchandise online, destroying the copy of the invoice on your personal computer certainly won't affect the store's records.

Rumors arise every now and then that engineers equipped with very sensitive devices can tell the difference between a 0 that was written over a 0 on a disk and a 0 that was written over a 1. The theory goes that successive writing operations are not perfectly aligned in physical space—a “bit” has width. When a bit is rewritten, its physical edges may slightly overlap or fall short of its previous position, potentially revealing the previous value. If such microscopic misalignments could be detected, it would be possible to see, even on a disk that has been zeroed, what the bits were *before* it was zeroed.

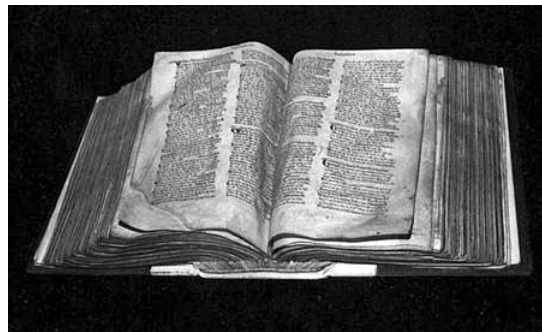
No credible authentication of such an achievement has ever been published, however, and as the density of hard disks continues to rise, the like-

lihood wanes that such data recovery can be accomplished. On the other hand, the places most likely to be able to achieve this feat are government intelligence agencies, which do not boast of their successes! So all that can be said for certain is that recovering overwritten data is within the capabilities of at most a handful of organizations—and if possible at all, is so difficult and costly that the data would have to be extraordinarily valuable to make the recovery attempt worthwhile.

### ***How Long Will Data Really Last?***

As persistent as digital information seems to be, and as likely to disclose secrets unexpectedly, it also suffers from exactly the opposite problem. Sometimes electronic records become unavailable quite quickly, in spite of best efforts to save them permanently.

Figure 3.17 shows an early geopolitical and demographic database—the Domesday Book, an inventory of English lands compiled in 1086 by Norman monks at the behest of William the Conqueror. The Domesday Book is one of Britain's national treasures and rests in its archives, as readable today as it was in the eleventh century.



British National Archives.

FIGURE 3.17 The Domesday Book of 1086.

In honor of the 900th anniversary of the Domesday Book, the BBC issued a modern version, including photographs, text, and maps documenting how Britain looked in 1986. Instead of using vellum, or even paper, the material was assembled in digital formats and issued on 12-inch diameter video disks, which could be read only by specially equipped computers (see Figure 3.18). The project was meant to preserve forever a detailed snapshot of late twentieth-century Britain, and to make it available immediately to schools and libraries everywhere.

By 2001, the modern Domesday Book was unreadable. The computers and disk readers it required were obsolete and no longer manufactured. In 15 years, the memory even of how the information was formatted on the disks had been forgotten. Mocking the project's grand ambitions, a British newspaper exclaimed, "Digital Domesday Book lasts 15 years not 1000."



"Domesday Redux," from *Ariadne*, Issue 56.

FIGURE 3.18 A personal computer of the mid-1980s configured to read the 12-inch videodisks on which the modern "Domesday Book" was published.

Paper and papyrus thousands of years older even than the original Domesday Book are readable today. Electronic records become obsolete in a matter of years. Will the vast amounts of information now available because of the advances in storage and communication technology actually be usable a hundred or a thousand years in the future, or will the shift from paper to digital media mean the loss of history?

The particular story of the modern Domesday Book has a happy ending. The data was recovered, though just barely, thanks to a concerted effort by many technicians. Reconstructing the data formats required detective work on masses of computer codes (see Figure 3.19) and recourse to data structure books of the period—so that programmers in 2001 could imagine how others would have attacked the same data representation problems only 15 years earlier! In the world of computer science, "state of the art" expertise dies very quickly.

The recovered modern Domesday Book is accessible to anyone via the Internet. Even the data files of the original Domesday Book have been transferred to a web site that is accessible via the Internet.



themselves be updated periodically so that the information will be with us forever?

Or would we be wisest to do the apparently Luddite thing: to print everything worth preserving for the long run—electronic journals, for example—so that documents will be preserved in the only form we are *certain* will remain readable for thousands of years?



The digital revolution put the power to document ideas into the hands of ordinary people. The technology shift eliminated many of the intermediaries once needed to produce office memoranda and books. Power over the thoughts in those documents shifted as well. The authority that once accompanied the physical control of written and printed works has passed into the hands of the individuals who write them. The production of information has been democratized—although not always with happy results, as the mishaps discussed in this chapter tellingly illustrate.

We now turn to the other half of the story: how we get the information that others have produced. When power over documents was more centralized, the authorities were those who could print books, those who had the keys to the file cabinets, and those with the most complete collections of documents and publications. Document collections were used both as information choke points and as instruments of public enlightenment. Libraries, for example, have been monuments to imperial power. University libraries have long been the central institutions of advanced learning, and local public libraries have been key democratizing forces in literate nations.

If everything is just bits and everyone can have as many bits as they want, the problem may not be having the information, but finding it. Having a fact on the disk in your computer, sitting a few inches from your eyes and brain, is irrelevant, if what you want to know is irretrievably mixed with billions of billions of other bits. Having the haystack does you no good if you can't find your precious needle within it. In the next chapter, we ask: Where does the power now go, in the new world where access to information means finding it, as well as having it?